**SPHERA (RSE SpA): documentation file of the code master version**

1. Description and references 5

2. Warranties and responsabilities 7

3. Citation of SPHERA v.9.0.0 8

4. SPHERA developers/authors 9

5. SPHERA official users 10

6. The scheme for transport of solid bodies and the Semi-Analytic approach as a boundary treatment scheme for fixed boundaries 13

6.1. Smoothed Particle Hydrodynamics (SPH) 13

6.2. SPH approximation of the balance equations of fluid dynamics with the boundary treatment scheme of the Semi-Analytic approach 14

6.3. Wall drag and wall-function similarity theory: adaptation to flash floods 17

6.3.1. Bottom drag for flash floods and SPHERA wall functions 17

6.3.2. Flow regimes of flash floods 18

6.3.3. Structure of the turbulent Boundary Layer of a flash flood 19

6.3.4. Wall shear stress 19

6.3.5. Turbulent viscosity for the Neutral Surface Boundary Layer 20

6.3.6. Slip coefficient for the Neutral Surface Boundary Layer 20

6.3.7. Roughness characteristic length 21

6.3.8. Roughness length for flash floods: CORINE Land Cover, DEM and DTM 22

6.4. Dirichlet’s boundary conditions for the water depth 24

6.5. Inlet sections for Bazin’s sharp-crested weirs 24

6.6. SPH balance equations for rigid body transport 25

6.7. Improving 3D rotations 27

6.8. Reaction forces 29

6.9. Sliding friction force 30

6.9.1. Aerial stage (non-negative value for the input friction angle and body-frontier interactions) 30

6.9.2. Aerial stage (negative value for the input friction angle or body-body interactions) 30

6.9.3. Submerged stage 31

6.10. Body-boundary normal reaction force under sliding (at null normal velocity) 31

6.10.1. Aerial stage 31

6.10.2. Submerged stage 32

6.11. Fluid-body interaction terms 32

6.12. Solid-solid interaction terms 35

6.13. Normal restitution coefficient 37

7. The scheme for dense granular flows 38

7.1. Mixture model for dense granular flows 38

7.2. 2-interface 3D erosion criterion 44

7.3. A simplified approach for soil liquefaction 46

8. The DB-SPH boundary treatment scheme 47

8.1. DB-SPH particle approximation and modifications of the balance equations 47

8.2. 1D Linearized Partial Riemann Solver 48

8.3. Semi-particle volume 49

8.4. DB-SPH inlet and outlet sections 49

8.5. Shear stress boundary terms for mobile wall frontiers 50

9. Time integration schemes (Leapfrog, Euler, Heun) 51

10. The substation-flooding damage scheme 53

10.1. Mathematical models 53

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) 53

10.1.2. Flood-induced damage to the components of the electrical substations 54

10.2. The numerical models 55

11. Developer guide 56

11.1. SPHERA v.9.0.0: synthetic description of the program units 56

11.1.1. Program units for the boundary conditions (“BC”) 56

11.1.2. Program units for the continuity equation 56

11.1.3. Program units for the momentum equation 56

11.1.4. Program units for the transport of solid bodies 56

11.1.5. Program units for the constitutive equation 56

11.1.6. Program units for the boundary treatment scheme DB-SPH 57

11.1.7. Program units for the erosion criterion 57

11.1.8. Program units on geometry (i.e., analytic geometry, algebra, …) 58

11.1.9. Program units for the initial conditions (IC) 58

11.1.10. Draft program units for the turbulent dispersion of granular material 58

11.1.11. Program units for the main algorithms 58

11.1.12. Modules 58

11.1.13. Program units for the neighbouring search, the smoothing operators and the interface detection. 58

11.1.14. Program units for post-processing 59

11.1.15. Program units for pre-processing 61

11.1.16. Program units for the boundary treatment scheme SA-SPH 67

11.1.17. Program units for managing Fortran character variables 67

11.1.18. Program units for time integration 67

11.2. Style formatting 67

11.3. Modifications with respect to SPHERA v.8.0 68

12. User guide 69

12.1. Installation 69

12.2. Commented template of the main input file of SPHERA v.9.0.0 70

12.3. Tutorials 70

12.3.1. “body\_body\_impacts” 71

12.3.2. “body\_boundary\_impacts” 71

12.3.3. “dam\_breach\_ICOLD\_trunks” 71

12.3.4. “db\_2bodies” 71

12.3.5. “db\_2D\_demo” 71

12.3.6. “db\_Alpe\_Gera” 71

12.3.7. “db\_Alpe\_Gera\_Lanzada\_substations” 71

12.3.8. “db\_body\_exp\_UniBas” 71

12.3.9. “db\_ICOLD” 71

12.3.10. “db\_multi\_body” 71

12.3.11. “db\_squat\_obstacle” 71

12.3.12. “db\_tall\_obstacle” 71

12.3.13. “dike\_breach\_2D\_expSchHag12JHR\_ID22” 72

12.3.14. “edb\_2D\_demo” 72

12.3.15. “edb\_2D\_FraCap02” 72

12.3.16. “edb\_2D\_FraCap02\_Taipei” 72

12.3.17. “edb\_2D\_Spi05” 72

12.3.18. “edb\_ICOLD” 72

12.3.19. “edb\_KarlSand” 72

12.3.20. “edb\_Pon10” 72

12.3.21. “floating\_cube\_stability” 72

12.3.22. “flushing\_2D” 72

12.3.23. “flushing\_3D” 72

12.3.24. “jet\_plate” 72

12.3.25. “rectangular\_side\_weir\_Fr\_0\_491” 73

12.3.26. “San\_Fernando\_Lower\_van\_Norman\_dam\_liquefaction” 73

12.3.27. “sloshing\_tank\_TbyTn\_0\_78” 73

12.3.28. “sloshing\_tank\_TbyTn\_1\_07” 73

12.3.29. “spherical\_Couette\_flows” 73

12.3.30. “SPH\_udb\_exp\_Kim2015HYDROL” 73

12.3.31. “submerged\_landslide” 73

12.3.32. “still\_water\_tank” 73

12.3.33. “wave\_motion\_for\_WaveSAX” 73

12.3.34. “wedges\_falls\_on\_still\_water” 73

13. Use procedure of the modelling chain of SPHERA 74

13.1. Engauge Digitizer 74

13.2. Site overview 74

13.3. The SRTM3 dataset and further DEM-DTM datasets 74

13.4. GDAL 75

13.5. DEM2xyz: preliminary elaboration of the raw DEM and DTM files 75

13.6. Paraview: post-processing of the preliminary DEM and DTM files 76

13.7. Elaboration of a unique DEM-DTM integrated surface 76

13.8. DTM/DEM dataset merging 78

13.9. DEM2xyz: further modifications to the DEM-DTM improvements 78

13.10. Paraview: post-processing of the final output of DEM2xyz 79

13.10.1. Simple use procedure to design a weir 79

13.11. Modification of the spatial resolution of the DEM-DTM 79

13.12. SnappyHexMesh 80

13.13. ply2SPHERA\_perimeter 80

13.1. Paraview: procedures to pre-process the CORINE Land Cover (CLC) maps 80

13.2. SPHERA 81

13.3. Gnuplot 81

13.4. GSView 81

13.5. Grid Interpolator: post-processing of the 2D output fields of SPHERA 81

13.6. Paraview: post-processing and visualization of the 2D and 3D output fields of SPHERA 82

13.7. Image Magick 82

13.8. Virtual Dub 83

14. SPHERA acknowledgments 84

15. SPHERA registration 85

16. GNU Free Documentation License 86

References 95

Documentation Copyright:

Copyright (C) 2015-2022 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).

# 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 hundreds of squared kilometres) and fast landslides, sloshing tanks, sea waves and hydroelectric plants.

With Copyright 2005-2022 (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, Emanuela Abbate, 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, [1]):

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, [2]):

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, [3]):

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, [4]):

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, [5]):

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](mailto: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.

Further updates not reported in this documentation refer to the following documents:

* Amicarelli A., Abbate E., Frigerio A.; submitted; SPH modelling of a dike failure with detection of the landslide sliding surface and damage scenarios for an electricity pylon; International Journal of Computational Fluid Dynamics.
* Andrea Amicarelli, Francesco Apadula; submitted; SPH modelling of the Vajont dam-overtopping flood with wall functions adapted to flash floods and weir-like inlet sections; Environmental Fluid Mechanics.

# 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. ”

# 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).”

# 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.

# 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.

|  |
| --- |
|  |

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

|  |
| --- |
|  |

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

|  |
| --- |
|  |

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

SPHERA is available at github.com. 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 with possible minor modifications to the code, respecting the code name, license, citations and Copyright;
4. independently introduce relevant modifications to SPHERA, thus obtaining a FOSS derived code (which has a different name from SPHERA but has to cite SPHERA as the original code, mentioning all the parts of the original code, its Copyright and license) and redistribute it (according to the GNU-GPL license and SPHERA citation terms and Copyright), or propose such modifications to RSE for their integration in SPHERA;
5. to propose to RSE some program units not belonging to SPHERA and developed with independent funds: they can 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: they can be released under the GNU-LGPL license and integrated in SPHERA, without constraints for the authors to make their original code a FOSS in its entirety.

|  |
| --- |
|  |

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

|  |
| --- |
|  |

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

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 (2021, [6]) public repository; 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).

If any activity dedicated to SPHERA is declared to RSE, that activity might be reserved by RSE for a specific user during a specific period to avoid redundancy and conflicts between users.

Finally, SPHERA is indexed in the list of SPH codes of SPHERIC (2021, [7]).

It is worth noticing that any code (included the Open-Source codes) has its own Copyright which is transmitted to the derived software.

It is worth noticing that a translation of the code (or some parts of it) in another language implies no Copyright change.

"unofficial modified subversion of SPHERA (RSE SpA)"

|  |
| --- |
|  |

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

# 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.6.6) dynamics and the 2-way interaction terms related to both fluid-body (Sec. 6.11) and solid-solid (Sec.6.12) interactions.

The following sub-sections provide details on: improving 3D rotations (Sec.6.7); the sliding friction force (Sec.6.9); the body-boundary normal reaction force under sliding (at null normal velocity, Sec.6.10); the normal restitution coefficient (Sec.6.13).

Please refer to SPHERA main references (Sec.1) for further details.

* 1. Smoothed Particle Hydrodynamics (SPH)

Smoothed Particle Hydrodynamics (SPH) represents a mesh-less CFD technique, whose computational nodes are represented by numerical fluid particles.

In the continuum, the functions and derivatives in the fluid dynamics balance equations are approximated by convolution integrals, which are weighted by interpolating (or smoothing functions), called kernel functions.

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

|  |  |
| --- | --- |
|  | (6.1) |

where *W* is the kernel function, *x0* the position of a generic computational point and *Vh* the integration volume, which is called kernel support. This is represented by a sphere of radius 2*h*, possibly truncated by the frontiers of the fluid domain.

The cubic *b*-spline function (Monaghan & Lattanzio, 1985, [8]):

|  |  |
| --- | --- |
|  | (6.2) |

is the reference kernel function in SPHERA, unless otherwise stated.

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:

|  |  |
| --- | --- |
|  | (6.3) |

The integration also involves the surface *Ah* 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, [9]; Hashemi et al., 2012, [10]; Macia et al., 2012, [11]; Mayrhofer et al., 2013, [12]; Ferrand et al., 2013, [13]; Amicarelli et al., 2013, [3]).

Far from boundaries, the SPH particle approximation of (6.1) reads:

|  |  |
| --- | --- |
|  | (6.4) |

where a summation on particle volumes (**replaces the volume integral. The subscripts “*0*” and “*b*” refer to the computational particle and its “neighbouring particles” (fluid particles within the kernel support of the computational particle), respectively.

Usually, the approximation (6.4) 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, [14]; Amicarelli et al., 2015, [2]; Di Monaco et al., 2011, [5]); sloshing tanks (Souto-Iglesias et al., 2006, [15]; Amicarelli et al., 2013, [3]); wave motion (e.g., Patel et al., 2009, [16]; Antuono et al., 2011, [17]; Liu et al., 2013, [18]; Omidvar et al., 2012, [19]); hydraulic turbines (Marongiu et al., 2010, [20]); fast landslides (e.g., Kumar et al., 2013, [21]); erosion and bed-load transport (Manenti et al., 2012, [4]); liquid jets (e.g., Koukouvinis et al., 2013, [22]); pollutant dispersion; astrophysics (e.g., Price, 2012, [23]); magneto-fluid dynamics (e.g., Price, 2012, [23]); multi-phase and multi-fluid flows (e.g., Kajtar & Monaghan, 2012, [24]).

* 1. SPH approximation of the balance equations of fluid dynamics with the boundary treatment scheme of the Semi-Analytic approach

This section relies on Di Monaco et al. (2011, [5]), Amicarelli et al. (2015, [2]) and Amicarelli et al. (2017, [1]), 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, [25]). Its basic features are deeply described in Di Monaco et al. (2011, [5]) and here briefly reported.

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

|  |  |
| --- | --- |
|  | (6.5) |

where  is the velocity vector, *p* pressure, ** fluid density, *ij* Kronecker’s delta, *x* position and *t* time. We need to compute (6.5) 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 (6.5), as provided by the SPH approximation of the first derivative of a generic function (*f*), according to the semi-analytic approach (“SA”; Di Monaco et al., 2011, [5]):

|  |  |
| --- | --- |
|  | (6.6) |

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 (6.6) is performed over all the fluid particles “*b*” (neighbouring particles with volume **) in the kernel support of the computational fluid particle (“*0*”). At the same time, the volume integral in (6.6) represents the boundary term, which is a convolution integral on the truncated portion of the kernel support. In this fictitious and outer volume (*Vh’*), one needs to define the generic function f (pressure, velocity or density alternatively).

The semi-analytic approach (“*SA*”) (the version of Di Monaco et al., 2011, [5]) hypothesizes the following linearization and assumptions to compute f in *Vh’*:

|  |  |
| --- | --- |
|  | (6.7) |

The peculiar “*SA*” values of the functions and derivatives within *Vh’* are assigned to represent a null normal gradient of reduced pressure at the frontier interface (while considering uniform density):

|  |  |
| --- | --- |
|  | (6.8) |

The integral computation considers the solid angle (*as*) as the integration variable. The functions *g1* (m-1) and *g2* are summations (over the plane boundaries intercepting the truncated kernel support) of summations (over *r*, distance between “*0*” and each external discretized volume) of analytical functions depending on *W* and *re* (s). This is the distance between the computational particle and the point on the line passing for “*0*” and the point on the frontier described by the solid angle. Eq.(6.7) is rearranged as follows:

|  |  |
| --- | --- |
|  | (6.9) |

The two integrals in (6.9) are discretized using 64 integration nodes (16 meridians and 4 sectors):

|  |  |
| --- | --- |
|  | (6.10) |

where *jz* is the zenith angle and *q* is the azimuth angle within the spheric kernel support. Considering this discretization, Eq.(6.9) is further rearranged as follows:

|  |  |
| --- | --- |
|  | (6.11) |

The velocity vector is taken as uniform in the outer part of the kernel support. Here *uSA* is decomposed in the sum of a vector normal to boundary  and a tangential vector .

Under free-slip conditions, the tangential component of the velocity vector in the truncated portion of the kernel support is:

|  |  |
| --- | --- |
|  | (6.12) |

whereas it assumes the following expression under no-slip conditions:

|  |  |
| --- | --- |
|  | (6.13) |

where *t* is the unit vector aligned with the tangential component of the fluid velocity (with respect to the wall).

The normal component of the velocity vector in the truncated portion of the kernel support, no matter about the slip conditions, reads:

|  |  |
| --- | --- |
|  | (6.14) |

where *nw* is the normal vector of the wall surface, as defined by its local orientation.

Thus, the associated velocity vectors can be assigned to the truncated portion of the kernel support both under no-slip conditions:

|  |  |
| --- | --- |
|  | (6.15) |

and free-slip conditions:

|  |  |
| --- | --- |
|  | (6.16) |

together with the following velocity differences, both under free-slip conditions:

|  |  |
| --- | --- |
|  | (6.17) |

and no-slip conditions:

|  |  |
| --- | --- |
|  | (6.18) |

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:

|  |  |
| --- | --- |
|  | (6.19) |

where *Cs* is introduced to represent a fluid-body interaction term.

This approximation of the continuity equation does not violate the mass conservation as the particle volume is undetermined.

On the other hand, we can analogously derive the approximation of the momentum equation (the notation indicates the SPH particle -discrete- approximation):

|  |  |
| --- | --- |
|  | (6.20) |

where *as*represents a new acceleration term due to the fluid-body interactions, *M* is the artificial viscosity (Monaghan, 2005, [26]), *m* the particle mass and *r* the relative distance between the neighbouring and the computational particle.

The inner pressure gradient term in (6.20) is conservative. The inner viscous shear stress term in (6.20) is conservative. The inner artificial viscosity term in (6.20) is conservative provided that the particle artificial viscosity and the particle density are homogeneous.

Zeroing (molecular) viscosity reduces CPU time. A configuration with no-slip conditions and null viscosity makes sense if it is demonstrated that the value of viscosity is uninfluential.

No-slip conditions might possibly provide the same results as free-slip conditions if both the following conditions are satisfied: the test case approximately provides no-slip conditions even if free-slip conditions are used; dx is small enough.

If no-slip conditions can be used with body transport, then they are more convenient than free-slip conditions in terms of CPU time.

The second-to-last term of (6.20) is defined by Di Monaco et al. (2011, [5]). It is a viscous term, whose formulation is mixes Cleary’s and Morris’ formulations (Basa et al., 2009, [27]):

|  |  |
| --- | --- |
|  | (6.21) |

where ** and *r,2* are negligible constants avoiding divergent behaviours. In case ** and *r* do not tend to zero, then (6.21) assumes the following form:

|  |  |
| --- | --- |
|  | (6.22) |

One considers Morris’ term (Basa et al., 2009, [27]):

|  |  |
| --- | --- |
|  | (6.23) |

and that the following expression:

|  |  |
| --- | --- |
|  | (6.24) |

is a factor analogous to Cleary’s formulation (Basa et al., 2009, [27]):

|  |  |
| --- | --- |
|  | (6.25) |

It follows that (6.22) is equal to Morris’ term only in case of uniform density and viscosity:

|  |  |
| --- | --- |
|  | (6.26) |

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

|  |  |
| --- | --- |
|  | (6.27) |

The artificial sound speed *c* is 10 times higher than the maximum fluid velocity (WC approach) and “*ref*” stands for a reference state.

It is worth noticing that the variables at the monitoring elements are computed as follows. If a monitoring element has a neighbouring SASPH frontier, the variables of the monitoring element assume the values of the closest SPH particle. Otherwise, SPH particle approximations apply using the discrete Shepard coefficient of the element.

* 1. Wall drag and wall-function similarity theory: adaptation to flash floods

The formulation for the slip coefficient presented in this section is an alternative to imposing the free-slip or no-slip conditions of Sec.6.2.

SPHERA simulates the solid walls impacted by a flash flood in terms of “explicit roughness” and “sub-grid roughness”. The first is integrated in the topographic surface as explicit solid obstacles with fluid-structure interactions. The latter is not explicitly reproduced, but its effects are considered by the wall-function similarity theory, as reported in this section. A formulation for the wall shear stress *tw* (Pa) locally exerted by the sub-grid roughness to a flash flood is adapted from the wall-function theory. Here the wall shear stress is equivalent to the wall drag per unit of surface as the formulation of the current section applies to the sub-grid roughness. The novelties with respect to the state-of-the-art on floods are presented (Sec.6.3.1). The detection of the flow regime of a flash flood is investigated (Sec.6.3.2). The structure of the turbulent BL of a flash flood is discussed (Sec.6.3.3). The representation of the wall shear stress (Sec.6.3.4) for the Neutral Surface BL (NSBL) involves the turbulent viscosity (Sec.6.3.5), the slip coefficient (Sec.6.3.6) and the roughness characteristic length. The latter is provided with a formulation for the rough-wall regime of flash floods (6.3.8). This accounts for the CORINE Land Cover, Digital Elevation Models (DEMs) and Digital Terrain Models (DTMs).

