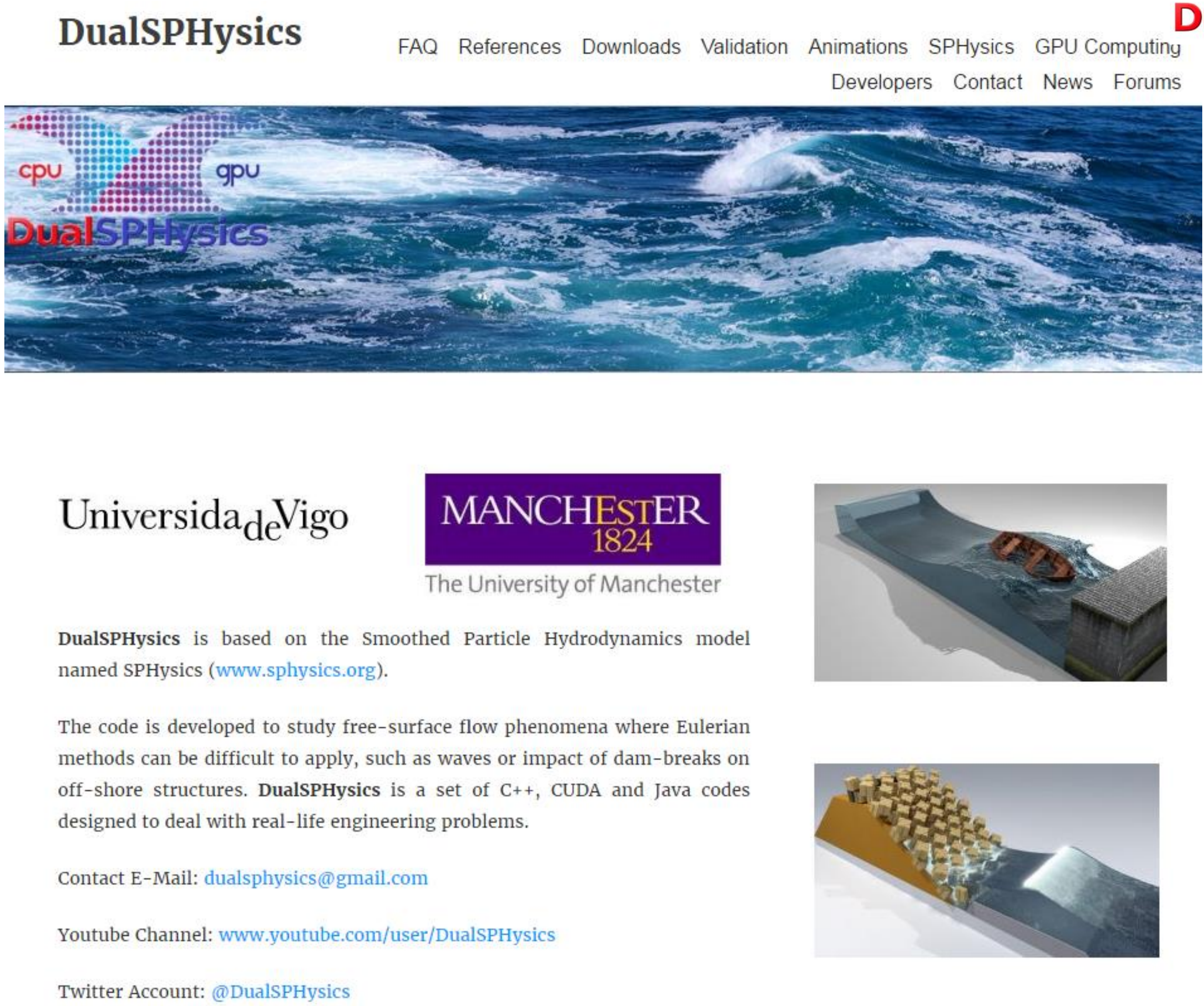
Graphical User Interface for SPH codes: DesignSPHysics

A. Vieira, O. García-Feal, J.M. Domínguez,
A.J.C. Crespo, M. Gómez-Gesteira
EPHYSLAB, Environmental Physics Laboratory
Universidade de Vigo

DualSPHysics project



DualSPHysics is based on the Smoothed Particle Hydrodynamics model named SPHysics (www.sphysics.org).