* + 1. Bottom drag for flash floods and SPHERA wall functions

Many approaches based on wall functions are adopted by CFD codes close to the solid walls of industrial flows to reduce the computational time (e.g., Davidson, 2015, [28]). Alternatively, the spatial resolution refinement necessary to solve the smallest motion scales would normally require unaffordable computational resources. This condition is further emphasized when dealing with the large domains of the environmental flows. In this context, wall functions are commonly used to represent the Atmospheric SBL (e.g., Pelliccioni et al., 2012, [29]; Jancewicz & Szymanowski, 2017, [30]), but they are rarely applied to simulate floods, where the state-of-the-art is represented by imposing free-slip conditions (i.e., no wall drag) or assuming Chezy-like formulations. The last case is the usual choice for SWE 1D and 2D models for floods (e.g., Cea et al., 2010), as described in the following. Friction velocity is expressed as function of the skin friction coefficient under uniform conditions. This coefficient is alternatively expressed by means of Colebrook’s generalized formula (Da Peppo & Datei, 2003, [31]), among the most accurate expressions for SWE equations, or simplified expressions like Manning’s or Gauckler-Strickler’s formula (e.g., Chen et al., 2012, [32]). With respect to the state-of-the-art solutions for the bottom drag exerted by floods (e.g., Kim et al., 2015, [33]), the current formulation shows the following novelties:

1. consistency with the NSBL wall-function similarity theory;
2. dependency on obstacle height, CLC class and the truncation errors of DTMs and DEMs;
3. 3D local and time-dependent formulation: e.g., the wall drag varies within a flow section and changes with time, whereas Manning’s coefficient is 1D and stationary;
4. basic accuracy under 1D/2D uniform conditions: the wall drag is correctly influenced by the hydraulic radius *RH* (m) and the water depth *h* (m), whereas Manning’s coefficient *n* (s×m-1/3) only depends on the soil-cover type;
5. basic accuracy under 1D/2D non-uniform conditions: in SWE formulations the bottom drag depends on the head loss per unit length, which is associated with the whole flow section and suffers from uniform-flow assumptions, especially concerning the hydrodynamic thrust;
6. unique formulation for any kind of flow-section shape: complicated formulae are rarely used in SWE formulations to partially cope with this topic;
7. suitability for any sub-grid roughness element (e.g., river bottoms, floodplains, buildings, trees);
8. differential treatment for “sub-grid roughness” via wall functions and “explicit roughness” by means of explicit obstacles and fluid-structure interactions; solid obstacles are a key-issue for SWE formulations, where no fluid-structure interaction is simulated: an explicit drag coefficient is requested for every relevant obstacle (Di Risio & Sammarco, 2020, [34]) or complex schemes consider the effects of rigid vegetation (Baruah & Sarma, 2021, [35]);
9. the wall drag benefits from the absence of limits to the topographic slopes and from the SPH simulation of non-linear currents, the velocity depth-variations, the advective term in the momentum equation, non-hydrostatic pressure fields and wave motion.
   * 1. Flow regimes of flash floods

Flash floods typically flow under rough-wall turbulent regimes, as discussed in the following. Hereafter, only Neutral BLs (NBLs) are considered: Convective and Stable BLs involving buoyancy effects normally do not occur within the domain of a flash flood.

The Reynolds number *Re* and the friction Reynolds number *Re\** synthesize a fluid flow on a wall:

|  |  |
| --- | --- |
|  | (6.28) |

where *U* (m/s) is a velocity scale proportional to the freestream velocity, *D* (m) is a length scale proportional to the BL depth and *u\** (m/s) is the friction velocity. The hydraulic roughness *ks* (m) can be expressed as function of the 90th percentile of the wall roughness-size distribution *d90* (m), according to van Rijn (1982, [36]), and can be approximated by the median roughness diameter *d50* (m), more commonly available:

|  |  |
| --- | --- |
|  | (6.29) |

Depending on the dimensionless numbers mentioned above, the following motion regimes, listed in *Re* increasing order, can be observed:

1. laminar regime: regular motion without turbulence (viscous BL); the wall shear stress depends on the dynamic viscosity *m* (Pa×s), the length and velocity scales: ;
2. transitional turbulent regime with hydraulically smooth walls, here “smooth-wall regime” for sake of brevity: the turbulent regime is not fully developed and the wall roughness does not affect *tw* (*Re\**≤5); the top of the viscous sub-layer fully covers the roughness elements; the wall shear stress depends on the momentum turbulent fluxes occurring in the turbulent sub-layers by means of density: ;
3. transitional turbulent regime with intermediate hydraulic roughness, here “intermediate-roughness regime” for sake of brevity: the turbulent regime is not fully developed and the wall roughness influences *tw* (5<*Re\**<70); the top of the viscous sub-layer is locally completely eroded on the top of the tallest roughness elements; the wall shear stress depends on the hydraulic roughness: ; the velocity in the SBL is related to the wall shear stress via *u\**: *u* depends on *ks* too;
4. fully-developed turbulent regime with hydraulically rough walls, here “rough-wall regime” for sake of brevity: the turbulent regime is fully developed within the BL; the viscous sub-layer is locally absent or very thin so that dynamic viscosity does not influence the wall shear stress: ; this condition has been verified only for *Re\**≥70; the SBL velocity does not depend on the dynamic viscosity (e.g., Ge et al., 2017, [37]).

In case of flash floods *n*=1.0×10-6 m2/s and the orders of magnitude of the velocity and length scales are *U*10-1-101m/s and *D*10-1-101m, respectively. Reynolds number is so high *Re*104-108 that turbulent BLs systematically develop, whereas laminar regimes do not occur. Further, the rough-wall regime is normally observed, whereas the other turbulent regimes are rare, as explained in the following. The orders of magnitude of the friction velocity and the hydraulic roughness are *u\**10-2-100m/s and *ks*=30*z0*10-3-101m, respectively, where *z0* (m) is the roughness characteristic length (Sec.6.3.7): *Re\**101-107. This means that flash floods normally flow under the rough-wall regime with some exceptions under the intermediate-roughness regime.

* + 1. Structure of the turbulent Boundary Layer of a flash flood

During a SPH simulation for floods, the computational particles close to the solid walls lie below the “outer layer” (or “core layer”) of the turbulent BL. The underlying “inner layer” is composed by the SBL, the buffer layer (BFL) and the viscous sub-layer (VSL). In particular, the flood particles mentioned above lie within the SBL. The underlying BFL and VSL are normally so thin that they are out of scale. The top height of the VSL and the BFL are *zTOP,VSL*=5*n*/*u\** (m) and *zTOP,BFL*=30*n*/*u\** (m), respectively (Montuori, 1997, [38]). In case of flash floods, their orders of magnitude are *zTOP,VSL*10-6-10-3 m and *zTOP,VSL*10-5-10-2m, according to the estimates of Sec.6.3.2. The top height of the SBL is *zTOP,SBL*=0.2*dT* (Davidson, 2015, [28]; Krank, 2019, [39]), where *dT* is the depth of the turbulent BL. In SPHERA, only the fluid region distant from wall no more than *dx* is interested by wall functions. The exact detection of the 1-particle layer close to the topographic surface should be avoided as it is useless and too cumbersome in this mesh-less context.

* + 1. Wall shear stress

The SASPH boundary term in the momentum equation of SPHERA involves a shear-stress gradient boundary term *BCME,F,t* (m/s2) presented in Di Monaco et al. (2011, [5]) and validated in following studies. Fixed a SPH computational particle close to a wall, the term *BCME,F,t* depends on the dynamic viscosity, a fictitious boundary velocity, the wall geometry, the velocity and position of the computational particle. The above viscosity was either the mixture viscosity of dense granular flows (Amicarelli et al., 2017, [1]) or the molecular viscosity of mono-phase fluids under laminar regimes (Paggi et al., 2021, [40]). In case of flash floods, the viscosity included in *BCME,F,t* has to be represented by the turbulent viscosity *mT* (Pa×s), since molecular viscosity is negligible in the SBL and here floods are mono-phase flows. Thus, a simple formulation for *mT* in the NSBL is adopted (Sec.6.3.5).

For each particle-wall interaction, a fictitious boundary velocity is defined outside the fluid domain, within the truncated portion of the SPH kernel support, namely the influence region of the SPH particle. This velocity is assigned by means of the slip coefficient *fs*, which was previously treated as an input parameter to represent either free-slip conditions or no-slip conditions. Here a formulation for the slip coefficient of computational particles in the NSBL is presented (Sec.6.3.6).

Both the formulations for *mT* and *fs* are consistent with the wall-function similarity theory.

* + 1. Turbulent viscosity for the Neutral Surface Boundary Layer

A simple formulation for the turbulent viscosity is adopted for the wall functions. This is consistent with the NSBL similarity theory. In the Prandtl logarithmic function for the NSBL:

|  |  |
| --- | --- |
|  | (6.30) |

*r* (m) is the distance from the wall, *t* (Pa) is the shear stress and *kv*=0.40 is the von Karman constant. The turbulent shear stress is approximately constant in the SBL and is approximately equal to the molecular shear stress at wall. The spatial constraint in (6.30) reduces to the condition *r*≥*z0* as the particles representing the flood domain close to the solid walls lie in the SBL (Sec.6.3.3). Adopting the mixing-length turbulence scheme and considering the friction velocity in (6.30), the turbulent viscosity of a generic particle-wall interaction “*0w*” is approximated as follows:

|  |  |
| --- | --- |
|  | (6.31) |

where the SPH particle lies in the NSBL and the bottom line relates to the limiter of (6.37).

* + 1. Slip coefficient for the Neutral Surface Boundary Layer

In many CFD models, the slip coefficient is associated with the computational nodes close to the walls and allows to represent the wall shear stress (Sec.6.3.4). The slip coefficient is here a proxy variable associated with a numerical boundary velocity *uBC* (m/s) to impose Neumann’s boundary conditions. A flawless numerical assessment of the wall shear stress would require an expression to relate *uBC* or *fs* with the SPH wall shear stress, which could be constrained to the wall-function theory. Unfortunately, such SPH approximation does not allow any suitable formula inversion. Alternatively, a simplified expression for *fs* is proposed, according to the NSBL wall-function theory.

The shear stress*t* exerted by the fluid on a wall “*w*” is expressed as follows:

|  |  |
| --- | --- |
|  | (6.32) |

where (Pa) is the stress tensor, is the stress vector exerted on a wall with normal *n*, *t* is the unit vector tangential to the wall and aligned with the local velocity vector.

Combining (6.32) and (6.30), the wall shear stress associated with a point in the NSBL reads:

|  |  |
| --- | --- |
|  | (6.33) |

The wall shear stress relates to a computational particle “0” interacting with the wall “*w*” by means of a linear Centred-Finite-Difference approximation of the velocity gradient:

|  |  |
| --- | --- |
|  | (6.34) |

where *ut* (m/s) indicates the tangential component of velocity with respect to the wall and *uBC* is formally associated with a virtual particle outside the fluid domain, whose position is symmetric to *x0* with respect to the wall. The definition of the slip coefficient *fs* permits to express the wall shear stress as function of the wall velocity instead of *uBC*:

|  |  |
| --- | --- |
|  | (6.35) |

The equivalent expression for the fictitious boundary velocity reads:

|  |  |
| --- | --- |
|  | (6.36) |

In the region between the computational particle and the wall, velocity cannot be larger than the particle velocity or smaller than the wall velocity. The associated limits are represented by symmetric free-slip conditions with *fs*=0 and anti-symmetric no-slip conditions with *fs*=1.

Provided the expressions for the NSBL wall shear stress (6.33) and turbulent viscosity (6.31), a formulation for the slip coefficient of the computational particles in the NSBL is proposed:

|  |  |
| --- | --- |
|  | (6.37) |

where the bottom line is a numerical limiter avoiding *fs*>1 and *z0* depends on the flow regime (Sec.6.3.7). Free-slip conditions are asymptotically represented by . Each “boundary element - particle” interaction has its own slip coefficient.

* + 1. Roughness characteristic length

The formulations for the slip coefficient and the turbulent viscosity of Sec.6.3.5-6.3.6 need the definition of the roughness length *z0*. This section applies to the rough-wall regime and is integrated in the *z0* formulation for flash floods (Sec.6.3.8). The other regimes are not represented in SPHERA. However, the intermediate-roughness regime might be useful for certain floods for which an alternative formulation is proposed at the end of the current section.

In the rough-wall regime, (6.29) can be used to extend the Nikuradse wall function for covers of uniform granular material to non-uniform materials:

|  |  |
| --- | --- |
|  | (6.38) |

Considering (6.29), the roughness median diameter *d50* is interpreted as the sub-grid roughness depth *Hsgr* (m) halved. The roughness length in the rough-wall regime assumes the following form:

|  |  |
| --- | --- |
|  | (6.39) |

Under the same regime, the top height of the BFL reads:

|  |  |
| --- | --- |
|  | (6.40) |

In particular, *zTOP,BFL*<*z0* when *Re\**>900. Under these conditions, the part of the SBL below *z0* is not represented by the logarithmic law (*r*≥*z0*) as the mean velocity of the resolved turbulent scales is null. For instance, this is the case of a flood sub-domain close to the ground in a dense forest whose trees are sub-grid roughness elements. This is also a physical interpretation to the no-slip low-pass filter in the formulation (6.37) for the slip coefficient of SPHERA, which numerically imposes an analogous and more rigid constraint (*r*≥*z0e*) to the application of the logarithmic law.

The SBL Nikuradse wall function for the smooth-wall regime provides the following expression for the roughness length, after minor rearrangements to use the natural logarithm:

|  |  |
| --- | --- |
|  | (6.41) |

* + 1. Roughness length for flash floods: CORINE Land Cover, DEM and DTM

A formulation for the roughness length under flash floods is proposed. It is an adaptation of the formulation of Sec.6.3.7 for the rough-wall regime to consider the CORINE Land Cover (CLC) maps, Digital Elevation Models (DEMs), Digital Terrain Models (DTMs) and the obstacle heights.

The topographic surface is represented in SPHERA by a DTM, a DEM or a hybrid DEM/DTM surface. In the last case, the DEM is locally replaced by the DTM only where their height difference is smaller than *Hf,DEM* (m), which is a high-pass filter on the height of the DEM roughness elements. The roughness length *z0* is expressed as follows:

|  |  |
| --- | --- |
|  | (6.42) |

where the term *z0,e* (m) is associated with the truncation error (m) of the topographic surface, *z0,LC* (m) relates to the soil-cover roughness obtained by experimental assessments, *z0,CLC* (m) is the soil-cover contribution linked to the CLC classes (Table 6.1) and *Hs* is the Heaviside step function. This activates the soil-cover term only where the DEM is locally removed by the filter *z0,f,DEM*, associated with *Hf,DEM*. The DEM absence is represented by the formal condition *Hf,DEM*=*Hr,max*, where *Hr* is the depth of the overall roughness, considering both its sub-grid and explicit components. After (6.39), the following expressions apply to (6.42):

|  |  |
| --- | --- |
|  | (6.43) |

where *sz* (m) is the standard deviation of the DEM or DTM heights. The truncation error of each topographic surface is uniform within the same file and is treated as a numerical roughness depth in (6.43). It is assessed considering the *sz* variations between two representative spatial resolutions, e.g., *Dx* (m) and 2*Dx*, but only if *Dx* is fine enough. Otherwise, any local convergence of *sz* is unreliable. This condition is peculiar of any coarse DEM (e.g., a DEM with *Dx*=31m as in the dataset SRTM3, 2014, [41]), whose truncation error is negligible with respect to *z0,LC*. This means that the local roughness elements of a coarse DEM are not explicitly represented. Thus, a coarse DEM can be treated as a DTM with negligible *z0,e*, when using (6.43).

|  |  |  |  |
| --- | --- | --- | --- |
| **CLC class code** | **2018 CLC class description** | ***z0,CLC* (m)**  **SPHERA**  **(CLC class mid-point)** | ***z0* (m)**  **Jancewicz & Szymanowski (2017)** |
| **10000** | **Artificial areas** |  |  |
| 11000 | Urban fabric | 1.0 | 1.5 (“continuous urban fabric”)  0.6 (“discontinuous urban fabric”) |
| 12000 | Industrial, commercial and transport units | 1.0 | 1.5 (“continuous urban fabric”)  0.6 (“discontinuous urban fabric”)  (0.04) (“roads and associated lands”) |
| 13000 | Mine, dump and constructon sites | 0.15 | 0.01 (“dump sites; mineral extraction sites”)  0.3 (“construction sites”) |
| 14000 | Artificial non-agricultural vegetated areas | 0.3 | 0.3 |
| **20000** | **Agricultural areas** |  |  |
| 21000 | Arable land | 0.03 | 0.03 |
| 22000 | Permanent crops | 0.3 | 0.3 (“agro-forestry areas”)  0.3 (“fruit trees and berry plantations”) |
| 23000 | Pastures | 0.03 | 0.03 |
| 24000 | Heterogeneous agricultural areas | 0.2 | 0.3 (“agro-forestry areas”)  0.1 (“complex cultivation patterns; land principally occupied by agriculture with significant areas of natural vegetation”) |
| **30000** | **Forests and semi-natural areas** |  |  |
| 31000 | Forests | 1.4 | 1.4 (“forests”)  (0.4) (“transitional woodland-shrub”) |
| 32000 | Shrubs and/or herbaceous vegetation associations | 0.2 | 0.2 (“sparsely vegetated areas”) |
| 33000 | Open spaces with little or no vegetation | 0.02 | 0.03 (“natural grassland”)  0.01 (“bare rock”) |
| **40000** | **Wetlands** |  |  |
| 41000 | Inland wetlands | 0.005 | 0.005 (Davenport updated table, “marsh”) |
| 42000 | Coastal wetlands | 0.005 | 0.005 (Davenport updated table, “marsh”) |
| **50000** | **Water bodies** |  |  |
| 51000 | Inland waters | 0.0002 | 0.0002 (Davenport’s updated table, “open sea or lake irrespective of wave size”) |
| 52000 | Marine waters | 0.0002 | 0.0002 (Davenport’s updated table, “open sea or lake irrespective of wave size”) |

Table 6.1. Values of the roughness length *z0,CLC* related to the CLC 2018 classes (Kosztra et al., 2019, [42]), obtained by simple elaboration of the *z0* values validated in Jancewicz & Szymanowski (2017, [30]) on meteorological studies.

According to (6.42), *z0* values are associated with the CLC 2018 classes (Kosztra et al., 2019, [42]). These values are obtained by means of a simple elaboration of the *z0*-values reported in Jancewicz & Szymanowski (2017, [30]). Those values were validated on numerical meso-scale meteorological studies. Under those applications, the DTM truncation errors were implicitly involved in *z0,CLC*. The similarity between the Atmospheric SBL and the SBL of flash floods considers that the *z0* does not depend on the fluid type under rough-wall regimes, but only on the wall geometry via (6.39).

The CLC maps are usually available as shapefiles (“.shp”). In the numerical chain of SPHERA, these files are elaborated by means of GDAL (2021, [43]) and Paraview (Kitware, 2021, [44]) to obtain a set of “.ply” files of CLC triangulated polygons and a text file which links each polygon to its CLC class. These files are elaborated by SPHERA at the resolution requested.

Delaunay’s triangulation executed by Paraview presents some surface overestimations outside the concave polygons. This causes some local polygon-overlapping. To solve this shortcoming, a priority index is introduced in SPHERA to associate the most proper CLC polygon with each point where *z0* is assessed. This index is called “vertex-usage code of a face” *FVUC*. It reduces the discretization errors of Delaunay’s triangulation. The complete removal of such errors, if necessary, is a peculiar task for visualization tools more than SPH codes. Nonetheless, the following algorithm is adopted in SPHERA for *FVUC*. Any barycentre of a cell of the positioning grid of SPHERA can simultaneously lie within the spurious face of a locally concave CLC polygon and the correct face of a locally convex CLC polygon. Usually, all the vertices of the wrong face are already used by other faces of the same triangulated polygon. Contrarily, at least one vertex of the correct face is not used by the other faces of the locally convex polygon. The vertices of a spurious face are usually overused, with respect to the vertices of a correct face. Starting from this principle, often useful but not always exact, a strategy has been adopted to assign higher priorities to the faces with underused vertices.

For each triangular face of a CLC polygon, *FUVC* is defined as follows:

|  |  |
| --- | --- |
|  | (6.44) |

where *Vo* is the occurrence of the generic vertex in all the faces of the same CLC polygon of the current face. The minimum operator applies to alle the  possible combinations of ordered lists of the three vertices “*i,j,k*” of the current face. This operator and the powers of ten in (6.44) guarantee that *FUVC* is dominated by the least used vertex and then, in case of tie, by the following vertices, still ordered according to their underuse in the triangulated polygon. The polygon searching adopts the same “dynamic-vector method” executed in SPHERA to search the neighbouring SPH particles, i.e., those particles influencing the balance equations of any current computational particle.

Finally, where the CLC maps are unavailable, the field of *z0,LC* is reconstructed by reliable *z0*-values measured in the Atmospheric BL for analogous soil covers or using the last formula in (6.43) after an assessment of the maximum height of the sub-grid obstacles.

The same slip coefficients also interest the partial smoothing of the velocity field (on the fly computation is preferred to saving the values), even if this procedure is not activated by default.

One notices that the SASPH method does not apply the slip coefficient to the continuity equation.

Analogous approaches based on wall functions are commonly adopted by CFD codes (e.g., Ferrand et al., 2013, [13]; CFX, 2020, [45]).

* 1. Dirichlet’s boundary conditions for the water depth

The numerical scheme for Dirichlet’s boundary conditions on water depth is reported in the present section. The zone associated with these boundary conditions are defined as *BCz,max* and has to be width at least 2*h* along any horizontal direction of the discretized domain. Many *BCz,max* zones can be defined within the same domain.

Every *BCz,max* zone is continuously filled with SPH particles from the maximum local height of the SPH particles, or from the height of the solid bottom in case of dry conditions, until the height of the free-surface level provided by the input quantity *zBC,max* (m), with local hydrostatic or uniform conditions for pressure (according to the user choice).

The velocity of the new emitted particles is imposed by means of a complete SPH smoothing procedure for the velocity field, selectively executed to exclude the new particles from the role of SPH neighbours. It is not necessary to modify the quantities of the SPH particles already present in the zone *BCz,max*. Every zone of this kind has to be delimited by at least 5 adjacent open sections in the input files, in order to correctly impose Dirichlet’s boundary conditions for the fluid depth. Nonetheless, the zones *BCz,max* can also be used with other purposes (e.g., to treat water reservoirs with a fixed level). The solid bottom (e.g., the DEM-DTM) has to be locally represented by a uniform and isotropic Cartesian grid.

A synthetic description of the algorithm of the boundary treatment of the zones *BCz,max* is reported in the following.

Before the execution of the simulation time steps, and analogously to imposing initial conditions for the inlet sections, the program unit “*BC\_zmax\_t0*” (executed only in 3D, with the boundary treatment method SASPH) selects the zone vertices and the maximum DEM-DTM heights (useful for dry conditions; “*BC\_zmax\_t0*” calls the program unit “*z\_min\_max\_DEM\_DTM\_9p\_stencil*”).

Every time step, after the update of the particle positions (and analogously to the inlet section update), the program unit “*BC\_zmax\_anyt*” is called (only in 3D, with the boundary treatment method SASPH), which calls in turn the subroutine “*z\_FS\_max\_9p\_stencil*” (both for the Leapfrog time scheme and the last time stage of the explicit Runge-Kutta time schemes).

Every time step, after the SPH neighbouring search (i.e., after the execution of the program unit “*CalcVarLength*” and its sub-units), following the subroutine “*BC\_zmax\_anyt*”, a complete SPH smoothing procedure for velocity is applied only to the new particles emitted in the *BCz,max* zones (only in 3D, with the boundary treatment method SASPH), excluding the new particles from the role of SPH neighbours. The procedure is executed with any time integration scheme of SPHERA. At the end of the initialization of the new particles, the particle zone is formally changed from the *BCz,max* zone to the zone of the associated solid bottom *Car\_top\_zone*, so that the new particles can be treated as standard computational nodes after the initialization, even during their staying in the *BCz,max* zone.

The new program units for the treatment of the *BCz,max* zones are “*BC\_zmax\_t0*”, “*BC\_zmax\_anyt*” and “*z\_FS\_max\_9p\_stencil*”. The following program units have been extracted from the subroutine “*GeneratePart*” to make the code more effective: “*domain\_edges*”, “*pos\_plus\_white\_noise*” “*z\_min\_DEM\_DTM\_9p\_stencil*”, “*particle\_position\_extrusion*”, “*particles\_in\_out\_dams*”. The program unit “*z\_min\_DEM\_DTM\_9p\_stencil*” has been generalized and renamed “*z\_min\_max\_DEM\_DTM\_9p\_stencil*”.

* 1. Inlet sections for Bazin’s sharp-crested weirs

A non-stationary rectangular inlet section of SPHERA can be defined as a weir-like inlet section. In this case, SPHERA estimates the inlet weir water depth and imports the inlet weir flow rate from input time series, when available. Otherwise, the inlet flow rate is reconstructed by any pre-processing model.

Swamee’s formula (1988, [46]) is the reference solution for the flow rate overtopped *Qw* (m3/s) on a sharp-crested Bazin’s weir with non-negligible upstream velocity:

|  |  |
| --- | --- |
|  | (6.45) |

where *h* (m) is the reservoir free-surface height over the weir top. *Lw* (m) and *Hw* (m) are the weir length and height, respectively. In SPHERA, a weir inlet section approximates (6.45) by means of a constant discharge coefficient *mw*. This is equal to the “Least-Relative-Error Average” (LRRA) of the maximum and minimum values of *mw*. Further, *h* approximates the water depth at the weir section. This approach avoids the a-priori knowledge of the time series of the inlet water depth.

The LRRA average *xLRRA* of the extreme values of a generic quantity *x* belonging to the interval [*xmin*, *xmax*] is here defined as the constant value which represents the above interval minimizing the relative error *er*. The error obtained by replacing the minimum value *xmin* with *xLRRA* is set equal to the error associated with the replacement of *xmax* with *xLRRA*:

|  |  |
| --- | --- |
|  | (6.46) |

where *xmp* is the arithmetic (mid-point) average. Any other relative error within the above interval is smaller than the error at the extremes. The application of Swamee’s formula (1988, [46]) reveals that *mw,max*=1.20 and *mw,min*=0.61. After (6.46), *mw,LRRA*=0.81, with a relative error *er* for both *mw* and *Q* equal to *er,Q,LRRA*=33%. The arithmetic average and the geometric average “*GM*” do not achieve this result: *xLRRA*≤*xGM*≤*xmp*. Their estimates and relative errors are *mw,GM*=0.85, *er,Q,GM*=39%, *mw,mp*=0.90 and *er,Q,mp*=50%. Swamee’s formula (1988, [46]) is approximated with the constant coefficient *mw,LRRA*:

|  |  |
| --- | --- |
|  | (6.47) |

The inversion of (6.47) provides a closed-form solution for the water depth of an inlet weir section:

|  |  |
| --- | --- |
|  | (6.48) |

The application of the error propagation law to (6.48) reads:

|  |  |
| --- | --- |
|  | (6.49) |

The relative error on the water depth *er,h* is equal to 2/3 the relative error on the flow rate:

|  |  |
| --- | --- |
|  | (6.50) |

The replacement of *mw*(*h*,*Hw*) with *mw,LRRA* determines a maximum relative error on the inlet water depth of *er,h,LRRA*=22%, but three benefits are obtained: the assessment of the time dependent inlet water depth is only function of the flow rate overtopped and the weir length; the execution of numerical methods to invert Swamee’s formula (1988, [46]) is avoided; the relative error associated with a constant discharge coefficient is minimized.

* 1. SPH balance equations for rigid body transport

This section relies on Amicarelli et al. (2015, [2]), Amicarelli et al. (2020, [47]), the RSE developments in Paggi et al. (2021, [48]) and Amicarelli et al. (2021, [49]), whose reading is suggested for further details.

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

|  |  |
| --- | --- |
|  | (6.51) |

The first formula of (6.51) represents the balance equation of the momentum of a solid rigid body, which translates with no rotation around its barycentre. The second formula of (6.51) represents the balance equation of the momentum of a solid rigid body which rotates with no translation of its barycentre; it is the balance equation of the body angular momentum. The following kinematics formulae are used for time integrating the body velocity and angular velocity to obtain the barycentre position and the orientation of the body:

|  |  |
| --- | --- |
|  | (6.52) |

where ** is the vector of the angles lying between the body axes and the global reference system.

Here the subscript “*B*” refers to a generic computational body and “*CM*” to its centre of mass.

The first formula of (6.51) represents the balance equations for the momentum.*FTOT* is the global/resultant force acting on the solid body. The last formula of (6.51) expresses the balance equation of the body angular momentum, where ** denotes the angular velocity of the generic body.. *MTOT* represents the associated torque acting on the body and the matrix of the moments of inertia of the computational body (Einstein’s notation works for the subscript “*l*”):

|  |  |
| --- | --- |
| , | (6.53) |

In this sub-section, *r* implicitly represents the relative distance from the body centre of mass.

One considers the final formulation of the second formula of (6.51). When the first/second term in the squared parentheses is negligible, then the resulting rotation is called torque-free/torque-induced precession. The minor effect of the torque-free term on a torque-induced precession is called nutation.

Torque-free precession occurs when the integral of the external forces, internal forces and external torques are null, but the integral of the internal torques is non-null. The internal torques are induced by the internal forces. These normal and shear stresses within the body material are necessary to permit the centripetal accelerations of the material points of the body.

Torque-free precessions are prevented if at least one of the following requirements is satisfied: body symmetry around the current rotation axis; body symmetry around the plane normal to the current rotation axis and passing for the body barycentre; null angular velocity.

The inverse of the matrix of the moment of inertia acts like a rotation matrix for the torque in the second formula of (6.51) and thus can induce a change in the rotation axis in the presence of an external torque. Torque-induced precession is prevented if at least one of the following conditions is satisfied: body symmetry around the current rotation axis; body symmetry around the plane normal to the current rotation axis and passing for the body barycentre; null torque.

In order to solve the system (6.51), we need to model the global force and torque, as described in the following. The resultant force is composed of several terms:

|  |  |
| --- | --- |
|  | (6.54) |

*G* represents the gravity force, whereas *PF* and *TF* the vector sums of the pressure and shear forces provided by the fluid. Analogously, *PS* and *TS* 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.6.9).

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:

|  |  |
| --- | --- |
|  | (6.55) |

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 *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, [9]), here implemented and adapted as described in Sec.6.6. Further, the solid-solid interaction term (*Ps*) is presented in Sec.6.11.

On the other hand, the torque in (6.51) is discretized as the summation of each vector product between the relative position *rs*, of a surface body particle with respect to the body centre of mass, and the corresponding total particle force:

|  |  |
| --- | --- |
|  | (6.56) |

Time integration of (6.51) 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:

|  |  |
| --- | --- |
|  | (6.57) |

Finally, the model updates the body particle normal vectors and absolute positions, according to the following kinematics formulas (is the increment in the body rotation angle during the on-going time step and *Rij* is the body rotation matrix):

|  |  |
| --- | --- |
| ,, | (6.58) |

At the moment, the following simplifications are assumed: the drag force *TF* is not represented; each body is given in input a uniform density.

* 1. Improving 3D rotations

The equations:

|  |  |
| --- | --- |
|  | (6.59) |

are not exact and represent a limit when working with the a rotation matrix based on Euler angles: the first equation of (6.59) 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:

|  |  |
| --- | --- |
|  | (6.60) |

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 R). This approach is also called “matrix to axis-angle” and is used by Rodrigues formula to define the rotation matrix:

|  |  |
| --- | --- |
|  | (6.61) |

where the rotation axis is represented by the unit vector *nR*.

The cross-product matrix is a skew symmetric matrix:

|  |  |
| --- | --- |
|  | (6.62) |

The expansion of *Rij* reads:

|  |  |
| --- | --- |
|  | (6.63) |

In 2D, the Rodrigues’ rotation matrix is equal to Euler’s rotation matrix.

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

|  |  |
| --- | --- |
|  | (6.64) |

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.6.6).

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:

|  |  |
| --- | --- |
|  | (6.65) |

The cross product is not sufficient to distinguish between *R* and (**-*R*):

|  |  |
| --- | --- |
|  | (6.66) |

whereas the following formula cannot be used:

|  |  |
| --- | --- |
|  | (6.67) |

However, consider that *R* from the cross product automatically and correctly provides an angle between 0 and ** (sin(*R*) is non-negative).

The sine and cosine functions of the rotation angle read:

|  |  |
| --- | --- |
|  | (6.68) |

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

|  |  |
| --- | --- |
|  | (6.69) |

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.

* 1. Reaction forces

Following the superposition principle of effects, the reaction forces of a generic body dynamics configuration are computed as vector sums of the reaction forces associated with two different configurations: the associated hyperstatic configuration and the associated dynamic configuration in the absence of other forces. Under hyperstatic conditions, the system of governing equations concern 6 balance equations (associated with the 6 degrees of freedom of the 3D body) and n compatibility equations (representing the deformation consistency among different parts of the same body).

The contributions to the reaction forces in the dynamic configuration should superpose the effects of the following dynamics mechanisms: 1D impingements in any direction in 3D; 1D sliding in any direction in 3D; pitch; roll; yaw.

SPHERA treats hyperstatic conditions as a dynamic configuration with spurious negligible oscillations: the relative velocities between the solid elements allow to dynamically develop the reactions necessary to maintain the hyperstatic configuration. The normal reaction in SPHERA is formally null, if the normal velocity is zero. However, according to Monaghan (2005, [26]), the normal forces of the boundary force particles dynamically restore the normal reaction forces. This approach is similar to a simple spring-model approach, but the reactions are triggered by the relative velocities, not the relative positions. In any case, these relative velocities and the associated displacements should be negligible with respect to the spatial resolution of the simulation.

SPHERA directly treats 1D impingements in 3D and 1D sliding mechanisms in 3D. SPHERA models the boundary reaction forces associated with the pitch and roll mechanisms thanks to the particle-based assessment of both the impact velocity and the sliding friction limiter. Instead, SPHERA does not represent the boundary reaction forces to the yaw mechanism as the simulated sliding friction force is based on a body approach. Treating the sliding friction force as the vector sum of particle-based friction forces would allow in future to treat the reaction forces associated with the yaw dynamics mechanism.

Contrarily to the normal reaction force, which is exerted gradually during the particle interaction, the sliding friction force is instantaneously applied at its maximum value as the particle interaction begins. This is correct in the frame of rigid bodies and point contacts. On the other hand, the formulation for the normal reaction guarantees the correct representation of the bouncing velocity, no matter about the maximum force. This means that the maximum normal reaction could be reproduced either instantaneously or gradually for rigid bodies. However, the current modelling of sliding friction might represent a systematic overestimation especially in case of curved surfaces (e.g., a wheel rolling on a flat plate) due to the artificial increase of the contact surface and the contact time (a point contact is replaced by a distributed contact). Possible solutions are: the reduction of the influence region just for sliding friction; an alternative gradual activation of sliding friction as if the solids were deformable. Nonetheless, curved surfaces are not currently permitted when initializing a rigid body.

* 1. Sliding friction force

The formulation of the current section refers to Amontons’ law applied to a single body interacting with multiple frontiers. It might be revised for a single body interacting with multiple bodies. The current code version assumes a unique friction angle applies to all the body-frontier interactions.

The overall normal of the neighbouring frontiers is the unit vector aligned with the vector sum of the neighbouring normals. In case of need, this sum might be weighted in a proper manner.

* + 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:

|  |  |
| --- | --- |
|  | (6.70) |

where ** is the sliding friction angle, the subscript “*dry*” refers to dry conditions, *sw* is the unit vector parallel to the local frontiers and with the same direction as the velocity vector of the body barycentre (projected on the local DEM).

The coefficient of sliding friction (*sf*) reads:

|  |  |
| --- | --- |
|  | (6.71) |

The absolute value for the exact formulation reads:

|  |  |
| --- | --- |
|  | (6.72) |

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

|  |  |
| --- | --- |
|  | (6.73) |

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:

|  |  |
| --- | --- |
|  | (6.74) |

where  is the maximum particle velocity (all over the computational body) tangential to the interacting local frontier elements.

* + 1. Aerial stage (negative value for the input friction angle or body-body interactions)

The sliding friction force (*Ts*: 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:

|  |  |
| --- | --- |
|  | (6.75) |

Its absolute value is:

|  |  |
| --- | --- |
|  | (6.76) |

where *DTM* is the slope angle and *k* the unit vector aligned with the vertical axis:

|  |  |
| --- | --- |
|  | (6.77) |

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.

|  |  |
| --- | --- |
|  | (6.78) |

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

Under this configuration, normal reaction does not provide any torque (nevertheless the limiter for the sliding friction force depends on the velocity of the solid particles interacting with the frontiers).

* + 1. Submerged stage

The sliding friction provided by a solid boundary to a wet body assumes the following form:

|  |  |
| --- | --- |
|  | (6.79) |

where *sw* 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 frontier surface). If this velocity is null, then the friction force is zero.

The maximum absolute value for the sliding friction reads:

|  |  |
| --- | --- |
|  | (6.80) |

where ** is here the sliding friction angle (the current code version assumes a unique value for all the body-frontier interactions).

In order to prevent a dissipative force to increase the energy of the system, the following approximated limiter applies:

|  |  |
| --- | --- |
|  | (6.81) |

where is the maximum particle velocity (all over the portion of the computational body within the boundary-body interaction region) tangential to the interacting local frontier elements. This limiter is excessive as it avoids any surface velocity reversal.

The sliding friction force contributes to the global torque. The application point of the friction force for any solid body is computed as the average position of the body particles involved in the "body particle - frontier" interactions. Each interaction has the same weight in this averaging process (every particle might be considered more than once).

The following hypotheses are assumed: all the surface body particles in contact with wet boundaries have a defined pressure; all the body-boundary interfaces are hydraulically connected with the fluid; normal reaction forces (with bodies and boundaries, included the current boundaries providing friction) are not considered to assess the sliding force.

All the SPH fluid particles within any solid body at the beginning of the simulation (initial conditions for the fluid particle positions in case of Fluid - Structure interactions), due to design, are removed before the simulation time starts.

* 1. Body-boundary normal reaction force under sliding (at null normal velocity)
     1. Aerial stage

The exact formulation is correctly represented:

|  |  |
| --- | --- |
|  | (6.82) |

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.

* + 1. Submerged stage

The normal reaction is formally null.

|  |  |
| --- | --- |
|  | (6.83) |

According to Monaghan (2005, [26]), 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).

|  |  |
| --- | --- |
|  | (6.84) |

* 1. Fluid-body interaction terms

This section relies on Amicarelli et al. (2015, [2]), Amicarelli et al. (2020, [47]), the RSE developments in Paggi et al. (2021, [48]) and Amicarelli et al. (2021, [49]), whose reading is suggested for further details.

The fluid-body interaction terms rely on the boundary technique introduced by Adami et al. (2012, [9]), implemented and adapted for free-slip conditions (Amicarelli et al., 2015, [2]). 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 (6.6) is here represented by all the body particles inside the kernel support of the computational fluid particle. Further, Adami et al. (2012, [9]) 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 (6.19), used to treat solid frontiers (free-slip conditions):

|  |  |
| --- | --- |
|  | (6.85) |

Modelling the shear stress gradient term in the Navier-Stokes equation requires an alternative formulation to (6.85). The discretization of this term, which is coherent with the definition of the inter-particle velocity under no-slip conditions, assumes the following form:

|  |  |
| --- | --- |
|  | (6.86) |

Analogously, the fluid-body interaction term in the momentum equation (6.20) assumes the form:

|  |  |
| --- | --- |
|  | (6.87) |

where the subscript “*s0*” denotes a generic solid-fluid inter-particle interaction. The first term in the RHS of (6.87) is reported by Adami et al. (2012, [9]) whereas the last term will be discussed later on this section.

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):

|  |  |
| --- | --- |
|  | (6.88) |

The “wall” acceleration at the position of a generic body particle can then be derived by linearizing (6.88). This depends on the particular computational fluid particle “*0*” we are considering, so that we can refer to the interaction subscript “*s,0*”:

|  |  |
| --- | --- |
|  | (6.89) |

where *dln* is a vectorial length element along the centreline of the two particles, projected along the wall element normal.

One applies a SPH interpolation over all the pressure values estimated according to Adami et al. (2012, [9]) to derive a unique pressure value for a body particle:

|  |  |
| --- | --- |
|  | (6.90) |

In case of no-slip conditions, in-built motion interests all the acceleration components and the shear stress gradient term is involved; (6.88) is replaced by the following expression:

|  |  |
| --- | --- |
|  | (6.91) |

Analogously to (6.89), one obtains:

|  |  |
| --- | --- |
|  | (6.92) |

In case of no-slip conditions, (6.90) is replaced by the following expression:

|  |  |
| --- | --- |
|  | (6.93) |

which is at the moment approximated by this formula:

|  |  |
| --- | --- |
|  | (6.94) |

The pressure value of (6.90) or (6.94) is finally used in (6.87).

Only a minority of the body particles represents the body surface, but we also need many inner body particles to estimate ps. 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 *ns* in (6.93) -“*s0*” interaction-. If the on-going body particle “*s*” belongs to the body surface, then it is immediately considered as representative. Otherwise, the fluid particle “*0*” isolates its visible neighbouring surface body particles. Visibility is assessed considering the sign of the projection of the inter-particle distance on the body particle normal. The visible neighbour, which is the closest to the joining segment of particles “*0*” and “*s*”, is then selected. This particle provides the normal “*ns*” for the fluid-solid particle interaction “*s0*” in (6.93).

The assumption (6.88) 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 (6.93). Nevertheless, we prefer defining a maximum threshold for |*as*|, here 10*g*.

The search for the fluid-body interaction normals is only carried out in case of free-slip conditions.

An improvement on the searching algorithm for the fluid-body interaction normals provides more accurate results and a computational time reduction with respect to Amicarelli et al. (2015, [2]), as described in the following. In that paper, the searching volume V*s* was defined as the parallelepiped described by the coordinates of the interacting particles:

|  |  |
| --- | --- |
|  | (6.95) |

This caused some issues (i.e. searching area close to zero or no accurate result for the searching algorithm) under the following configurations: interacting particles having almost the same value for one coordinate; surface solid particle being one of the interacting particles. To solve these issues, the new searching algorithm extends the searching region, which is here defined as the intersection between the two identical spheres centred at the interacting particle positions and radius equal to the inter-particle distance (Figure 6.1). The new definition of the searching volume reads:

|  |  |
| --- | --- |
|  | (6.96) |

where *d*(…) (m) is here the distance between two points.

|  |
| --- |
|  |

Figure 6.1. Searching volume in fluid-body interactions to define the interaction normal to the fluid-body interface.