The code is developed to study free-surface flow phenomena where Eulerian methods can be difficult to apply, such as waves or impact of dam-breaks on off-shore structures. **DualSPHysics** is a set of C++, CUDA and Java codes designed to deal with real-life engineering problems.

Contact E-Mail: dualsphysics@gmail.com

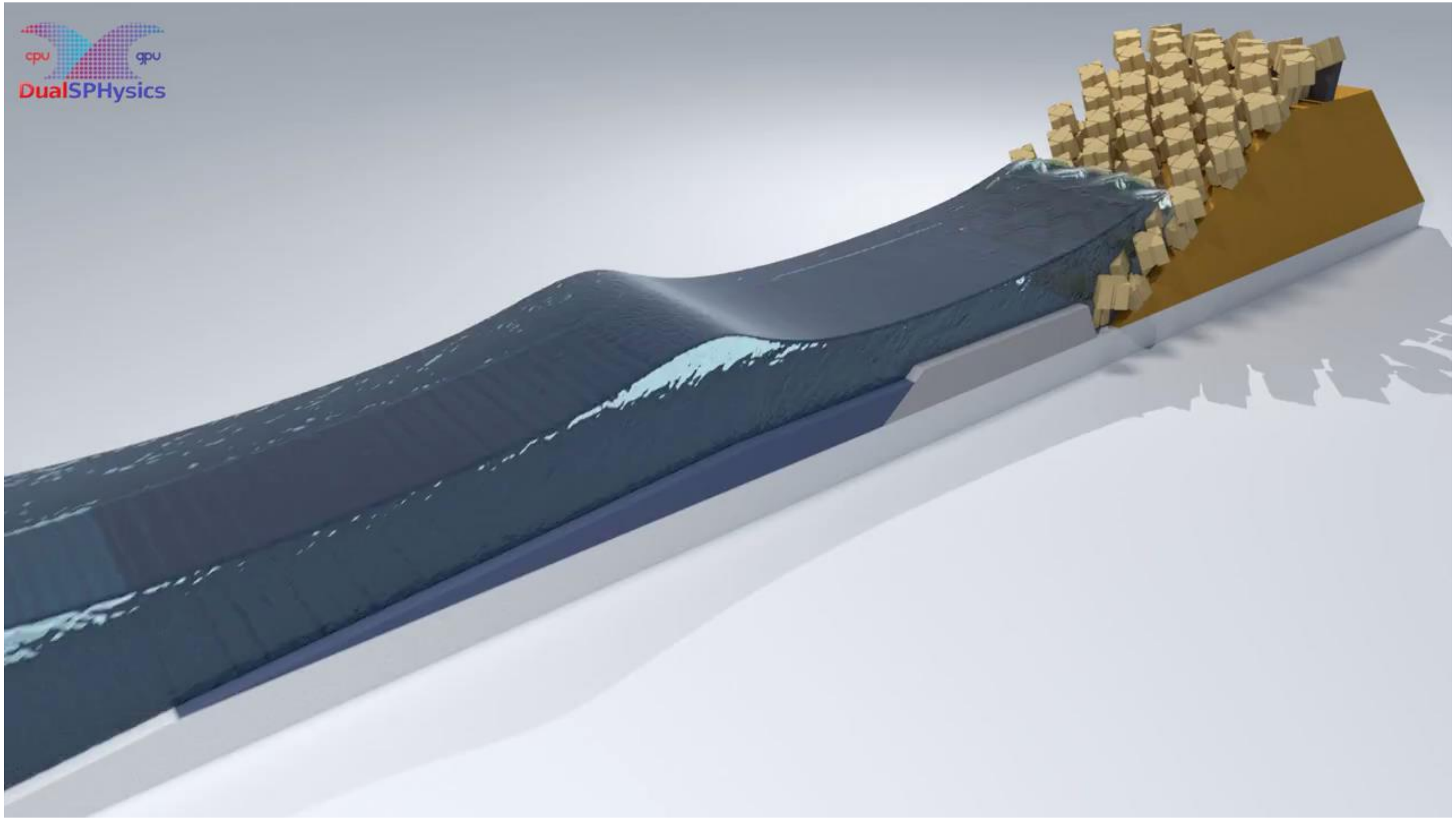
Youtube Channel: www.youtube.com/user/DualSPHysics

Twitter Account: @DualSPHysics

www.dual.sphysics.org

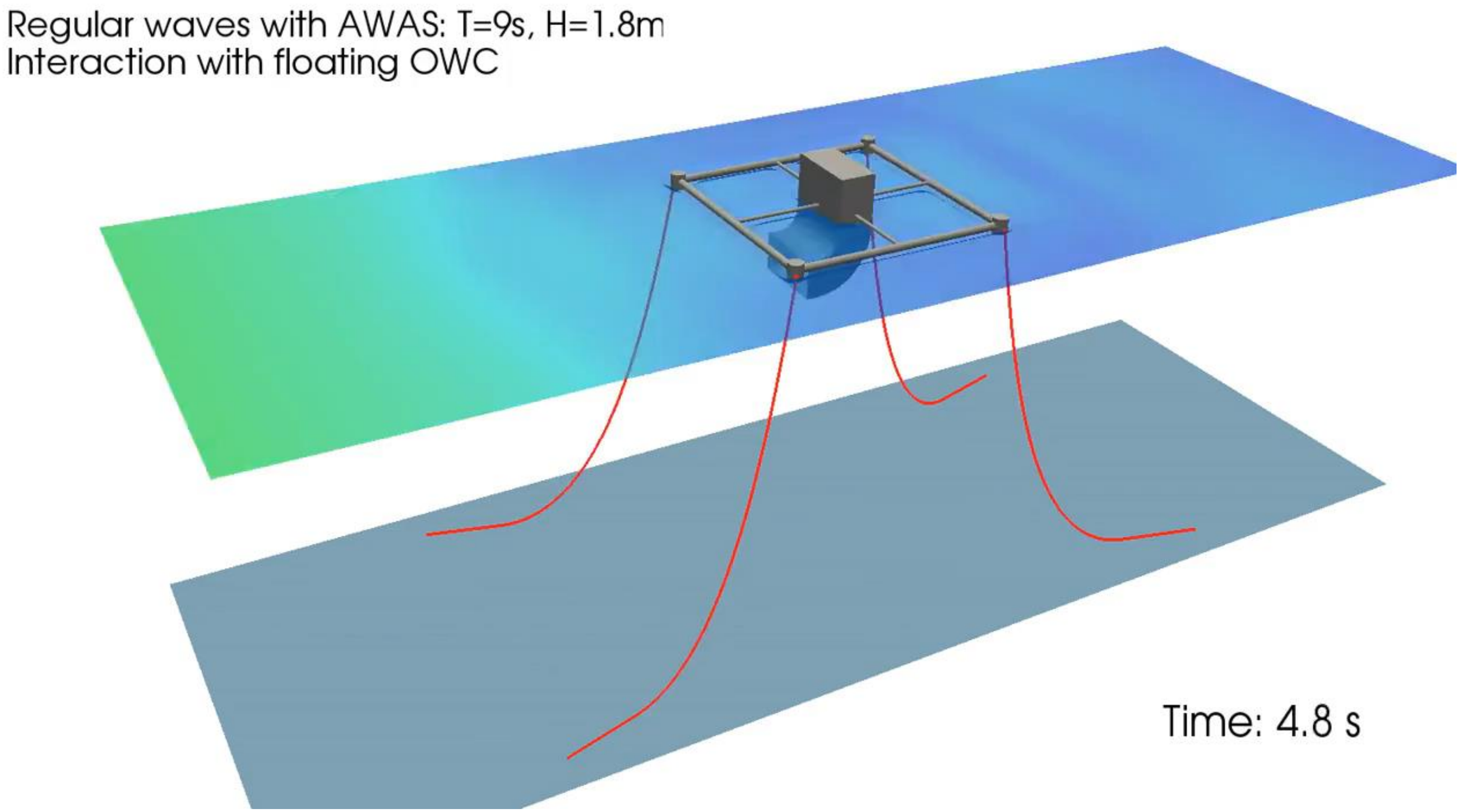
DualSPHysics applications

Study of the run-up in armour block sea breakwater in Zeebrugge



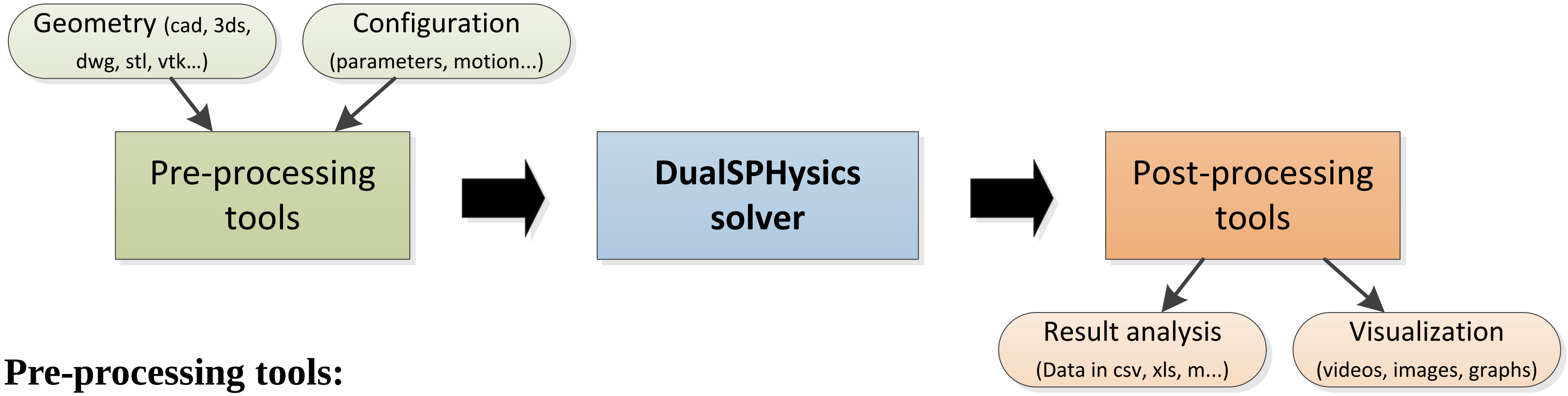
DualSPHysics applications

Numerical design of WECs


Regular waves with AWAS: T=9s, H=1.8m
Interaction with floating OWC
Time: 4.8 s

DualSPHysics project

DSPH project includes:


Geometry (cad, 3ds, dwg, stl, vtk...)
Configuration (parameters, motion...)
Pre-processing tools
DualSPHysics solver
Post-processing tools
Result analysis (Data in csv, xls, m...)
Visualization (videos, images, graphs)

Pre-processing tools:

- Converts geometry into particles.
- Provides configuration for simulation.