The search for the fluid-body interaction normals now explicitly reports possible fluid-solid mass penetrations (just in case of free-slip conditions). In case of penetration (allowed for rough spatial resolutions), the supplementary search for a formal normal now only involves surface body particles.

The modified algorithm extends the searching region, which is defined as the intersection between the two identical spheres centred at the interacting particle positions and with radius equal to the inter-particle distance.

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

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

|  |  |
| --- | --- |
|  | (6.97) |

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

|  |  |
| --- | --- |
|  | (6.98) |

The maximum and minimum operators in the Right Hand Side of (6.98) apply to the whole domain.

Modelling the shear stress gradient term in the Navier-Stokes equation requires the introduction of an additional fluid-body coupling term in the Right Hand Side (RHS) of the fluid momentum equation. It is the last term of (6.87). The discretization of this coupling term, which takes into account the shear stress exchanged at the fluid-body interface, is derived in coherence with the SPH particle approximation in the inner domain.

The solid particle volume is computed as follows:

|  |  |
| --- | --- |
|  | (6.99) |

The inter-particle velocity *us0* represents the field of the fluid velocity virtually reconstructed within the portion of the kernel support, which is truncated by the solid body.

The component of the inter-particle velocity, which is normal to the interface, guarantees no mass penetration (symmetric conditions) at the interface:

|  |  |
| --- | --- |
|  | (6.100) |

Under no-slip conditions, the component of the inter-particle velocity, which is tangential (subscript “*T*”) to the interface, guarantees a uniform velocity gradient around the interface (if its position is assumed to be the average of the positions of the interacting particles):

|  |  |
| --- | --- |
|  | (6.101) |

where the unit vector *t* is tangential to the interface.

Under no-slip conditions, the difference between the inter-particle velocity and the fluid particle velocity in Eq. (51) is expressed as follows:

|  |  |
| --- | --- |
|  | (6.102) |

* 1. Solid-solid interaction terms

This section relies on Amicarelli et al. (2015, [2]), Amicarelli et al. (2020, [47]), and Amicarelli et al. (2021, [49]), whose reading is suggested for further details.

The solid-solid interaction term in (6.55) -*Ps*- 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 *Ps*, as explained hereafter.

The “boundary force particle” method of Monaghan (2005, [26]) defines repulsive forces to represent a conservative full elastic impingement between two SPH interacting particles (of any medium). In particular, the acceleration  of particle “*j*”, due to the impingement with particle “*k*”, is aligned with the inter-particle distance *r* and inversely proportional to its absolute value *r*:

|  |  |
| --- | --- |
|  | (6.103) |

The analytic function *fbfp* is symmetric with respect to the impact point. The dependence of (6.103) on the particle masses allows conserving both global momentum  and kinetic energy (one may notice that  and ). 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 *Ps*, which represents the impingements between a generic computational body (“*B*”) and all its neighbouring bodies (“*K*”) and frontiers (“*K\**”).

*Ps* is decomposed in elementary 2-body (*PBK*) and body-frontier (*PBK\**) interactions:

|  |  |
| --- | --- |
|  | (6.104) |

where *NBK\** is the number of neighbouring boundary frontiers.

Adopting the same principles of the boundary force particle method, *PBK* 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*”:

|  |  |
| --- | --- |
|  | (6.105) |

The components of the inter-particle relative distance, *rpar* and *rper*, are parallel and perpendicular to the neighbour normal, respectively. The term within brackets in (6.105) 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 (*dxs* is the size of the body particles). The weighting function ** is expressed according to Monaghan (2005, [26]) and depends on *q*= *rjk*/*h*:

|  |  |
| --- | --- |
|  | (6.106) |

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 , which replaces the term (0.1*c*) in the formulation of Monaghan (2005, [26]) 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:

|  |  |
| --- | --- |
|  | (6.107) |

where *t0* 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, (6.107) 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, (6.105) introduces the coefficient *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 *I*, (6.105) would drastically under-estimate the impingement forces if the whole mass of the bodies did not lie within the impact zone (of depth 2*h*). To avoid this shortcoming, a formulation for *I* is presented hereafter. Consider the absolute value of the impingement force *Ps* as a function of the global parameters of the bodies, instead of the particle values. This second formulation for *PBK* is denoted as follows:

|  |  |
| --- | --- |
|  | (6.108) |

The inter-body velocity impact 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, 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 *I*, by equalling *PBK* to *PBK’*:

|  |  |
| --- | --- |
|  | (6.109) |

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

|  |  |
| --- | --- |
|  | (6.110) |

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 *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 (6.109) to model a body-body impact. In this case, for example, a definition for the direction of *Ps’* 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):

|  |  |
| --- | --- |
|  | (6.111) |

The body-boundary forces are normalized by the number of neighbouring frontiers for a given body. This number is approximated by the maximum number of neighbouring frontiers of a single body particle belonging to the body.

* 1. Normal restitution coefficient

The simulated normal restitution coefficient (*Rn*) 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 *Rn*<1. The equivalent value of the restitution coefficient might be estimated a posteriori.

# The scheme for dense granular flows

This section describes the mathematical and numerical models for dense granular flows (Amicarelli et al., 2017, [1], Sec.7.1) and its possible speed-up by means of a 2-interface 3D erosion scheme (Amicarelli & Agate, 2014, [50], Sec.7.2), which extends the (1-interface) 2D erosion scheme of Manenti et al. (2012, [4], 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).

* 1. Mixture model for dense granular flows

This whole section constantly refers to Amicarelli et al. (2017, [1]), 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; Armstrong et al., 2010, [51]) 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 (Armstrong et al., 2010, [51]):

|  |  |
| --- | --- |
|  | (7.1) |

where ** (kg×m-3) is density, ** the phase volume fraction, *u* (m×s-1) the velocity vector, *t* represents time and *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 (Armstrong et al., 2010, [51]):

|  |  |
| --- | --- |
|  | (7.2) |

The volume balance equation assumes the following form:

|  |  |
| --- | --- |
|  | (7.3) |

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

|  |  |
| --- | --- |
|  | (7.4) |

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

|  |  |
| --- | --- |
|  | (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; Armstrong et al., 2010, [51]):

|  |  |
| --- | --- |
|  | (7.6) |

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

|  |  |
| --- | --- |
|  | (7.7) |

Following the multi-phase approach of Colagrossi & Landrini (2003, [52]), the SPH approximation of (7.7) can be expressed as follows:

|  |  |
| --- | --- |
|  | (7.8) |

where *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 Di Monaco et al. (2011, [5]) and represents the effects of wall frontiers.

This approximation of the continuity equation does not violate the mass conservation as the particle volume is undetermined.

Considering the KTGF, the momentum equation for the fluid phase can be expressed as follows (Armstrong et al., 2010, [51]):

|  |  |
| --- | --- |
|  | (7.9) |

where *ij* (Pa) is the deviatoric (or shear) stress tensor, *g* (m×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 *Kgs* (kg×m-3×s-1).

The momentum equation for the solid phase can be expressed as follows (Armstrong et al., 2010, [51]):

|  |  |
| --- | --- |
|  | (7.10) |

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

|  |  |
| --- | --- |
|  | (7.11) |

Considering the assumption on conservative SPH particles, the shear stress gradient term of the fluid phase can be expressed as follows (Armstrong et al., 2010, [51]):

|  |  |
| --- | --- |
|  | (7.12) |

where ** (Pa×s) represents viscosity. Under the hypothesis of plain strain, the shear stress gradient term is represented by Schaeffer (1987, [53]; visco-plastic model for dry granular material based on internal friction), by means of a parameter, which KTFG names frictional viscosity, as described in the following.

The pressure of the solid phase is treated as follows:

|  |  |
| --- | --- |
|  | (7.13) |

where (Pa) are the effective stresses along the principal directions and (Pa) is the mean effective stress (Schaeffer, 1987, [53], 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 (Schaeffer, 1987, [53]):

|  |  |
| --- | --- |
|  | (7.14) |

where ** (rad) is the internal friction angle, *eij* (s-1) is the strain-rate tensor and *I2*(*eij*) (s-2) represents its quadratic 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.(7.14) can be rearranged as follows:

|  |  |
| --- | --- |
|  | (7.15) |

The strain-rate tensor reads:

|  |  |
| --- | --- |
|  | (7.16) |

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

|  |  |
| --- | --- |
|  | (7.17) |

Renormalization (Randles & Libertsky, 1996, [54]) applies to the velocity derivatives in (7.17), only for 2D simulations:

|  |  |
| --- | --- |
|  | (7.18) |

The KTGF introduces the following definition for the frictional viscosity *fr* (Pa×s):

|  |  |
| --- | --- |
|  | (7.19) |

to obtain:

|  |  |
| --- | --- |
|  | (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 fr, Eq.(7.20) provides:

|  |  |
| --- | --- |
|  | (7.21) |

which is valid in the “packing limit” of the KTGF (i.e. for *s* close enough to the value of 0.59, which is the maximum attainable volume fraction for a sheared inelastic hard sphere fluid, Kumaran, 2015, [55]). The model then defines the mixture total pressure *p* (Pa) as:

|  |  |
| --- | --- |
|  | (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,  is computed as the difference between the total pressure and the fluid pressure:

|  |  |
| --- | --- |
|  | (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:

|  |  |
| --- | --- |
|  | (7.24) |

where *εs,p* = ca.0.59 and H(x) is the Heaviside step function:

|  |  |
| --- | --- |
|  | (7.25) |

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

|  |  |
| --- | --- |
|  | (7.26) |

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

|  |  |
| --- | --- |
|  | (7.27) |

where the mixture viscosity ** is finally defined as:

|  |  |
| --- | --- |
|  | (7.28) |

and **(m2×s-1) is the mixture kinematic viscosity.

Following the multi-phase approach of Colagrossi & Landrini (2003, [52]), with the boundary treatment method proposed by Di Monaco et al. (2011, [5]), the SPH approximation of (7.27) becomes:

|  |  |
| --- | --- |
|  | (7.29) |

where *r* (m) is the distance between two interacting particles*,* whereas *M* (m2×s-1) stands for the artificial viscosity (Monaghan, 1992, [56]). The boundary value *uSA* (m×s-1) of the velocity in the external portion *Vh’* (m3) of the kernel support is assigned according to Di Monaco et al. (2011, [5]). So far, the last term, representing the bottom drag, has been validated in 2D.

The inner pressure gradient term in (7.29) is conservative provided that the particle mass is homogeneous because:

|  |  |
| --- | --- |
|  | (7.30) |

The inner viscous shear stress term in (7.29) is conservative provided that the particle density and the particle kinematic viscosity are homogeneous. The inner artificial viscosity term in (7.29) is conservative provided that the particle artificial viscosity and the particle density are homogeneous.

The artificial viscosity is always activated, both for approaching and separating particles (the latter configuration was not considered in Di Monaco et al., 2011, [5]):

|  |  |
| --- | --- |
|  | (7.31) |

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, i.e., fully saturated soil and dry soil:

|  |  |
| --- | --- |
|  | (7.32) |

where the subscript “*blt-top*” refers to the top of the bed-load transport layer (or the layer of saturated material). The water content of a generic SPH mixture particle is constant all over the simulation, but saturation conditions depend on the SPH particle. This is the default Lagrangian saturation scheme. An alternative Eulerian saturation scheme, based on tuning parameters, is also available. It is described at the end of this section.

Eq.(7.32) assumes a 1D filtration flow parallel to the slope of the granular material. This simplifying hypothesis is still consistent with SPH conservative particles; radis the topographic angle at the top of the bed-load transport layer and lies between the local interface normal *nblt-top* and the vertical:

|  |  |
| --- | --- |
|  | (7.33) |

The angle limiter in (7.33) allows one to assign null *pf* 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”):

|  |  |
| --- | --- |
|  | (7.34) |

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% (Monaghan, 2005, [26]). 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 (*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, *max* does not influence even the computational time (the Courant-Friedrichs-Lewy *CFL* numbercriterion dominates over the viscous term criterion), if the following condition is satisfied:

|  |  |
| --- | --- |
|  | (7.35) |

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, [1]), 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 (*cref*) 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:

|  |  |
| --- | --- |
|  | (7.36) |

The sound speed (*cref*) should be at least 10 times (Monaghan, 2005, [26] assumed *Cp,max*=2; 5 times *Cp,max* for higher pressure peaks) higher than the maximum velocity in the fluid (WC approach). This position (constant *AWC* equal to 10) provides a maximum relative error on density of 1%, whereas the assumption *AWC*=4.5 increases the density relative error to 5% (Monaghan, 2005, [26]).

The velocity scale *Uscale* reads:

|  |  |
| --- | --- |
|  | (7.37) |

where *Ymax* is the maximum water depth and  is the maximum absolute value of velocity (the maxima operate both over the whole simulated time and the whole 3D domain space).

In order to impose the initial pressure field for granular flows, a dynamic setup of hydrostatic conditions is suggested within the elasto-plastic layer because “mono-phase” hydrostatic conditions are instantaneously imposed.

One notices that *εs,p* is used in the balance equations of this model, but the value of *εs* at the initial conditions should respect the mass conservation of the mixture, which is not necessarily under the packing limit at the beginning of the simulations. Sometimes, the void ratio *ev* is available instead of the phase volume fractions. The mixture porosity is related to the void ratio by means of the following expression:

|  |  |
| --- | --- |
|  | (7.38) |

Under these circumstances, the initial density of the solid phase is assessed as function of the mixture specific weight, the porosity, the fluid density and the gravity acceleration:

|  |  |
| --- | --- |
|  | (7.39) |

Further, a criterion is applied to detect the Landslide Sliding Surfaces LSSs.

In the following, an alternative Eulerian saturation condition is reported.

Two supportive and user-defined parameters are needed: the time with minimum saturation (*tmin,sat*, s) and the time with maximum saturation (*tmax,sat*, s). They represent two arbitrary times, which identify a relative minimum and a following relative maximum in soil saturation. When *t*≤*tmin,sat*, there is always a phreatic zone where a free surface (“*FS*”) is detected along the vertical (and dry soil elsewhere). When *t*≥ *tmin,sat*, the saturation zones (phreatic or dry) are frozen as they were at *tmax,sat*. For intermediate times and zones, the infiltration front advances linearly in time. Details are provided in (28), (29) and (30). For , the following zones are defined:

|  |  |
| --- | --- |
|  | (7.40) |

For , the following zones are defined:

|  |  |
| --- | --- |
|  | (7.41) |

For, the following zones are defined:

|  |  |
| --- | --- |
|  | (7.42) |

In the phreatic zone the fluid pressure is expressed by (7.32), whereas in the infiltration zone it reads:

|  |  |
| --- | --- |
|  | (7.43) |

when the height of the infiltration front is linear in time:

|  |  |
| --- | --- |
|  | (7.44) |

* 1. 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, [4]), 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, [57]) reads:

|  |  |
| --- | --- |
|  | (7.45) |

where *c* is Shields parameter and *Re\** is the grain Reynolds number:

|  |  |
| --- | --- |
|  | (7.46) |

where ks is a roughness length scale with ks=3d50 as an approximation for ks=3d90. It is a local erosion criterion, thus roughness is computed locally, instead of adopting a formulation for bed forms (van Rijn, 1982, [36]).

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 *z0*, according to the formula of Manenti et al. (2012, [4]) and those associated to the similarity theory or the SNBL:

|  |  |
| --- | --- |
|  | (7.47) |

where *kv* 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:

|  |  |
| --- | --- |
|  | (7.48) |

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 *z0*, which is in turn function of *u\**-), then *Re\** and *c*. Shield parameter is computed:

|  |  |
| --- | --- |
|  | (7.49) |

and compared with *c*. The erosion criterion is satisfied if *q*≥*q*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*, which is defined as follows:

|  |  |
| --- | --- |
|  | (7.50) |

*k*is always non-negative and smaller than (or equal to) its 2D value *k***. In fact, if the slope angle transversal to the main flow direction (**) is not null, erosion is enhanced. Further, in the presence of a bed with a locally ascendant slope (**<0), *k* can be higher than the unity. In this case, (7.50) can possibly provide a second non-physical solution, with *k*, 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:

|  |  |
| --- | --- |
|  | (7.51) |

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, **only depends on the direction of the velocity vector of the closest particle (3D definition):

|  |  |
| --- | --- |
|  | (7.52) |

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

|  |  |
| --- | --- |
|  | (7.53) |

The transversal slope angle ** is defined as:

|  |  |
| --- | --- |
|  | (7.54) |

The unity vector *n2* represents the bi-normal to the fluid particle trajectory and is independent on the sign of **.

The value of *k* is a solution of the quadratic equation of Seminara et al. (2002, [58]):

|  |  |
| --- | --- |
|  | (7.55) |

In the presence of two admissible roots, the model chooses the closest to *k*, provided ; in the absence of roots, the model assumes *k*=*k*

The drag coefficient *CD* is approximated by the formula of Morrison (2013, [59]) for a fluid flow around a sphere:

|  |  |
| --- | --- |
|  | (7.56) |

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

|  |  |
| --- | --- |
|  | (7.57) |

with  equal to the absolute value of velocity at the closest particle and *d50* representing the 50-th percentile of the particle-size distribution of the soil.

In this context, the lift is assumes the form:

|  |  |
| --- | --- |
|  | (7.58) |

where *zint* is the interface height. A formula for the lift coefficient is derived, by interpolating the experimental data of Seminara et al. (2002, [58]):

|  |  |
| --- | --- |
|  | (7.59) |

with *CL* 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:

|  |  |
| --- | --- |
|  | (7.60) |

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.

* 1. A simplified approach for soil liquefaction

This section refers to Amicarelli et al. (2016, [60]).

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:

|  |  |
| --- | --- |
|  | (7.61) |

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

In case the liquefaction time (*tliq*) is known and *fliq* is approximately linear, (7.61) becomes:

|  |  |
| --- | --- |
|  | (7.62) |

where *tq0* and *tqf* are the quake starting and ending times.

# The DB-SPH boundary treatment scheme

This section describes the “Discrete Boundary” (DB) - SPH method for boundary treatment (Amicarelli et al., 2013, [3]). 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).

* 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:

|  |  |
| --- | --- |
|  | (8.1) |

In (8.1), the volume integral in (6.3) 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 (*Vh*). Eq.(8.1) is normalized by the integral Shepard coefficient (**) to obtain this further definition:

|  |  |
| --- | --- |
|  | (8.2) |

** varies as function of the involved computational particle “*0*”. Provided fixed time and position, **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 **. 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, [13]) 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, [13]).

The “semi-particles” represent special fluid particles, which are smallest than the (inner) fluid particles. Each semi-particle is associated to a surface wall element. Semi-particle positions are formally located at the solid frontiers of the fluid domain, but the volumes of the semi-particles completely lie in the inner domain and touch the solid boundaries. The union of the semi-particle volumes represents a thin film of fluid, which is a buffer zone between the inner domain (filled with computational particles) and the wall frontiers. The film depth is smaller than the characteristic length of the fluid particles (*dx*).

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  (3D) or length (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 . 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 *p0*≠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, [13]):

|  |  |
| --- | --- |
|  | (8.3) |

where the kernel is normalized by a corrected estimation of the integral Shepard coefficient.

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

|  |  |
| --- | --- |
|  | (8.4) |

The integral Shepard coefficient is replaced with the discrete Shepard coefficient at the free surface, which is numerically defined where . ** 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 ** would imply the expensive estimation of 3D analytical integrals. Instead, the present model follows the procedure of Ferrand et al. (2013, [13]), as synthesized by (8.5) and (8.6). Consider the Lagrangian derivative of **:

|  |  |
| --- | --- |
|  | (8.5) |

The initial values of ** are approximately provided by the associated values of **, as the model exactly assigned the initial values of the fluid particle volumes:

|  |  |
| --- | --- |
|  | (8.6) |

The integral Shepard coefficient ** 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 ** 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 ** (and then of **) 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 **initialization.
   1. 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, [20]), which allows wall pressure being approximately compatible with the 3D pressure and velocity fields in the inner domain (constrained to the frontier kinematics).

The definition of the initial conditions (“*L*”, “Left”) of the 1D LPRS are described by means of a first order spatial reconstruction scheme.

For each interaction (“*0a*”) between a surface element (“*a*”) and a fluid particle (“*0*”), the LPRS initial conditions are defined at the position of the wall element. Here the model estimates density and the velocity components, by means of a first-order spatial reconstruction scheme around he computational particle (f alternatively refers to density and every velocity component):

|  |  |
| --- | --- |
|  | (8.7) |

The velocity vector is projected along the normal of the surface wall element to obtain *un*.