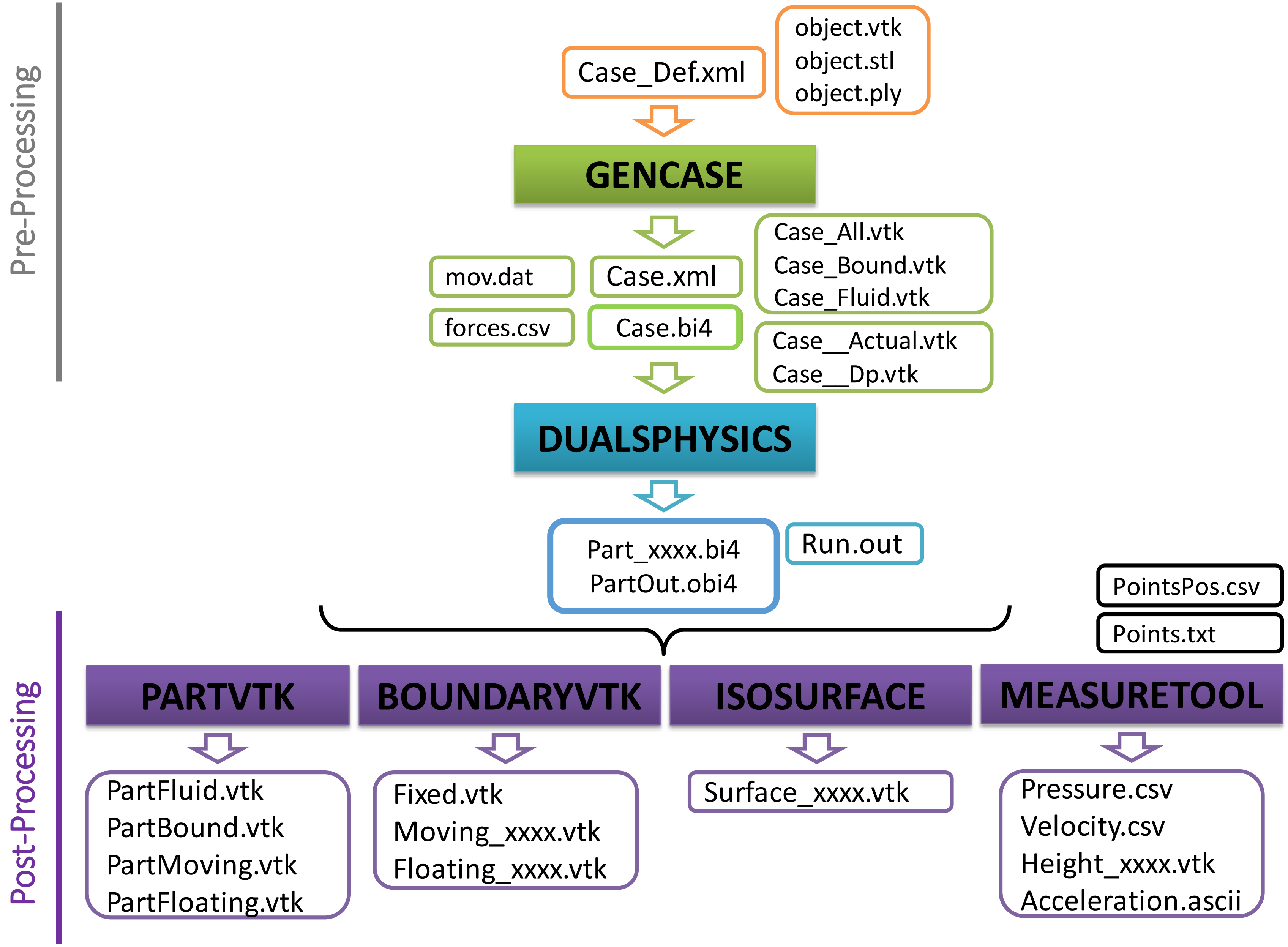
DualSPHysics solver:

- Runs simulation using SPH particles.
- Obtains data simulation for time intervals.

Post-processing tools:

- Calculates magnitudes using particle data.
- Visualisation starting form SPH particles.