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):

|  |  |
| --- | --- |
|  | (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:

|  |  |
| --- | --- |
|  | (8.9) |

* 1. Semi-particle volume

The volume of a semi-particle *ws* is defined for complex and generic geometries:

|  |  |
| --- | --- |
|  | (8.10) |

where *kd* is the semi-particle shape coefficient, *kw* is the semi-particle depth coefficient and *xw* (m) is the size of the surface wall elements.

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 the number *D* of the spatial dimensions. From those exact values, the following interpolating formula is obtained:

|  |  |
| --- | --- |
|  | (8.11) |

where *naf* represents the number of adjacent elements detected and the subscript “*i*” here represents the generic adjacent element. The angles *ai* (rad) lie between a generic surface element and each of the adjacent elements. According to the adopted formalism, the model needs to add p at the original assessment if the angle between the element normal vectors varies between -p/2 and p/2. The reference formula for *ai* reads:

|  |  |
| --- | --- |
|  | (8.12) |

Further details are available in [49].

* 1. 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 searches these particles within an influence sphere of characteristic length , where *Lc* 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.

* 1. Shear stress boundary terms for mobile wall frontiers

The shear stress boundary terms of Ferrand et al. (2013, [13]) are adapted to treat mobile frontiers and integrated in the boundary treatment scheme of Amicarelli et al. (2013, [3]).

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:

|  |  |
| --- | --- |
|  | (8.13) |

where  (m/s) is the velocity vector, is the gradient operator, *r* (kg/m3) is density, *W* (m-3) is the kernel function, ** is the integral Shepard coefficient, *m* (Pa×s) is the dynamic viscosity, *n* is the normal of the surface boundary element with area *wa* (m2) and the subscript “*0*” represents the computational particle whose momentum equation is under assessment.

Applying the derivation chain rule and assuming that the normal slowly varies with time, it follows:

|  |  |
| --- | --- |
|  | (8.14) |

where  (m) is the Cartesian position and Einstein’s notation applies to the subscript “*k*”. 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:

|  |  |
| --- | --- |
|  | (8.15) |

where *d* (m) is the inter-element distance. Assuming a continuous shear stress, the last term of (8.14) is expressed as a SPH approximation normalized by the discrete Shepard coefficient (**):

|  |  |
| --- | --- |
|  | (8.16) |

Thus, the shear stress boundary term due to the surface wall elements is:

|  |  |
| --- | --- |
|  | (8.17) |

The analogous term due to the semi-particles “*s*” (i.e., fluid elements along the mobile boundaries associated with the surface boundary elements “*a*”, Amicarelli et al., 2013, [3]), reads:

|  |  |
| --- | --- |
|  | (8.18) |

where n (m2/s) is the kinematic viscosity, *m* (kg) is the SPH particle mass and *r* (m) is the distance from the barycentre of the SPH kernel support.

# Time integration schemes (Leapfrog, Euler, Heun)

Time integration for both fluid and solid body particles is ruled by a second-order Leapfrog scheme, as described in Amicarelli et al. (2015, [2]) and Di Monaco et al. (2011, [5]):

|  |  |
| --- | --- |
|  | (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:

|  |  |
| --- | --- |
|  | (9.2) |

The scheme above can be rearranged in the following form:

|  |  |
| --- | --- |
|  | (9.3) |

where the subscripts here represent the time step ID.

RK2 assumes the following form:

|  |  |
| --- | --- |
|  | (9.4) |

This 2-stage formulation implies 2 stages (sub-loops) for each time step. During the first stage, the temporary value *fRK1,i+1* is computed. During the second stage, the time step value *fRK2,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:

|  |  |
| --- | --- |
|  | (9.5) |

where *dt* (s) is the time step duration and *CFL* the Courant-Friedrichs-Lewy number. Following Adami et al. (2021, [9]), the viscous term stability parameter is set to *C*=0.05.

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

|  |  |
| --- | --- |
|  | (9.6) |

where *D* is thedomain 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):

|  |  |
| --- | --- |
|  | (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):

|  |  |
| --- | --- |
|  | (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):

|  |  |
| --- | --- |
|  | (9.9) |

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

|  |  |
| --- | --- |
|  | (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:

|  |  |
| --- | --- |
|  | (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.

# The substation-flooding damage scheme

A substation-flooding damage model is presented and integrated in SPHERA (Amicarelli et al., 2021, [61]). 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.

* 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).

* + 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, [62]) considers a threshold water depth of *Yth*=1.2m 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, [63]) explains that the above threshold value is overestimated. In particular, the majority of the UK electrical substations has shown a threshold of *Yth*=0.30m (Crawford & Seidel, 2013, [64]). Considering this value and a maximum threshold of *Yth,max*=0.50m, Holmes (2015, [63]) 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 (*Ysub*, m):

|  |  |
| --- | --- |
|  | (10.1) |

The above relationship has been validated on other open-field data (Holmes, 2015, [63]).

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:

|  |  |
| --- | --- |
|  | (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, [65]) report the interval *trei*=0’-500’; Maliszewski & Perrings (2012, [66]) suggest an average value of *trei,e*=99’ and a maximum value of *trei,max*=330’. Both the above references consider a typical value of *trei* smaller than two hours. However, their analyses only consider blackout events under normal (non-extreme) environmental conditions. Reed (2008, [67]) proposes to use a particular realization of the “gamma” probability density function, for *trei* 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, [67]-), the present model assumes an expected restoration time of 11 hours (*trei,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:

|  |  |
| --- | --- |
|  | (10.3) |

where *∆t* 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*≥*trei,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).

* + 1. 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 (*Dsub*). This damage model is complete as it quantifies both vulnerability and damage (the latter in monetary units):

|  |  |
| --- | --- |
|  | (10.4) |

The value of the electrical substations (*Va,sub*) can be expressed in US dollars (HAZUS-MH, 2011, [62]):

|  |  |
| --- | --- |
|  | (10.5) |

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

|  |  |
| --- | --- |
|  | (10.6) |

*Vu,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 *Vu,sub*, after elaboration of the point values reported by HAZUS-MH (2011, [62]):

|  |  |
| --- | --- |
|  | (10.7) |

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

* 1. 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 *Ysub* as spatial average of the water depth values at the DEM vertices within the polygon of the electrical substation;
* update of the variable *Ysub,max*;
* assessment of the variables *POS* and *EOS*;
* update of the variables *EOT*, *Dsub* and*Vu,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, [68]). 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 “Atlarete 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).

# Developer guide

The developer guide is represented by the following sub-sections: a synthetic description of the program units (Sec.11.1); the style formatting (Sec.11.2); the modifications with respect to SPHERA v.8.0 (Sec.11.3).

* 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.

* + 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 11.1).

* + 1. 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 11.2).

* + 1. 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 11.3).

* + 1. 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, [2]; Table 11.4).

* + 1. Program units for the constitutive equation

The folder “Constitutive\_Equation” contains the program units for the constitutive equation (Table 11.5; Amicarelli et al., 2017, [1]).

|  |  |
| --- | --- |
| **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 11.1. 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 quadratic 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, [5]), also with DB-SPH boundary treatment scheme. |
| PressureSmoothing\_3D | Partial smoothing for pressure (Di Monaco et al., 2011, [5]), also with DB-SPH boundary treatment scheme. |

Table 11.2. 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, [26]) 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 11.3. Program units for the momentum equation (“BE\_momentum”; SPHERA v.9.0.0).

* + 1. 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, [3]; Table 11.6).

|  |  |
| --- | --- |
| **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, [2]) |
| 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, [2]) |
| body\_to\_smoothing\_vel | Contributions of body particles to velocity partial smoothing (Amicarelli et al., 2015, [2]) |
| Gamma\_boun | Interpolative function defined by Monaghan (2005, [26]) for boundary force particles (Amicarelli et al.,2015, [2]). |
| 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, [2]). |

Table 11.4. 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, [1]) |
| Viscapp | Constitutive equation with tuning parameters (validated in Manenti et al.,2012, [4]) |

Table 11.5. Program units for the constitutive equation (“Constitutive\_Equation”; SPHERA v.9.0.0).

* + 1. 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, [4]) and its further developments (Table 11.7).

* + 1. 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 11.8).

* + 1. Program units for the initial conditions (IC)

The folder “IC” contains the program units on the management on the initial conditions (Table 11.9).

* + 1. 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”.

* + 1. 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 11.10).

Table 11.11 lists the program units called by the program unit “time\_step\_loop”, when the time integration scheme Leapfrog is activated (SPHERA). The call order is highlighted.

* + 1. Modules

The folder “Modules” contains the Fortran modules of SPHERA v.9.0.0. (Table 11.12).

* + 1. 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 11.13).

|  |  |
| --- | --- |
| **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, [3]). |
| 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, [3]). |
| 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, [3]). |
| 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, [3]). |

Table 11.6. Program units for the boundary treatment scheme DB-SPH (“DB\_SPH”; SPHERA v.9.0.0).

* + 1. Program units for post-processing

The folder “Post\_processing” contains the program units to post-process the code results (Table 11.14). 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;
* 3D 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”).

Monitor interpolations should take into account the influence of SA-SPH frontiers in the program unit “interpolations\_for\_monitoring\_element”. However, it is cumbersome to perform a SA-SPH neighbouring search starting from a point (not a particle): it would be smarter (despite the approximation) to find the closest fluid particle, consider its SA-SPH neighbouring frontiers and interpolate (as for partial smoothing). Instead, a simpler solution is implemented:

* while assessing contributions to the interpolation for the monitoring element from fluid particles, find the closest fluid particle within the kernel support of the monitoring element;
* at the end of the program unit, if the possible closest fluid particle has neighbouring SA-SPH frontiers, then the interpolation are replaced by this particle value, else keep the interpolation. It seems smarter to refer to the closest fluid particle as the particle values are already representative of their kernel support.

SPHERA returns the time series of the maximum height (file “.plb”) and the minimum height (file “.zlft”) of the free surface. The latter represents the position of the lowest water-air interface, along a generic vertical monitoring line. This quantity seems proper to assess the water depth, obtained after difference with the height of the underlying solid surface. The algorithm searches in the positioning grid the lowest height of the free surface (from the bottom towards the top of a monitoring line) and stops at the first positioning cell which is empty and has at least one cell filled of water below (along the vertical). The filtering procedure is based on pinpointing air masses between portions of liquid along a vertical monitoring line. This procedure might be improved integrating an additional control on the possible detachment between the lowest portion of the local liquid sub-domain and the underlying solid surface.

|  |  |
| --- | --- |
| **Program unit** | **Synthetic Description** |
| compute\_k\_BetaGamma | To compute k\_BetaGamma=teta\_c/teta\_c,00. k\_BetaGamma is the ratio between Shields critical non-dimensional stress for a generic 3D slope (teta\_c) and its analogous value defined by Shields diagram (teta\_c,00) on flat bed. |
| drafts | Mohr-Coulomb 2D erosion criterion (Manenti et al., 2012, [4]). Shields erosion criterion works better (Manenti et al., 2012, [4]). |
| fixed\_bed\_slope\_limited | Forced deposition (or no erosion) for particles at least 2h below the fixed bed (as it is defined in the associated column) during the same time step: i.e. the maximum slope of the fixed bed is 2h/2h. This avoids eventual too fast propagation of erosion along the vertical (erosion is an interface phenomenon). |
| Shields | 3D erosion criterion based on the formulation of both Shields-van Rijn 2D criterion and Seminara et al. (2002, [58]) 3D criterion. 2D Shields erosion criterion based on pure fluid - fixed bed interactions (Manenti et al., 2012, [4]). 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, [4]; Amicarelli et al., CAF, submitted). Shields threshold for low Re\* (Amicarelli et al., CAF, submitted). |

Table 11.7. 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 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. |
| point\_inout\_quadrilateral | 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 11.8. Program units on Geometry (“Geometry”; SPHERA v.9.0.0).

|  |  |
| --- | --- |
| **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 11.9. 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 11.10. Program units for the main algorithms (“Main\_algorithm”; SPHERA v.9.0.0).

* + 1. Program units for pre-processing

The folder “Pre\_processing” cntains the program units (Table 11.15) to pro-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** | **Program unit**  **(direct calls)** | **Program unit**  **(1st order indirect calls)** | **Program unit**  **(2nd order ind. calls)** |
| time\_step\_loop |  |  |  |
|  | time\_step\_duration |  |  |
|  | fluid\_particle\_imposed\_kinematics |  |  |
|  | RHS\_momentum\_equation |  |  |
|  |  | inter\_EqMoto |  |
|  |  |  | viscomon |
|  |  |  | viscomorris |
|  |  | AddBoundaryContributions\_to\_ME3D |  |
|  |  |  | wall\_function\_for\_SASPH |
|  |  | AddElasticBoundaryReaction\_3D |  |
|  | RHS\_body\_dynamics |  |  |
|  |  | body\_pressure\_mirror |  |
|  | Leapfrog\_momentum |  |  |
|  | time\_integration\_body\_dynamics |  |  |
|  | velocity\_smoothing |  |  |
|  |  | body\_to\_smoothing\_vel |  |
|  |  | velocity\_smoothing\_SA\_SPH\_3D |  |
|  |  |  | wall\_function\_for\_SASPH |
|  | velocity\_smoothing\_2 |  |  |
|  | Leapfrog\_trajectories |  |  |
|  | CancelOutgoneParticles\_3D |  |  |
|  | GenerateSourceParticles |  |  |
|  | BC\_zmax\_anyt |  |  |
|  |  | z\_FS\_max\_9p\_stencil |  |
|  | OrdGrid1 |  |  |
|  | NormFix |  |  |
|  | CalcVarLength |  |  |
|  | liquid\_particle\_ID\_array |  |  |
|  | velocity\_smoothing |  |  |
|  | velocity\_smoothing\_2 |  |  |
|  | ComputeBoundaryDataTab\_3D |  |  |
|  |  | FindCloseBoundaryFaces3D |  |
|  |  | ComputeBoundaryVolumeIntegrals\_P0 |  |
|  | KTGF\_update |  |  |
|  |  | initialization\_fixed\_granular\_particle |  |
|  |  | Shields |  |
|  |  |  | fixed\_bed\_slope\_limited |
|  |  |  | compute\_k\_BetaGamma |
|  | liquid\_particle\_ID\_array |  |  |
|  | Continuity\_Equation |  |  |
|  |  | body\_particles\_to\_continuity |  |
|  | SASPH\_continuity |  |  |
|  |  | AddBoundaryContribution\_to\_CE3D |  |
|  | Leapfrog\_continuity |  |  |
|  | CalcPre |  |  |
|  | body\_pressure\_mirror |  |  |
|  | PressureSmoothing\_3D |  |  |
|  |  | body\_to\_smoothing\_pres |  |
|  | body\_pressure\_postpro |  |  |
|  | mixture\_viscosity |  |  |
|  | time\_step\_post\_processing |  |  |

Table 11.11. Program units called by the program unit “time\_step\_loop”, when the time integration scheme Leapfrog is activated (SPHERA).

|  |  |
| --- | --- |
| **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, [5]; Manenti et al., 2012, [4]; Amicarelli et al., 2013, [3]; Amicarelli et al., 2015, [2]). |
| 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, [5]). |
| Static\_allocation\_module | Module to define global (and statically allocated) variable. |
| Time\_module | Module for time recording. |

Table 11.12. 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 11.13. 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 11.14. 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 trasnport in fluid flows (Amicarelli et al., 2015, [2]). |
| ReadCheck | Minor program unit |
| ReadDBSPH | Reading input data for the DB-SPH boundary treatment scheme (Amicarelli et al., 2013, [3]). |
| ReadInput | Reading input data. |
| ReadInputBoundaries | Reading input data for the boundary treatment scheme SA-SPH (semi-analytic approach; Di Monaco et al., 2011, [5]). |
| 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 11.15. 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 npi. It performs implicit computation of gradPsuro. (Di Monaco et al., 2011, [5]). |
| AddBoundaryContribution\_to\_CE3D | To compute boundary terms for the 3D continuity equation (rodivV). Equation refers to particle npi. It performs implicit computation of gradPsuro. (Di Monaco et al., 2011, [5]). |
| AddBoundaryContributions\_to\_ME2D | To compute boundary terms for the 2D momentum equation (gradPsuro, ViscoF). Equations refer to particle npi. (Di Monaco et al., 2011, [5]). |
| AddBoundaryContributions\_to\_ME3D | To compute boundary terms for 3D momentum equation (gradPsuro, ViscoF). Equations refer to particle npi. It performs implicit computation of gradPsuro. (Di Monaco et al., 2011, [5]). |
| AddElasticBoundaryReaction\_2D | To add supplementariìy normal boundary reaction to support eventual insufficient pressure gradient boundary term. In case of few neighbouring particles and presence of normal component of mass force (gravity). The normal reaction is computed with the formula R=(c0^2/d) ln(zi/d) [for zi<d], stemming from the compressible reaction of the fluid, where:  c0^2 = E/ro0 is the square of the sound speed within the fluid;  zi is the distance of the particle Pi from the boundary face;  d is a reference distance from which the reaction is added.  Check that the elastic boundary reaction never works.  To compute the boundary integral IntWdS  (Di Monaco et al., 2011, [5]). |
| AddElasticBoundaryReaction\_3D | To add supplementary normal boundary reaction to support eventual insufficient pressure gradient boundary term. In case of few neighbouring particles and presence of normal component of mass force (gravity). The normal reaction is computed with the formula R=(c0^2/d) ln(zi/d) [for zi<d], stemming from the compressible reaction of the fluid, where:  c0^2 = E/ro0 is the square of the sound speed within the fluid;  zi is the distance of the particle Pi from the boundary face;  d is a reference distance from which the reaction is added.  Check that the elastic boundary reaction never works. (Di Monaco et al., 2011, [5]). |
| BoundaryMassForceMatrix2D | Generation of the generalised boundary mass force matrix RN, on the base of the cosine matrix T and the parameter Fi. (Di Monaco et al., 2011, [5]) |
| BoundaryMassForceMatrix3D | Generation of the generalised boundary mass force matrix RN, on the base of the cosine matrix T and the parameter Fi. (Di Monaco et al., 2011, [5]) |
| BoundaryPressureGradientMatrix3D | To generate the pressure gradient matrix RRP, based on the cosine matrix T and the parameter vector Psi. (Di Monaco et al., 2011, [5]) |
| BoundaryReflectionMatrix2D | Generation of the generalised reflection matrix R, based on the cosine matrix T and the parameters PsiS and PsiN. (Di Monaco et al., 2011, [5]) |
| BoundaryVolumeIntegrals2D | To compute the boundary volume integrals IntWdV. (Di Monaco et al., 2011, [5]) |
| CompleteBoundaries3D | (Di Monaco et al., 2011, [5]) |
| ComputeBoundaryDataTab | To calculate the array to store close boundaries and integrals. (Di Monaco et al., 2011, [5]) |
| ComputeBoundaryIntegralTab | To compute local coordinates (x,y,z) of a grid of points, regularly distributed on the semisphere z<0 (radius = 2h), whose centre is the origin O of local axis. The semisphere will be superposed to the influence sphere (kernel support) of a generic particle near a plane boundary face, and oriented in such a way that the axis (x,y,z) coincide with the face local axes (r,s,n). In the first three columns of the array BoundaryIntegralTab() the coordinates (x,y,z) of each point are stored; in the forth column the relative d\_alpha (portion of solid angle relative to the point, necessary for integrations) is stored. BITcols = 4. (Di Monaco et al., 2011, [5]) |
| ComputeBoundaryVolumeIntegrals\_P0 | (Di Monaco et al., 2011, [5]) |
| ComputeKernelTable | To pre-compute and store in kerneltab(0:ktrows,0:ktcols) the following values:  kerneltab(0:ktrows, 0) = rob = rb/h  kerneltab(0:ktrows, 1) = Int W\* ro2 dro (from rob to 2)  kerneltab(0:ktrows, 2) = Int dW\*/dro ro dro (from rob to 2)  kerneltab(0:ktrows, 3) = Int dW\*/dro ro^2 dro (from rob to 2)  kerneltab(0:ktrows, 4) = Int dW\*/dro ro^3 dro (from rob to 2)  (Di Monaco et al., 2011, [5]) |
| ComputeSurfaceIntegral\_WdS2D | Computing the surface integral of kernel W along the segments intercepted by the kernel support (radius=2h) of the particle i, whose local coordinates are xpi=LocXY(1,icbs) and ypi=LocXY(2,icbs), on the adjacent boundary side icbs. (Di Monaco et al., 2011, [5]) |
| ComputeVolumeIntegral\_WdV2D | Computing the integral of WdV extented to the volume delimited by the kernel support (radius=2h) of the particle i, whose local coordinates are xpi=LocXY(1,icbs) and ypi=LocXY(2,icbs), and the adjacent boundary side icbs. (Di Monaco et al., 2011, [5]) |
| DefineBoundaryFaceGeometry3D | To define boundary faces from 3D geometry. (Di Monaco et al., 2011, [5]) |
| DefineBoundarySideGeometry2D | Definition of the boundary sides. (Di Monaco et al., 2011, [5]) |
| DefineBoundarySideRelativeAngles2D | Detection of the previous adjacent side and associated relative angle (for each boundary side). (Di Monaco et al., 2011, [5]) |
| DefineLocalSystemVersors | To define the directional cosines of the local reference system. (Di Monaco et al., 2011, [5]) |
| EvaluateBER\_TimeStep | (Di Monaco et al., 2011, [5]) |
| 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, [5]) |
| FindBoundaryIntersection2D | To find the intersection segment between the kernel support of particle i, whose local coordinates are xpi=LocXY(1,icbs) and ypi=LocXY(2,icbs), and the straight boundary side iside=Cloboside(icbs), which lies on the local x-axis and extends from x=0 to bsidelen = BoundarySide(iside)%Length. It returns:  xpmin: minimum abscissa of intersected segment  xpmax: maximum abscissa of intersected segment  interlen: length of the intersected segment  (Di Monaco et al., 2011, [5]) |
| FindCloseBoundaryFaces3D | To finds the "close" boundary faces, i.e. those faces located at a distance from the particle npi smaller than or equal to 2h. It returns:  Ncbf: number of close boundary faces  Clobface(1 to Ncbf): list of close boundary faces  LocX(1:SPACEDIM,Ncbf): local coordinates of particle npi with respect each boundary side  The algorithm looks for the boundary faces intersected by the cell boxes of the reference frame located all around particle npi, and cancels the repeated ones. (Di Monaco et al., 2011, [5]) |
| FindCloseBoundarySides2D | To finds the "close" boundary sides, i.e. those sited at a distance from particle npi<=2h. It returns:  Ncbs: number of close boundary sides (= 0, 1, 2)  Cloboside(1:Ncbs): list of close boundary sides  LocXY(1:PLANEDIM,1:Ncbs): local coordinates of particle npi with respect each boundary side (vertex V1)  (Di Monaco et al., 2011, [5]) |
| 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, [5] |
| 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, [5]) |
| 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, [5]) |
| IWro2dro | Computes a SA-SPH definite integral (Di Monaco et al., 2011, [5]) |
| J2Wro2 | Computes a SA-SPH definite integral (Di Monaco et al., 2011, [5]) |
| JdWsRn | Computes a SA-SPH definite integral (Di Monaco et al., 2011, [5]) |
| SelectCloseBoundarySides2D | Selecting among the close boundary sides, those that really give contribution to the equations of particle 'npi'. It returns:  IntNcbs: number of close boundary sides, which give contribution (= 0, 1, 2)  Intboside(1:IntNcbs): list of close boundary sides, which give contribution  IntLocXY(1:PLANEDIM,1:Ncbs): local coordinates of particle np with respect each boundary side, which gives contribution  (Di Monaco et al., 2011, [5]) |
| WIntegr | Computing a SA-SPH definite integral (Di Monaco et al., 2011, [5]) |

Table 11.16. 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 11.17. Program units for time integration (“Time\_integration”; SPHERA v.9.0.0).

* + 1. 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, [5]; Table 11.16).

* + 1. Program units for managing Fortran character variables

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

* + 1. Program units for time integration

The folder “Time\_integration” contains those program units, which concern RK1 and RK2 time integration schemes (Table 11.17).

* 1. 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.

* 1. Modifications with respect to SPHERA v.8.0

The list of modifications to SPHERA are reported in the github code repository as a git log (SPHERA, 2021, [6]).

# User guide

SPHERA installation is straightforward (Sec.12.1), 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.12.2). This template defines and comments all the input parameters. Finally, SPHERA v.9.0.0 tutorials are discussed in Sec.12.3.

* 1. Installation

SPHERA (2021, [6]) source and executable files are distributed on a dedicated Git repository on github.com. 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).

|  |  |  |  |
| --- | --- | --- | --- |
| **Makefile variable** | **Suggested value** | **Executable**  **name:**  **SPHERA\_j\_abcdefghi** | **Description** |
| VERSION | v9\_UserInitials\_CommitData | j=VERSION | Master subversion compiled by the user |
| COMPILER | ifort | a=1 | Compiler |
|  | gfortran | a=2 |  |
| EXECUTION | optimized | b=1 | Execution mode |
|  | debug | b=2 |  |
| PARALLELIZATION | OMP | c=1 | parallel simulations |
|  | NO | c=2 | sequential simulations |
| PD\_SPACE | -DSPACE\_3D | d=3 | 3D (macro) |
|  | -DSPACE\_2D | d=2 | 2D (macro) |
| PD\_VERBOSITY | (blank space) | e=0 | Synthetic output |
|  | -DVERBOSITY | e=1 | Verbose output (macro, inactive) |
| PD\_KTGF | -DKTGF\_FULL | f=1 | KTGF scheme (dense granular flows) with possible 2-interface 3D erosion criterion (macro, inactive) |
|  | -DKTGF\_NO | f=0 | no KTGF scheme (macro, inactive) |
|  | -DKTGF\_EC2D | f=2 | KTGF scheme with possible 1-interface 2D erosion criterion (macro, inactive) |
| PD\_SOLID\_BODIES | -DSOLID\_BODIES | g=1 | Body transport (macro) |
|  | (blank space) | g=0 | No body transport |
| PD\_BTM | -DBTM\_SASPH | h=1 | BOUNDARY TREATMENT METHOD: SASPH (macro, inactive) |
|  | -DBTM\_DBSPH | h=2 | BOUNDARY TREATMENT METHOD: DBSPH (macro, inactive) |
| PD\_TIS | -DTIS\_LEAPFROG | i=3 | TIME\_INTEGRATION SCHEME: Leapfrog (macro, inactive) |
|  | -DTIS\_EULER | i=1 | TIME\_INTEGRATION SCHEME: Euler (macro, inactive) |
|  | -DTIS\_HEUN | i=2 | TIME\_INTEGRATION SCHEME Heun (macro, inactive) |

Table 12.1. Makefile variables; macros for the preprocessor directives; executable names (Makefile of SPHERA, 2021, [6]). The following constraints/incompatibilies are automatically corrected by the Makefile: “KTGF” needs “SASPH“; “SOLID\_BODIES” needs “SASPH” and “LEAPFROG”; DBSPH” needs “EULER”.

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”).

The Makefile (under the folder “src”) allows compiling SPHERA under different configurations, as explained in Table 12.1. SPHERA is optimized by means of the following features: preprocessor directives (three macros are active); compilation options “-O2” (for both the compilers “gfortran” and “ifort”) and “-ipo” (for the compiler “ifort”).

* 1. Commented template of the main input file of SPHERA v.9.0.0

The commented template of the main input file of SPHERA is reported in the code repository (SPHERA, 2021, [6]) and is synthesized in Table 12.2. The comments define all the input parameters and describe the meaning of their possible values. Further, suggested or default values are reported.

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

Table 12.2. Sections and relevant quantities of the main input file of SPHERA v.9.0.0 (2018, [6]).

* 1. 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.

* + 1. “body\_body\_impacts”

This tutorial is completely described in Amicarelli et al. (2015, [2]). The paper version available on ResearchGate might help in case the published version is unavailable.

* + 1. “body\_boundary\_impacts”

This tutorial is completely described in Amicarelli et al. (2015, [2]). The paper version available on ResearchGate might help in case the published version is unavailable.

* + 1. “dam\_breach\_ICOLD\_trunks”

This tutorial is described in Amicarelli & Agate (2014, [50]) and Amicarelli & Agate (2015, [69]). Here, the tutorial is represented by means of the mixture model for dense granular flows of Amicarelli et al. (2017, [1]).

* + 1. “db\_2bodies”

This tutorial is completely described in Amicarelli et al. (2015, [2]). The paper version available on ResearchGate might help in case the published version is unavailable.

* + 1. “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. (2020, [47]) completely describes this test case. This project report is Open-Access and also includes a synthetic English version.

* + 1. “db\_Alpe\_Gera”

This tutorial is completely described in Amicarelli & Paggi (2019, [70]).

* + 1. “db\_Alpe\_Gera\_Lanzada\_substations”

This tutorial is completely described in Amicarelli (2021, [61]). This project report is Open-Access and also includes a synthetic English version.

* + 1. “db\_body\_exp\_UniBas”

This tutorial is completely described in Amicarelli et al. (2015, [2]). The paper version available on ResearchGate might help in case the published version is unavailable.

* + 1. “db\_ICOLD”

This tutorial is described in Amicarelli & Agate (2014, [50]).

* + 1. “db\_multi\_body”

This tutorial is completely described in Amicarelli et al. (2015, [2]). The paper version available on ResearchGate might help in case the published version is unavailable.

* + 1. “db\_squat\_obstacle”

This tutorial is described in Amicarelli et al. (2013, [3]). The paper version available on ResearchGate might help in case the published version is unavailable. Here, the surface mesh is generated by SnappyHexMesh (OpenCFD Ltd, 2018, [71]) and SPHERA source code improvements for the DB-SPH treatment (Amicarelli & Agate, 2015, [69]) apply.

* + 1. “db\_tall\_obstacle”

This tutorial is described in Amicarelli et al. (2013, [3]). The paper version available on ResearchGate might help in case the published version is unavailable. Here, the surface mesh is generated by SnappyHexMesh (OpenCFD Ltd, 2018, [71]) and SPHERA source code improvements for the DB-SPH treatment (Amicarelli & Agate, 2015, [69]) apply.

* + 1. “dike\_breach\_2D\_expSchHag12JHR\_ID22”

This tutorial is completely described in Amicarelli et al. (2016, [60]).

* + 1. “edb\_2D\_demo”

This tutorial of SPHERA v.9.0.0 (2021, [6]) was used as a reference test case to carry out a sensitivity analysis in Manenti et al. (2018, [72]). This paper is Open-Access and also describes in detail the present test case.

* + 1. “edb\_2D\_FraCap02”

This tutorial is described in Amicarelli et al. (2017, [1]). The paper version available on ResearchGate might help in case the published version is unavailable.

* + 1. “edb\_2D\_FraCap02\_Taipei”

This tutorial is described in Amicarelli et al. (2017, [1]). The paper version available on ResearchGate might help in case the published version is unavailable.

* + 1. “edb\_2D\_Spi05”

This tutorial is described in Amicarelli et al. (2017, [1]). The paper version available on ResearchGate might help in case the published version is unavailable.

* + 1. “edb\_ICOLD”

This tutorial is described in Amicarelli et al. (2017, [1]). The paper version available on ResearchGate might help in case the published version is unavailable.

* + 1. “edb\_KarlSand”

This tutorial is described in Amicarelli et al. (2017, [1]). The paper version available on ResearchGate might help in case the published version is unavailable.

* + 1. “edb\_Pon10”

This tutorial is described in Amicarelli et al. (2017, [1]). The paper version available on ResearchGate might help in case the published version is unavailable.

* + 1. “floating\_cube\_stability”

This is a very simple and demonstrative tutorial: a solid cube is leaned on still water (rough resolution). Amicarelli & Agate (2014, [50]) completely describes this test case.

* + 1. “flushing\_2D”

This tutorial is described in Amicarelli & Agate (2014, [50]). Here, the tutorial is represented by means of the mixture model for dense granular flows of Amicarelli et al. (2017, [1]).

* + 1. “flushing\_3D”

As a 3D version of the previous tutorial, this test case is available on Amicarelli et al. (2014, [73]). This project report is Open-Access and also includes a synthetic English version.

* + 1. “jet\_plate”

The 7 variants of this tutorial are described in Amicarelli et al. (2013, [3]) and Amicarelli et al. (2015, [2]). Here the DBSPH boundary scheme is treated with automated procedures (Amicarelli & Agate, 2015, [69]) and the scheme for body transport is corrected in terms of no-slip conditions for fluid-body interactions (RSE developments reported in Paggi et al., 2021, [48]).

* + 1. “rectangular\_side\_weir\_Fr\_0\_491”

This tutorial is described in Amicarelli (2018, [74]). This project report is Open-Access and also includes a synthetic English version.

* + 1. “San\_Fernando\_Lower\_van\_Norman\_dam\_liquefaction”

This tutorial is described in Amicarelli (2016, [60]). Here, the tutorial is represented by means of the mixture model for dense granular flows of Amicarelli et al. (2017, [1]).

* + 1. “sloshing\_tank\_TbyTn\_0\_78”

This tutorial is described in Amicarelli et al. (2013, [3]). The paper version available on ResearchGate might help in case the published version is unavailable. Here, the surface mesh is generated by SnappyHexMesh (OpenCFD Ltd, 2018, [71]) and SPHERA source code improvements for the DB-SPH treatment (Amicarelli & Agate, 2015, [69]) apply.

* + 1. “sloshing\_tank\_TbyTn\_1\_07”

This tutorial is described in Amicarelli et al. (2013, [3]). The paper version available on ResearchGate might help in case the published version is unavailable. Here, the surface mesh is generated by SnappyHexMesh (OpenCFD Ltd, 2018, [71]) and SPHERA source code improvements for the DB-SPH treatment (Amicarelli & Agate, 2015, [69]) apply.

* + 1. “spherical\_Couette\_flows”

This tutorial is described in Amicarelli et al. (2021, [49]).

* + 1. “SPH\_udb\_exp\_Kim2015HYDROL”

This tutorial is described in Amicarelli (2021, [61]).

* + 1. “submerged\_landslide”

This tutorial is described in Amicarelli & Agate (2014, [50]). Here, the tutorial is represented by means of the mixture model for dense granular flows of Amicarelli et al. (2017, [1]).

* + 1. “still\_water\_tank”

This tutorial is a very simple and rough-resolution 2D test case on hydrostatic conditions.

* + 1. “wave\_motion\_for\_WaveSAX”

This tutorial is described in Amicarelli et al. (2020, [47]).

* + 1. “wedges\_falls\_on\_still\_water”

This tutorial is completely described in Amicarelli et al. (2015, [2]). The paper version available on ResearchGate might help in case the published version is unavailable.

# Use procedure of the modelling chain of SPHERA

An overview of the numerical modelling chain of SPHERA is reported in Figure 13.1.

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, 2021, [75], text editor), gfortran (GNU, Free Software Foundation, 2021, [76]; Fortran compiler), gmake (GNU, Free Software Foundation, 2021, [77]; for the Makefile execution), gdb (GNU, Free Software Foundation, 2021, [78]; debugger), gprof (GNU, Free Software Foundation, 2021, [79]; profiler for scalar executions), Valgrind (Valgrind Developers, 2021, [80]; memory management tool), Scalasca (Forschungszentrum Jülich & TU Darmstadt, 2021, [81]; code profiler for OMP executions). The RSE tools are developed and distributed with no charge by means of Git (Torvalds et al., 2021, [82]; main “Distributed Version Control System” -DVCS-) and GitHub (GitHub Inc., 2021, [83]; 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 (2021, [84]). 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, Engauge 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, 2021, [85]; Fortran compiler); idb (Intel Corporation, 2021, [86]; Fortran debugger); cpuinfo (Intel Corporation, 2021, [87]; code profiler for OMP executions); ITAC, Trace Analyzer and VTUNE (Intel Corporation, 2021, [88]; code profiler for OMP/MPI executions); Advisor (Intel Corporation, 2021, [89]; profiler for executions with code vectorization); TotalView (Rogue Wave Software, 2018, [90]; 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.

* 1. Engauge Digitizer

Some input data need a digitalization procedure. Further, the profiles which will be simulated will need to be compared with the analogous experimental (or numerical) profiles available from experimental images or the scientific literature (indexed journals; Open-Data archives).

In all the above cases, it is normally admitted the digitization of experimental and numerical profiles from published sources by means of Engauge Digitizer (Mitchell et al., 2021, [91]), with proper citation of the source, in order to validate the code.

* 1. Site overview

1. Define the site boundaries.
2. Provide the reference geographic and cartographic coordinates of a representative point within the domain.
3. Preliminary assessment of the elapsed time *te* (s) as function of the fluid and topography spatial resolution (*dx* and *dxDEM,DTM*). The elapsed time roughly depends on the average number of SPH particles *Navg*, the number of boundary elements *NBC,elements* and the number of time steps *Nstep* (*te**Navg* ∙ *NBC,elements* ∙ *Nstep*). Define the main variables influencing *Navg*, *NBC,elements* and *Nstep*. They are test-case dependent. Compare them to an analogous test case with known *te*.
   1. The SRTM3 dataset and further DEM-DTM datasets

Published at the end of 2014, the dataset SRTM3 (USGS, 2014, [92]) 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 (Rexer & Hirt, 2014, [93]). However, these estimations refer to almost the whole terrestrial surface, included the high-latitude regions where errors are definitely higher. Any additional DEM and DTM files might replace/integrate the SRTM3 DEM files for the current test case. For Italian sites, the web sites of the regions and the Ministry MiTE might represent the main references.

The effective spatial resolution of the DEM/DTM is  times its formal spatial resolution as Delaunay’s triangulation applies each rectangular boundary element is halved.

* 1. GDAL

The software tool GDAL (OSGEO, 2021, [43]), 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 “.asc”.

|  |
| --- |
|  |

Figure 13.1. FOSS / Freeware / Open-Data tools of the modelling chain for SPHERA.

* 1. DEM2xyz: preliminary elaboration of the raw DEM and DTM files

“DEM2xyz v.2.0” (RSE SpA) reads a “DEM” file and writes the associated DEM in a corresponding “xyz” file, possibly changing the spatial resolution (as requested by the user). In case the absolute value of the mean latitude is provided with a non-negative value, the following conversion takes place "(*lon*, *lat*) in (°) to (*X*, *Y*) in (m)". In this case, an interpolation (weighted on the inverse of the distance square) is carried out to provide a regular cartographic output grid in (*X*, *Y*). The height of the DEM points which belong to the digging/filling regions (provided in input) is modified. After this treatment, each digging/filling region has null slope. Bathymetry is possibly extruded from the heights of the most upstream and downstream coastline points. The bathymetry/reservoir extrusion is corrected in case the volume reservoir is provided as an input parameter. Multiple reservoirs are admitted. Digging regions cannot overlap each other unless they are set to the same height. Reservoir/bathymetry regions cannot overlap each other. In case a digging region overlaps a reservoir region, the latter holds the priority. In the presence of a volume correction, two reference shapes are available: "reservoir" and "volcanic lake". DEM2xyz v.2.0 is compatible with SPHERA v.9.0.0 (RSE SpA).

The current procedure has to be repeated for each dataset (multiple datasets).

The origin of the reference system of the DEM2xyz output is the lower-left vertex of the DEM2xyz input (relative reference system).

Use the same relative reference system for all the DTMs: at this stage the origin of the relative reference system is the lower-left vertex of the lower-left DEM-DTM of the current dataset.

* 1. Paraview: post-processing of the preliminary DEM and DTM files

The output files of DEM2xyz (“.txt” or “.xyz”) can be post-processed by means of Paraview (Kitware, 2021, [94]), by executing the following procedure (to be repeated for each DTM and DEM file):

1. open the .txt/.csv file; change the "Field Delimiter Character" to a blank space by typing one blank space; choose the option “Merge consecutive limiters”;
2. apply the filter “TabletoPoints”;
3. eventually cut the proper domain especially if the virtual memory is insufficient: halve the DTM step by step by halving the maximum among 0.5 ∙ *ncolumns* (*x* values; “0.5” is due to the threshold anisotropy in Paraview when building “Delaunay2D” tessellations) and *nrows* (*y* values). This procedure will efficiently respect both the maximum number of elements, columns and rows for the filter “Delaunay2D” of Paraview.
4. apply the filter “Delaunay2D”; build a Delaunay tessellation at this stage that the DTMs are not merged to avoid crashes in case of big DEM-DTM files;
5. remove the spurious parts of the output of Delaunay2D;
6. "Save Data>.ply".
7. Save data as ASCII .ply and binary .ply; check both on PV.
8. Save data as .csv:
   1. use scientific notation; select more than 5 significant decimal digits only if necessary;
   2. save;
   3. check the .csv with a text editor and compare with the precisions to be defined;
   4. reorder the columns by means of Linux OS “dos2unix”, “yes”, “cut” and “paste” commands (their combination and use is test-case dependant): first column (*x*-values), second column (*y*-values), third column (null values), fourth column (*z*-values);
   5. check on Paraview.
   6. Elaboration of a unique DEM-DTM integrated surface

Hereafter is reported a use procedure for integrating several DEM (Digital Elevation Model) files with the associated DTM (Digital Terrain Model) files in a unique wall boundary for SPH simulations, by means of the following software tools: Paraview (Kitware, 2020, [94]); DEM2xyz (2020, [95]); ply2SPHERA\_perimeter (2020, [96]); Grid Interpolator (2020, [97]). Specific treatments concern the following items: power lines and electricity pylons, trees, sub-grid roughness, bridges, DTM and DEM intrinsic bugs.

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 filling/digging regions, detects the dam toe and the most upstream point over the coastline of the water bodies.