Input & output files



Pre-Processing

Case_Def.xml

object.vtk
object.stl
object.ply

GENCASE

mov.dat

forces.csv

Case.xml

Case.bi4

Case_All.vtk
Case_Bound.vtk
Case_Fluid.vtk

Case__Actual.vtk
Case__Dp.vtk

DUALSPHYSICS

Part_xxxx.bi4
PartOut.obi4

Run.out

PointsPos.csv

Points.txt

Post-Processing

PARTVTK

PartFluid.vtk
PartBound.vtk
PartMoving.vtk
PartFloating.vtk

BOUNDARYVTK

Fixed.vtk
Moving_xxxx.vtk
Floating_xxxx.vtk

ISOSURFACE

Surface_xxxx.vtk

MEASURETOOL

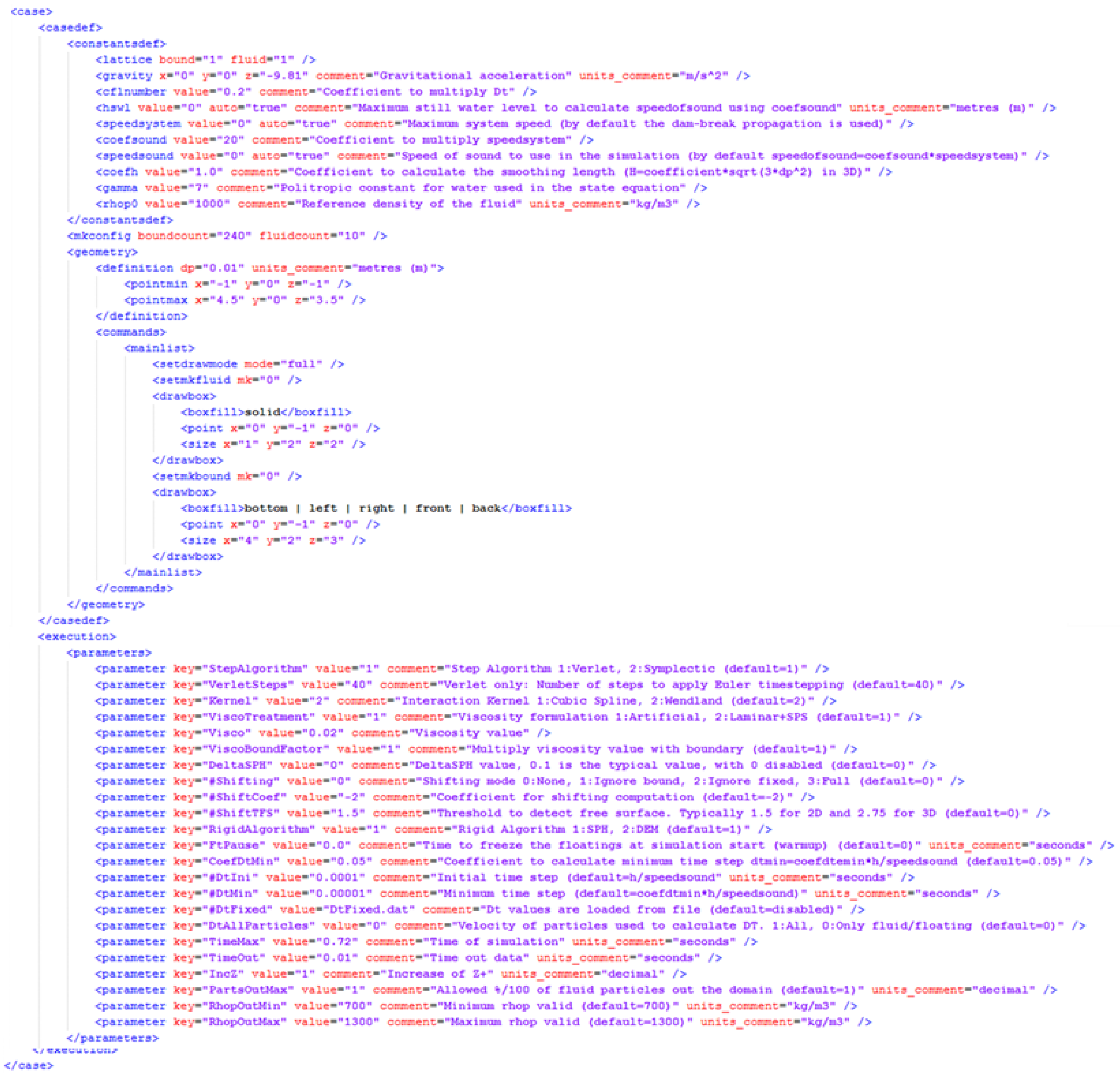
Pressure.csv
Velocity.csv
Height_xxxx.vtk
Acceleration.ascii