The current procedure is composed by the following steps and has to be repeated for each dataset (multiple datasets):

1. integrate the elaborated DTM and DEM 3D surfaces:
   1. assess the 2D field of the height difference (*zdiff* -m-) between the DEM and the DTM and save it with the “.csv” format (“Save Data”; in order to optimize the Virtual Memory in the following steps, select more than 5 significant decimal digits only if necessary);
   2. ad-hoc treatment of the intrinsic errors due to the definition of the DEM (i.e., a surface described by a 3D function, a single quote being associated with any couple of horizontal coordinates) by means of Paraview (local treatment of the field of zdiff, obstacle by obstacle, where necessary): select the horizontal coordinates of the “digging regions” to feed DEM2xyz (one can use the “Source Splines” for design removing the exceeding decimal digits, provided the precision requested by DEM2xyz):
      1. fix the elevated permeable obstacles:
         1. remove the power lines, the electricity pylons and the other truss structures as they are extruded to the ground as impermeable obstacles;
         2. the tree foliage can be left, except for those trees too close each other whose extrusion to the ground locally form a fictitious “tree impermeable wall”;
         3. removal of bridges as they are extruded to the ground as impermeable obstacles, whereas their depth is limited;
      2. remove the clearly fictitious obstacles (i.e., the DEM bugs) from the riverbeds;
         1. convert the *zdiff* file from the “.csv” to the “.xyz” format for Grid Interpolator (one can use the commands “cut” and “paste” from Linux terminal);
         2. convert the *zdiff* file from the “.xyz” to the “.asc” format by means of Grid Interpolator;
         3. apply the “digging regions” by means of DEM2xyz to zero the DEM/DTM differences at those nodes where the DEM bugs are selected, and convert the resulting file (variable *zdiff,digs*) from the “.asc” to the “.xyz” format;
         4. upload in Paraview the field of *zdiff,digs*, apply the filter *hf* (m) and add the DTM heights to obtain the DEM field of *zDEM,digs,filter*, filtered with respect to the sub-grid roughness and DEM bugs;

The Paraview use procedure for the DEM/DTM elaboration needs a filter for the local height difference between the DEM and the DTM. Hereafter the formulation of a generic filter in case one only uses the integer rounding function (“*rint*”) as discontinuous function. This is the case when using Paraview without Python scripts (Paraview filter “Python Caculator”; no “*if*” construct is available, no other discontinuous function is avilable such as the Heaviside step).

The definition of the DEM height filtered (*zDEM,filt*, -m-) depends on the DEM height (*zDEM*, m), the DTM height (*zDTM*, m), the filter threshold on the height difference (*zdiff,thr* m) and the height precision *zpr* (m):

|  |  |
| --- | --- |
|  | (13.1) |

The filter *hf* for the DEM/DTM roughness is applied for the following reasons:

1. not to explicitly simulate possible impermeable obstacles on the riverbed reported on the DEM, but not on the DTM (within the waterbody, water flows on the DTM not on the DEM);
2. not to explicitly simulate the roughness elements smaller than the sub-grid scale, already considered by the roughness length *z0* (m) in the wall function for the slip coefficient (Sec.**Errore. L'origine riferimento non è stata trovata.**);
3. to keep *hf* compatible with the chosen value of *z0* (obtained for example from Davenport’s scale; Plate, 1995, [98]), considering that *d50*=10*z0* (NSBL under rough-wall regime) and that the equivalent diameter of the sub-grid roughness elements is *d50*=2*H*, where *H* (m) is the average height of the same elements;
4. to reduce of the ad-hoc treatments associated with the DEM bugs;
5. to ease the imposition of a homogeneous field of *z0* not to distinguish the riverbed from the floodplain (in terms of sub-rid roughness); the last choice being feasible provided that the DEM-DTM surface is partitioned accordingly;
6. to keep the height precision compatible with the minimum scale of the DEM-DTM explicit roughness (they can be obtained by approximated assessments in Paraview, after comparing the DEM and the DTM).
   1. analogously repeat the previous steps to fix the intrinsic DTM errors (instead of the DEM errors; consider *zDEM,digs,filter* instead of *zdiff* and do not apply any filter);
   2. visualize the fixed obstacles removed during the previous steps by means of Paraview as passive targets which do not affect the SPH simulations;
   3. if necessary, merge two (or more) DEM/DTM adjacent files by means of Paraview: execute the following procedure only if it can be easily carried out (e.g., 2 files with coincident overlapping nodes), otherwise skip it:
      1. remove the overlapping region in one of the two files;
      2. remove from one file the redundant nodes;
      3. locally seam the boundary region between the two files (filter “Delaunay 2D”);
      4. merge the two resulting files and the boundary region (filter “Append Datasets”).
   4. Alternative merging procedure via Grid Interpolator: merging as removing overlapping parts with shared boundaries; the aim is a unique DEM-DTM surface well connected, not a unique DEM-DTM file (nx,max=1’000 and ny,max=1’000 each output file not to have issues with Paraview “Delaunay2D” filter; check identical shared boundaries and coherence with raw DEM-DTM files). The advantages of this merging are expressed as follows: no overlapping regions; clean merge (shared boundary points have the same height); no ad-hoc seals between Paraview files; it is not necessary to merge all the DTM files on Paraview (issues avoided with huge DTMs); the unique regular Cartesian grid (far from boundaries) avoids issues on SPHERA extrusions (Dirichlet’s boundary conditions for the water depth; initial conditions for water bodies extruded from DEM-DTM). Interpolation introduces some errors, thus use the minimum influence radius.
   5. Repeat the Paraview procedure 13.6.
   6. DTM/DEM dataset merging
7. Merge all the datasets (each featured by *dxDTM,DEM*) with possible roto-translation (in case of different cartographic systems). This is alternative to work since the beginning with the same cartographic system (to avoid possible complicated procedures with GDAL). The origin of the relative reference system at this final stage is the lower-left vertex of the lower-left DEM-DTM of the lower-left dataset;
8. Digging zones in DEM2xyz (for negative heights it gives NaN, e.g. *z*=-999.m) to avoid dataset overlapping (start from the finest);
9. Dataset sealing: extract the edge points of the finest dataset on Paraview and apply “Delaunay2D” with the other dataset; unique regular Cartesian grid (except for boundaries) with possible local refinement;
10. Possible interpolation to obtain a regular Cartesian grid, if requested (e.g., fluid extrusions, “BCzmax” zones).
    1. DEM2xyz: further modifications to the DEM-DTM improvements

DEM2xyz is executed again on each DEM-DTM file to:

1. possibly reconstruct the bathymetry below the water bodies;
2. possibly assign other digging/filling regions (also to fix errors);
3. cut the DEM-DTM files to reduce the computational time;
4. check the normal vectors to the surface elements of the DEM-DTM;
5. save the Paraview state;
6. generate the “.ply” files (both binary and ASCII) for ply2SPHERA\_perimeter.
7. In case of multi-spatial-resolution DTM, if possible, try to build a unique “.ply” file from the DTM points elaborated so far to avoid minor discrepancies at those edges where *dxDTM* changes.
   1. Paraview: post-processing of the final output of DEM2xyz

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.

* + 1. Simple use procedure to design a weir

Hereafter a simple and approximated use procedure to elaborate the DEM/DTM surface with Paraview in order to design a weir:

1. select a DEM-DTM zone extending 2*h* downstream the section considered and whose width is just sufficient to contain the riverbed (feature “select elements with polygone”);
2. vertically extrude the selection above up to the domain maximum height (filter “linear Extrusion”);
3. execute “Delaunay 3D” triangulation to obtain a tetrahedric grid within the extrusion;
4. extract from the extrusion volume the weir front wall (multiple use of the feature “clip”, even to remove the volume exceeding the free-surface height to be imposed, and the filters “Extract Surface” and “Delaunay 3D”) to obtain walls at least 2*h* thick;
5. analogously to the previous step, extract the weir lateral walls and remove the volume exceeding a proper height (larger than the free-surface height to be imposed and sufficient to avoid any lateral overtopping);
6. merge the lateral and front walls (filter “Append Datasets”);
7. remove the faces useless in the SPH simulation (feature “select elements with polygone” with selection inversion; filter “Extract Selection”);
8. check and possibly flip the face normals (filter “Generate Surface Normals” with “Flip Normals” and “splitting”);
9. visually check the weir design and save it in “.ply” format for ply2SPHERA\_perimeter.
   1. Modification of the spatial resolution of the DEM-DTM

The following steps can be finally executed to possibly modify the spatial resolution of the DEM-DTM already obtained:

1. execute again the tool DEM2xyz with a different spatial resolution factor considering the DEM-DTM already elaborated as an input file;
2. treat with Paraview the DEM2xyz “.xyz” output file;
   1. erase the points with *z*=-999m;
   2. apply the filter “Delaunay2D”;
   3. erase manually (by means of the Graphic User Interface) the few faces linking points unavailable in the input DEM-DTM;
   4. translate the DEM-DTM according to the origin of the Cartographic System;
   5. check the DEM-DTM normals;
   6. save the DEM-DTM at the new spatial resolution as a “.ply” output file;
3. elaborate the “.ply” DEM-DTM file with ply2SPHERA\_perimeter;
4. add the ply2SPHERA\_perimeter output files among the input files of SPHERA (sections “faces” and “vertices”).

In case Delaunay cannot be applied to the whole DEM-DTM, then it is safer to come back to the previous point on DEM2xyz to partition the DEM output instead of cutting it in Paraview; notice that one cannot merge the DEM-DTM files after the interpolation because one would need a unique DEM-DTM input file for DEM2xyz, otherwise the overlapping points would not be equal due to different interpolation results at boundaries.

* 1. SnappyHexMesh

In case the boundary treatment method of Sec.8 is used, SnappyHexMesh (OpenFOAM, OpenCFD Ltd, 2021, [71]) is used as a surface grid generator for the initial positioning grid of the DB-SPH elements.

* 1. ply2SPHERA\_perimeter

The numerical tool ply2SPHERA\_perimeter (RSE SpA, 2021, [96]) 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.

“ply2SPHERA\_perimeter v.2.0” (RSE SpA) is a minor pre-processing tool of the SPH code SPHERA v.8.0 (RSE SpA). It deals with the format conversion from “.ply” to the format of the sections “VERTICES” and “FACES” of SPHERA main input file to describe the perimeter of a 3D zone (for 3D simulations) or a 2D zone (for 2D simulations).

The “.ply” input file is located in the working directory. Here, there is also the main input file of ply2SPHERA\_perimeter v.2.0: ply2SPHERA\_perimeter.inp. This is composed by 3 lines, with the following structure:

<integer\_1> <integer\_2> <integer\_3> ! perimeter\_first\_vertex\_ID (ID of the first vertex of the perimeter); perimeter\_first\_face\_ID (ID of the first face of the perimeter); perimeter\_ID (ID of the perimeter)

<real> ! z\_offset (offset of the z-coordinate)

<character> ! ply\_file\_name (name of the .ply input file)

The tool ply2SPHERA\_perimeter (2021, [96]) treats “.ply” files with attribute. Alternatively, some program units of SPHERA (i.e., “Import\_ply\_meshes”, “CLC\_preprocessing”) manage “.ply” input files without attributes.

* 1. Paraview: procedures to pre-process the CORINE Land Cover (CLC) maps

Paraview v.5.9.0 (or later version) is mandatory to manage the input CLC “.shp” file. The following procedure might be applied:

1. Read the “.shp” file in the original Cartographic reference system;
2. Apply the filter “Transform” to switch to SPHERA reference system (*x*/*y* offset reduction);
3. Apply the filter “Clips” to cut SPHERA domain;
4. Save a screenshot plotting the first CLC category (vector map);
5. Save a screenshot plotting the CLC polygon IDs (vector map);
6. It is useful to save the CLC “.vtu” file only to show the CLC polygon boundaries. However, the colours of the boundaries of the CLC polygons are not representative: for each segment there are actually two overlapping segments (one per adjacent polygon) but only one segment is visible (normally, the one associated with the external polygon) and it seems unfeasible to see the right colour of the internal polygon, even using the opacity options. To extracting the CLC .vtu file, the following information might be useful. Only mandatory data are saved (otherwise the file would be too large to be opened with Paraview): “objectid”, “codice\_num”. The format of the CLC .vtu file reads:

* “objectid”: CLC polygon ID associated with the segment (each actual segment is represented by two twin segments associated with the inner/outer polygon);
* “codice\_num”: CLC code of the polygon associated with the segment;
* “offsets”: last-point ID for each segment (called “cell”);
* “connectivity”: point-ID list for each segment (called “cell”);
* “Points”: list of vertex coordinates (triplets *xyz*).

The CLC “.vtu” file file cannot be used to reproduce the “.ply” and “.txt” files for SPHERA without using PV v.5.9.0 to read again the “.shp” file. The CLC “.vtu” file cannot be directly provided to SPHERA because a Delaunay filter is mandatory to partition the CLC polygons. “Dealunay2D” can be applied to “.vtu” files, but the attribute of the CLC code is lost.

1. Apply “Delaunay2D” filter to export many “.ply” files (one per CLC polygon). After, only triangles are present in all the “.ply” files (one per CLC polygon). The Linux tool “tar” is mandatory to easily manage the “.ply” files. Save screenshot plotting the CLC polygons (use the visualization mode “surface without edges” because edges are too many).
2. It seems that Paraview cannot wite “.ply” files with scalar attributes even if it can read them. It is mandatory to add a “.txt” file to provide the CLC code. Use the following command chain: “Merge blocks” -> ”Spreadsheet” (visualize only CLC code and CLC polygon ID -Paraview “BlockNumber”-; increasing order to be set on “BlockNumber”) -> “Export” the visible spreadsheet. Saving is very slow (up to ca.20’ for 1M records). Unfortunately, with Paraview v.5.9.0, “Save Data” does not save the block ID of a multi-block dataset (like a CLC file). It would be interesting to try with Paraview v.4.0.1 to possibly save time in exporting the “.txt” file.
3. It seems there is no way to save the ”.shp” file in other formats (by means of “Save Data) during the intermediate stages of the above steps.
   1. SPHERA

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 o granular material and solid bodies (the other application fields the code has been applied to are: fast landslides, sea waves; sloshing fuel tanks; hydroelectric plants). The present documentation file is completely dedicated to this code.

1. Procedure to adapt the input file names and labels:
   1. Copy and paste the template input folder;
   2. Rename the input files depending on the test case (unique Linux command line using the command “rename”);
   3. Replace the input file name within the input files (unique Linux command line using the commands “find” and “sed”);
   4. Erase the lines on the test case label within the input files (unique Linux command line using the comands “find” and “sed”);
   5. Insert the proper lines on the test case label at the proper line number (unique Linux command line using the comands “find” and “sed”);
   6. Erase the lines on the test case description within the input files (unique Linux command line using the comands “find” and “sed”);
   7. Insert the proper lines on the test case description within the input files (unique Linux command line using the comands “find” and “sed”).
   8. Gnuplot

The output files of SPHERA which contain the profiles (1D) of the fluid dynamics variables are visualized by means of Gnuplot (Williams & Kelley, 2021, [99]), which returns the output file in the “.eps” format.

* 1. GSView

These files are read by GSView (Ghostgum Software Pty Ltd, 2021, [100]) and converted in the “.png” format.

* 1. Grid Interpolator: post-processing of the 2D output fields of SPHERA

The output files of SPHERA which contains the synthetic fluid dynamics fields need a following elaboration by means of Grid Interpolator (RSE SpA, 2021, [97]).

The software tool Grid Interpolator (RSE SpA, 2021, [97]), improved during this study, executes Shepard interpolations to any 3D input field with possible application of a despiking procedure. Grid Interpolator (RSE SpA, 2021, [97]) reads several DTM input files. Their overlapping areas have different grid points depending on the file. The outline of each input DTM is irregular, and its edges are not aligned with the Cartesian axes. Grid Interpolator (RSE SpA, 2021, [97]) produces a unique output DTM with variable spatial resolution. This is partitioned in adjacent DTM files with shared boundaries and without overlapping areas. Reference system conversions apply to switch from the local SPH reference system to the cartographic or the geographic reference system. Grid Interpolator (RSE SpA, 2021, [97]) is also used to post-process the 2D fields of the specific flow rate, the depth-average speed and the water depth.

Grid Interpolator v.2.0 (RSE SpA, 2021, [97]) reads a 3D field of values from an input grid and interpolates them on an output grid with a different spatial resolution. The input file is a xyz file (with two additional ad-hoc lines at the beginning). The output field is available in both the file formats xyz and DEM. This tool is also useful to post-process the 2D fields of the maximum specific flow rate and the maximum water depth as estimated by SPHERA v.9.0.0 (RSE SpA).

Two additional lines are reported at the beginning of the xyz input file “input\_field.prn”, as in the following example:

“

n\_points\_in x\_min y\_min z\_min x\_max y\_max z\_max dx\_out dy\_out dz\_out 21822 1152.77000 71.07100 0.00000 25779.70000 9926.20000 1.00000 9.47609 9.47609 1.00000

“

with the following parameter definition:

n\_points\_in: number of points in the input file;

x\_min: minimum x-coordinate;

y\_min: minimum y-coordinate;

z\_min: minimum z-coordinate;

x\_max: maximum x-coordinate;

y\_max: maximum y-coordinate;

z\_max: maximum z-coordinate;

dx\_out: spatial resolution of the output field along the x-axis direction;

dy\_out: spatial resolution of the output field along the y-axis direction;

dz\_out: spatial resolution of the output field along the z-axis direction.

The format of the first additional line does not alter the tool execution.

Fortran format specifier of the second additional line is ‘(i12,9(g12.5))’ .

The conversion from cartographic to geographic horizontal coordinates follow the same assumptions reported in DEM2xyz (RSE SpA). Here the linear conversion formula is inverted, with no need to express the latitude and longitude increments (already computed within the numerical chain of SPHERA by DEM2xyz):

|  |  |
| --- | --- |
|  | (13.2) |

* 1. Paraview: post-processing and visualization of the 2D and 3D output fields of SPHERA

Paraview shows the (2D and 3D) fluid dynamics fields produced by SPHERA and returns the associated image files.

* 1. Image Magick

The images above are concatenated in “.gif” animations by means of Image Magick (ImageMagick Studio LLC, 2021, [101]).

* 1. Virtual Dub

The compression of these animations, necessary in case the concatenation involves many files, is carried out by means of Virtual Dub (Avery Lee, 2021, [102]), which returns an “.avi” video output file.

# SPHERA acknowledgments

SPHERA v.9.0.0 (RSE SpA) is realised 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).

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 (Project Manager Antonella Frigerio);
* 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 April 21, 2016. Reference project: ‘A.5 - Sicurezza e vulnerabilità del sistema elettrico’, Project Manager Antonella Frigerio.
* under the Contract Agreement between RSE S.p.A. and the Ministry of Economic Development - General Directorate for the Electricity Market, Renewable Energy and Energy Efficiency, Nuclear Energy in compliance with the Decree of April 16, 2018; Project: “2.5 Modelli e strumenti di intervento, anche preventivo, per la difesa e il miglioramento della sicurezza e della resilienza delle reti” - Ricerca di Sistema (2.5 Models and action tools for the safety and resilience of the power grids - Research on the Italian Energy System); Project Manager: Francesco Apadula (formerly Antonella Frigerio); Agreement between the Italian Ministry of Economic Development and RSE SpA 2019-2021.

“We acknowledge the CINECA award under the ISCRA initiative, for the availability of High Performance Computing resources and support”: SPHERA simulations have been supported by the following instrumental-funding HPC projects: HSPHER22, HSPHER21, HSPHER20, HPCEFM7b, HPCEFM17, NMTFEPRA, HPCEFM16, HPCEFM15, HSPHMI14. TNMRA01, HPCEFM18, HSPHERA9, HSPHCS9, HPCNHLW1, HSPHER9b, HPCNHLW2.

The FOSS versions of SPHERA have been supported in RSE and promoted by: the Department Director Michel de Nigris; the Deputy Director, Project Manager and Research Team Manager Antonella Frigerio; the Project Manager and Research Team Manager Francesco Apadula (since 2020); the Project Managers Stefano Maran & Giovanni Pirovano (since 2022); the Research Team Managers Guido Pirovano (2016-2020) and Massimo Meghella (2015-2016).

# 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.

# 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.

# References

|  |  |
| --- | --- |
| [1] | A. Amicarelli, B. Kocak, S. Sibilla and J. Grabe, “A 3D Smoothed Particle Hydrodynamics model for erosional dam-break floods,” *International Journal of Computational Fluid Dynamics,* vol. 31, no. 10, pp. 413-434, 2017. |
| [2] | A. Amicarelli, R. Albano, D. Mirauda, G. Agate, A. Sole and R. Guandalini, “A Smoothed Particle Hydrodynamics model for 3D solid body transport in free surface flows,” *Computers & Fluids,* vol. 116, p. 205–228; DOI 10.1016/j.compfluid.2015.04.018, 2015. |
| [3] | A. Amicarelli, G. Agate and R. Guandalini, “A 3D Fully Lagrangian Smoothed Particle Hydrodynamics model with both volume and surface discrete elements,” *International Journal for Numerical Methods in Engineering,* vol. 95, pp. 419–450, DOI: 10.1002/nme.4514, 2013. |
| [4] | S. Manenti, S. Sibilla, M. Gallati, G. Agate and R. Guandalini, “SPH Simulation of Sediment Flushing Induced by a Rapid Water Flow,” *Journal of Hydraulic Engineering ASCE,* vol. 138, no. 3, pp. 272-284, 2012. |
| [5] | A. Di Monaco, S. Manenti, M. Gallati and S. Sibilla, “SPH modeling of solid boundaries through a semi-analytic approach,” *Engineering Applications of Computational Fluid Mechanics,* vol. 5, no. 1, pp. 1-15, 2011. |
| [6] | SPHERA (RSE SpA), “https://github.com/AndreaAmicarelliRSE/SPHERA,” 2020. [Online]. |
| [7] | T. a. C.-). SPHERIC (SPH scientific and industrial community affiliated to ERCOFTAC -European Research Community On Flow. [Online]. Available: http://spheric-sph.org/ , http://spheric-sph.org/sph-projects-and-codes. [Accessed 2019]. |
| [8] | J. Monaghan and J. Lattanzio, “A refined method for astrophysical problems,” *Astronomy and Astrophysics,* vol. 149, pp. 135-143, 1985. |
| [9] | S. Adami, X. Hu and N. Adams, “A generalized wall boundary condition for smoothed particle hydrodynamics,” *Journal of Computational Physics,* vol. 231, p. 7057–7075, 2012. |
| [10] | M. R. Hashemi, R. Fatehi and M. Manzari, “A modified SPH method for simulating motion of rigid bodies in Newtonian fluid flows,” *International Journal of Non-Linear Mechanics,* vol. 47, p. 626–638, 2012. |
| [11] | F. Macia, L. Gonzalez, J. Cercos-Pita and A. Souto-Iglesias, “A Boundary Integral SPH Formulation - Consistency and Applications to ISPH and WCSPH-,” *Progress of Theoretical Physics,* vol. 128, no. 3, pp. 439-462, 2012. |
| [12] | A. Mayrhofer, B. Rogers, D. Violeau and M. Ferrand, “Investigation of wall bounded flows using SPH and the uni fied semi-analytical wall boundary conditions,” *Computer Physics Communications,* vol. 184, p. 2515–2527, 2013. |
| [13] | M. Ferrand, D. Laurence, B. Rogers, D. Violeau and C. Kassiotis, “Unified semi-analytical wall boundary conditions for inviscid laminar or turbulent flows in the meshless SPH method,” *International Journal for Numerical Methods in Fluids,* vol. 71, no. 4, pp. 446-472, 2013. |
| [14] | R. Vacondio, B. Rogers, P. Stansby and P. Mignosa, “SPH Modeling of Shallow Flow with Open Boundaries for Practical Flood Simulation,” *J. Hydraul. Eng.,* vol. 138, no. 6, pp. 530-541, 2012. |
| [15] | A. Souto-Iglesias, L. Delorme, L. Pérez-Rojas and S. Abril-Pérez, “Liquid moment amplitude assessment in sloshing type problems with smooth particle hydrodynamics,” *Ocean Engineering,* vol. 33, p. 1462–1484, 2006. |
| [16] | M. Patel, R. Vignjevic and J. Campbell, “An SPH Technique for Evaluating the Behaviour of Ships In Extreme Ocean Waves,” *International Journal of Maritime Engineering,* vol. 151, pp. 39-47, 2009. |
| [17] | M. Antuono, A. Colagrossi, S. Marrone and C. Lugni, “Propagation of gravity waves through a SPH scheme with numerical diffusive terms,” *Computer Physics Communications,* vol. 182, no. 4, pp. 866-877, 2011. |
| [18] | X. Liu, H. Xu, S. Shao and P. Lin, “An improved incompressible SPH model for simulation of wave-structure interaction,” *Computers and Fluids,* vol. 71, pp. 113-123, 2013. |
| [19] | P. Omidvar, P. Stansby and B. Rogers, “Wave body interaction in 2D using smoothed particle hydrodynamics (SPH) with variable particle mass,” *Int. J. Numer. Meth. Fluids,* vol. 68, p. 686–705, 2012. |
| [20] | J.-C. Marongiu, F. Leboeuf, J. Caro and E. Parkinson, “Free surface flows simulations in Pelton turbines using an hybrid SPH-ALE method,” *J. Hydraul. Res.,* vol. 47, p. 40–49, 2010. |
| [21] | D. Kumar, A. Patra, E. Pitman and H. Chi, “Parallel Godunov smoothed particle hydrodynamics (SPH) with improved treatment of Boundary Conditions and an application to granular flows,” *Computer Physics Communications,* vol. 184, no. 10, pp. 2277-2286, 2013. |
| [22] | P. Koukouvinis, J. Anagnostopoulos and D. Papantonis, “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,* vol. 71, no. 9, p. 1152–1177. DOI: 10.1002/fld.3706, 2013. |
| [23] | D. Price, “Smoothed Particle Hydrodynamics and Magnetohydrodynamics,” *J. Comp. Phys.,* vol. 231, no. 3, pp. 759-794, 2012. |
| [24] | J. Kajtar and J. Monaghan, “On the swimming of fish like bodies near free and fixed boundaries,” *European Journal of Mechanics B/Fluids,* vol. 33, p. 1–13, 2012. |
| [25] | J. Vila, “On particle weighted methods and Smooth Particle Hydrodynamics,” *Mathematical Models and Methods in Applied Sciences,* vol. 9, no. 2, pp. 161-209, 1999. |
| [26] | J. Monaghan, “Smoothed particle hydrodynamics,” *Rep. Prog. Phys.,* vol. 68, p. 1703–1759, 2005. |
| [27] | M. Basa, N. Quinlan and M. Lastiwka, “Robustness and accuracy of SPH formulations for viscous flow,” *International Journal for Numerical Methods in Fluids,* vol. 60, no. 10, pp. 1127-1148, 2009. |
| [28] | P. Davidson, Turbulence: An introduction for scientists and engineers, USA: Oxford University Press, 2015. |
| [29] | A. Pelliccioni, P. Monti, C. Gariazzo and G. Leuzzi, “Some characteristics of the urban boundary layer above Rome, Italy, and applicability of Monin-Obukhov similarity,” *Environmental Fluid Mechanics,* vol. 12, no. 5, pp. 405-428, 2012. |
| [30] | K. Jancewicz and M. Szymanowski, “The Relevance of Surface Roughness Data Qualities in Diagnostic Modeling of Wind Velocity in Complex Terrain: A Case Study from the Snieznik Massif (SW Poland),” *Pure and Applied Geophysics,* vol. 174, no. 2, pp. 569-594, 2017. |
| [31] | L. Da Peppo and C. Datei, Fognature, Padova (Italy): Libreria Cortina, 2003. |
| [32] | A. Chen, B. Evans, S. Djordjevic and D. Savic, “A coarse-grid approach to represent building blockage effects in 2D urban flood modelling,” *J. Hydrol.,* vol. 426, pp. 1-16, 2012. |
| [33] | B. Kim, B. Sanders, J. Famiglietti and V. Guinot, “Urban flood modeling with porous shallow-water equations: A case study of model errors in the presence of anisotropic porosity,” *Journal of Hydrology,* vol. 523, pp. 680-692, 2015. |
| [34] | M. Di Risio and P. Sammarco, “Effects of floaters on the free surface profiles of river flows,” *Environmental Fluid Mechanics,* vol. 20, no. 3, pp. 527-537, 2020. |
| [35] | A. Baruah and A. Sarma, “A fully coupled two-dimensional flow—vegetation routing model in open channel flow,” *Environmental Fluid Mechanics,* vol. 21, no. 5, pp. 1091-1117, 2021. |
| [36] | L. Van Rijn, “The prediction of Bed-Forms and Alluvial Roughness, Report,” Delft Hydraulics laboratory, The Netherlands, 1982. |
| [37] | M.-W. Ge, L. Fang and Y.-Q. Liu, “Drag reduction of wall bounded incompressible turbulent flow based on active dimples/pimples,” *Journal of Hydrodynamics,* vol. 29, no. 2, pp. 261-271, 2017. |
| [38] | C. Montuori, Complementi di idraulica, Napoli: Liguori Editore, pp.1-465, 1997. |
| [39] | B. Krank, Wall Modeling via Function Enrichment for Computational Fluid Dynamics, Munich (Germany): Technical University of Munich, 2019. |
| [40] | M. Paggi, A. Amicarelli and P. Lenarda, “SPH modelling of hydrodynamic lubrication: laminar fluid flow - structure interaction with no-slip conditions for slider bearings,” *Computational Particle Mechanics,* vol. 8, pp. 665–679; https://doi.org/10.1007/s40571-020-00362-1, 2021. |
| [41] | “SRTM3/DTED1 (USGS),” 2020. [Online]. Available: http://earthexplorer.usgs.gov/. |
| [42] | B. Kosztra, G. Büttner, G. Hazeu and S. Arnold, “Updated CLC illustrated nomenclature guidelines,” European Environment Agency, European Topic Centre on Urban, land and soil systems; ETC/ULS Service Contract No 3436/R0-Copernicus/EEA.57441, Task 3, D3.1 – Part 1., 2019. |
| [43] | “GDAL,” OSGEO, 2021. [Online]. Available: https://github.com/OSGeo/gdal. |
| [44] | Paraview (Kitware). [Online]. Available: https://github.com/Kitware/ParaView. [Accessed 15 July 2021]. |
| [45] | CFX (Ansys). [Online]. Available: http://www.ansys.com/products/fluids/ansys-cfx. [Accessed 2020]. |
| [46] | P. Swamee, “Generalized rectangular weir equations,” *Journal of Hydraulic Engineering,* vol. 114, no. 8, pp. 945-949, 1988. |
| [47] | A. Amicarelli, S. Manenti, R. Albano, G. Agate, M. Paggi, L. Longoni, D. Mirauda, L. Ziane, G. Viccione, S. Todeschini, A. Sole, L. Baldini, D. Brambilla, M. Papini, M. Khellaf, B. Tagliafierro, L. Sarno and G. Pirovano, “SPHERA v.9.0.0: a Computational Fluid Dynamics research code, based on the Smoothed Particle Hydrodynamics mesh-less method,” *Computer Physics Communications,* vol. 250, p. 107157, 2020. |
| [48] | M. Paggi, A. Amicarelli and P. Lenarda, “SPH modelling of hydrodynamic lubrication: laminar fluid flow - structure interaction with no-slip conditions for slider bearings,” *Computational Particle Mechanics,* vol. 8, pp. 665–679; https://doi.org/10.1007/s40571-020-00362-1, 2021. |
| [49] | A. Amicarelli, E. Abbate and A. Frigerio, “SPH modelling of a dike failure with detection of the landslide sliding surface and damage scenarios for an electricity pylon,” *International Journal of Computational Fluid Dynamics,* p. preprint, 2021. |
| [50] | A. Amicarelli and G. Agate, “SPH modelling of granular flows,” in *9th SPHERIC ERCOFTAC International ERCOFTAC Workshop, 17-24*, Paris, 2014. |
| [51] | L. Armstrong, S. Gu and K. Luo, “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,* vol. 53, no. 21-22, pp. 4949-4959, 2010. |
| [52] | A. Colagrossi and M. Landrini, “Numerical simulation of interfacial flows by smoothed particle hydrodynamics,” *Journal of Computational Physics,* vol. 191, no. 2, pp. 448-475, 2003. |
| [53] | D. Schaeffer, “Instability in the Evolution Equations Describing Incompressible Granular Flow,” *Journal of Differential Equations,* vol. 66, pp. 19-50, 1987. |
| [54] | R. W. Randles and L. D. Libertsky, “Smoothed Particle Hydrodynamics. Some recent improvements and applications,” *Comput. Methods Appl. Mech. Engrg.,* vol. 139, pp. 375-408, 1996. |
| [55] | V. Kumaran, “Kinetic theory for sheared granular flows,” *C. R. Physique,* vol. 16, pp. 51-61, 2015. |
| [56] | J. Monaghan, “Smoothed Particle Hydrodynamics,” *Annu. Rev. Astron. Astrophys,* vol. 30, pp. 543-74, 1992. |
| [57] | L. Van Rijn, Principles of sediment transport in rivers, estuaries, and coastal seas, Aqua Publications, 1993. |
| [58] | G. Seminara, L. Solari and G. Parker, “Bed load at low Shields stress on arbitrarily sloping beds: Failure of the Bagnold hypothesis,” *Water Resour. Res.,* vol. 38, no. 11, pp. 1249, doi:10.1029/2001WR000681, 2002. |
| [59] | F. Morrison, An Introduction to Fluid Mechanics, New York: Cambridge University Press, 2013. |
| [60] | A. Amicarelli, G. Agate, B. Stefanova, S. Sibilla and J. Grabe, “A SPH model for dike overtopping and dam liquefaction with bed-load transport, bottom drag and mobile boundaries,” in *11th SPHERIC ERCOFTAC International Workshop SPHERIC 2016 – ERCOFTACT Conference, pp.424-431, Garching (Munich, Germany); ISBN: 9783000533587*, 2016. |
| [61] | A. Amicarelli, S. Manenti and M. Paggi, “SPH modelling of dam-break floods, with damage assessment to electrical substations,” *International Journal of Computational Fluid Dynamics,* online. |
| [62] | Hazus-MH, “Technical Manual, Multi-hazard Loss Estimation Methodology, Flood Model,” Department of Homeland Security, Federal Emergency Management Agency, Mitigation Division, Washington D.C., USA, 2011. |
| [63] | M. Holmes, Water infrastructure vulnerability due to dependency on third party infrastructure sectors, School of Civil Engineering and Geosciences, Newcastle University, May 2015; Doctorate thesis, 2015. |
| [64] | M. Crawford and S. Seidel, “Weathering the storm: building business resilience to climate change,” Center for climate and energy solutions, Arlington, VA, USA, 2013. |
| [65] | M. Chow, L. Taylor and M. Chow, “Time of outage restoration analysis in distribution systems,” in *IEEE Transactions on Power Delivery, 11(3):1652-1658*, 1996. |
| [66] | P. Maliszewski and C. Perrings, “Factors in the resilience of electrical power distribution infrastructures,” *Applied Geography,* vol. 32, pp. 668-679, 2012. |
| [67] | D. Reed, “Electric utility distribution analysis for extreme winds,” *Journal of wind engineering and industrial aerodynamics,* vol. 96, no. 1, pp. 123-140, 2008. |
| [68] | MATTM, “Ministero dell’Ambiente e della Tutela del Territorio e del Mare (MATTM); 2018; Infrastrutture elettriche presenti sul territorio italiano,” 2020. [Online]. Available: 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. |
| [69] | A. Amicarelli and G. Agate, “A 3D SPH integrated model for granular flows with transport of solid bodies: model couplings and enhanced boundary treatment based on surface wall elements,” in *10th SPHERIC ERCOFTAC International Workshop SPHERIC 2015 – ERCOFTACT Conference, 23-30; ISBN: 978-88-7847-487-1*, Parma (Italy), 2015. |
| [70] | A. Amicarelli and M. Paggi, “SPH modelling for fluid-structure interactions with complex 3D surfaces: catastrophic dam-break and flood-control work on a real topography and hydrodynamic lubrication on rough surfaces,” in *14th SPHERIC ERCOFTAC International Workshop SPHERIC 2019 - ERCOFTAC Conference, pp.9-16. ISBN: 978-0-902746-44-2*, Exeter University (UK), 2019. |
| [71] | OpenFOAM (OpenCFD Ltd), “https://github.com/OpenFOAM/OpenFOAM-dev,” 2021. [Online]. Available: https://github.com/isoAdvector/isoAdvector. |
| [72] | S. Manenti, A. Amicarelli and S. Todeschini, “WCSPH with Limiting Viscosity for Modeling Landslide Hazard at the Slopes of Artificial Reservoir,” *Water,* vol. 10, no. 4, pp. 515, doi:10.3390/w10040515, 2018. |
| [73] | A. Amicarelli and G. Agate, “Modellazione fluidodinamica SPH per inondazioni con trasporto di corpi solidi, dam break per dighe in terra e moti oscillatori in serbatoi: sviluppi numerici e validazioni,” project report (Ricerca di Sistema), RSE protocol 14001808, 2014. |
| [74] | A. Amicarelli, “SPH modelling of an experimental urban dam-break flood and a rectangular lateral side weir for flood-control works,” in *13th SPHERIC ERCOFTAC International Workshop SPHERIC 2018 - ERCOFTACT Conference, pp.103-109. ISBN: 978-1-908358-59-2*, Galway (Ireland), 2018. |
| [75] | gedit, GNOME Foundation, [Online]. Available: https://wiki.gnome.org/Apps/Gedit. |
| [76] | gfortran (GNU, Free Software Foundation), “http://gcc.gnu.org/fortran/,” 2021. [Online]. |
| [77] | “GNU Make,” GNU, Free Software Foundation, [Online]. Available: https://www.gnu.org/software/make/. |
| [78] | “gdb,” GNU, Free Software Foundation, [Online]. Available: https://www.gnu.org/software/gdb/. |
| [79] | “gprof,” GNU, Free Software Foundation, 2021. [Online]. Available: http://sourceware.org/binutils/docs/gprof/. |
| [80] | “Valgrind,” Valgrind Developers, 2021. [Online]. Available: http://valgrind.org/. |
| [81] | “Scalasca,” Forschungszentrum Jülich & TU Darmstadt, 2021. [Online]. Available: http://www.scalasca.org/. |
| [82] | “Git,” Torvalds et al., 2021. [Online]. Available: https://github.com/git/git. |
| [83] | “GitHub,” GitHub Inc., 2021. [Online]. Available: www.github.com. |
| [84] | “Free Software Foundation,” 2021. [Online]. Available: http://www.fsf.org/. |
| [85] | Intel Fortran Classic (ifort), “https://software.intel.com/content/www/us/en/develop/tools/oneapi/components/fortran-compiler.html#gs.aiojbj,” 2021. [Online]. |
| [86] | “idb,” Intel Corporation, 2021. [Online]. Available: www.intel.com. |
| [87] | “cpuinfo,” Intel Corporation, 2021. [Online]. Available: www.intel.com. |
| [88] | “VTUNE,” Intel Corporation, 2021. [Online]. Available: www.intel.com. |
| [89] | “Advisor,” Intel Corporation, 2021. [Online]. Available: www.intel.com. |
| [90] | TotalView, Rogue Wave Software, 2018. [Online]. Available: https://www.roguewave.com/products-services/totalview. |
| [91] | “Engauge Digitizer (Mitchell et al.),” [Online]. Available: https://github.com/markummitchell/engauge-digitizer. [Accessed 15 July 2019]. |
| [92] | “SRTM3/DTED1 (USGS),” 2014. [Online]. Available: http://earthexplorer.usgs.gov/. |
| [93] | M. Rexer and C. Hirt, “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,* vol. 61, no. 2, pp. 213-226, DOI: 10.1080/08120099.2014.884983, 2014. |
| [94] | Paraview (Kitware). [Online]. Available: https://github.com/Kitware/ParaView. [Accessed 15 July 2019]. |
| [95] | DEM2xyz (RSE SpA), “https://github.com/AndreaAmicarelliRSE/DEM2xyz,” 2020. [Online]. |
| [96] | “ply2SPHERA\_perimeter (RSE SpA),” 2021. [Online]. Available: https://github.com/AndreaAmicarelliRSE/ply2SPHERA\_perimeter. |
| [97] | Grid\_Interpolator (RSESpA), “https://github.com/AndreaAmicarelliRSE/Grid\_Interpolator,” 2020. [Online]. |
| [98] | E. Plate, Urban Climates and Urban Climate Modelling, Kluwer Academic Publishers, 1995. |
| [99] | Gnuplot (Williams & Kelley), “http://www.gnuplot.info/,” 2021. [Online]. |
| [100] | “GSView,” Ghostgum Software Pty Ltd, 2021. [Online]. Available: https://www.ghostscript.com/. |
| [101] | “Image Magick,” ImageMagick Studio LLC, 2021. [Online]. Available: https://www.imagemagick.org. |
| [102] | “Virtual Dub,” Avery Lee, 2021. [Online]. Available: http://www.virtualdub.org/. |
| [103] | B. Cushman-Roisin, “Environmental Fluid Mechanics,” 2019. [Online]. Available: http://www.dartmouth.edu/~cushman/books/EFM.html. [Accessed 21 May 2019]. |
| [104] | D. Citrini and G. Noseda, Idraulica (seconda edizione), Milano: C.E.A. Casa Editrice Ambrosiana, pp.1-463, 1987. |
| [105] | P. Bandyopadhyay, “Rough-wall turbulent boundary layers in the transition regime,” *Journal of Fluid Mechanics,* vol. 180, pp. 231-266, 1987. |