XML file

Constant Definition

Geometry Definition

Execution Parameters



```
<case>
    <casedef>
        <constantsdef>
            <lattice bound="1" fluid="1" />
            <gravity x="0" y="0" z="-9.81" comment="Gravitational acceleration" units_comment="m/s^2" />
            <cflnumber value="0.2" comment="Coefficient to multiply Dt" />
            <hswl value="0" auto="true" comment="Maximum still water level to calculate speedofsound using coefsound" units_comment="metres (m)" />
            <speedsystem value="0" auto="true" comment="Maximum system speed (by default the dam-break propagation is used)" />
            <coefsound value="20" comment="Coefficient to multiply speedsystem" />
            <speedsound value="0" auto="true" comment="Speed of sound to use in the simulation (by default speedofsound=coefsound*speedsystem)" />
            <coefh value="1.0" comment="Coefficient to calculate the smoothing length (H=coefficient*sqrt(3*dp^2) in 3D)" />
            <gamma value="7" comment="Politropic constant for water used in the state equation" />
            <rhop0 value="1000" comment="Reference density of the fluid" units_comment="kg/m3" />
        </constantsdef>
        <mkconfig boundcount="240" fluidcount="10" />
        <geometry>
            <definition dp="0.01" units_comment="metres (m)">
                <pointmin x="-1" y="0" z="-1" />
                <pointmax x="4.5" y="0" z="3.5" />
            </definition>
            <commands>
                <mainlist>
                    <setdrawmode mode="full" />
                    <setmkfluid mk="0" />
                    <drawbox>
                        <boxfill>solid</boxfill>
                        <point x="0" y="-1" z="0" />
                        <size x="1" y="2" z="2" />
                    </drawbox>
                    <setmkbound mk="0" />
                    <drawbox>
                        <boxfill>bottom | left | right | front | back</boxfill>
                        <point x="0" y="-1" z="0" />
                        <size x="4" y="2" z="3" />
                    </drawbox>
                </mainlist>
            </commands>
        </geometry>
    </casedef>
    <execution>
        <parameters>
            <parameter key="StepAlgorithm" value="1" comment="Step Algorithm 1:Verlet, 2:Symplectic (default=1)" />
            <parameter key="VerletSteps" value="40" comment="Verlet only: Number of steps to apply Euler timestepping (default=40)" />
            <parameter key="Kernel" value="2" comment="Interaction Kernel 1:Cubic Spline, 2:Wendland (default=2)" />
            <parameter key="ViscoTreatment" value="1" comment="Viscosity formulation 1:Artificial, 2:Laminar+SPS (default=1)" />
            <parameter key="Visco" value="0.02" comment="Viscosity value" />
            <parameter key="ViscoBoundFactor" value="1" comment="Multiply viscosity value with boundary (default=1)" />
            <parameter key="DeltaSPH" value="0" comment="DeltaSPH value, 0.1 is the typical value, with 0 disabled (default=0)" />
            <parameter key="#Shifting" value="0" comment="Shifting mode 0:None, 1:Ignore bound, 2:Ignore fixed, 3:Full (default=0)" />
            <parameter key="#ShiftCoef" value="-2" comment="Coefficient for shifting computation (default=-2)" />
            <parameter key="#ShiftTFS" value="1.5" comment="Threshold to detect free surface. Typically 1.5 for 2D and 2.75 for 3D (default=0)" />
            <parameter key="RigidAlgorithm" value="1" comment="Rigid Algorithm 1:SPH, 2:DEM (default=1)" />
            <parameter key="FtPause" value="0.0" comment="Time to freeze the floatings at simulation start (warmup) (default=0)" units_comment="seconds" />
            <parameter key="CoefDtMin" value="0.05" comment="Coefficient to calculate minimum time step dtmin=coefdtmin*h/speedsound (default=0.05)" />
            <parameter key="#DtIni" value="0.0001" comment="Initial time step (default=h/speedsound" units_comment="seconds" />
            <parameter key="#DtMin" value="0.00001" comment="Minimum time step (default=coefdtmin*h/speedsound)" units_comment="seconds" />
            <parameter key="#DtFixed" value="DtFixed.dat" comment="Dt values are loaded from file (default=disabled)" />
            <parameter key="DtAllParticles" value="0" comment="Velocity of particles used to calculate DT. 1:All, 0:Only fluid/floating (default=0)" />
            <parameter key="TimeMax" value="0.72" comment="Time of simulation" units_comment="seconds" />
            <parameter key="TimeOut" value="0.01" comment="Time out data" units_comment="seconds" />
            <parameter key="IncZ" value="1" comment="Increase of Z+" units_comment="decimal" />
            <parameter key="PartsOutMax" value="1" comment="Allowed %/100 of fluid particles out the domain (default=1)" units_comment="decimal" />
            <parameter key="RhopOutMin" value="700" comment="Minimum rhop valid (default=700)" units_comment="kg/m3" />
            <parameter key="RhopOutMax" value="1300" comment="Maximum rhop valid (default=1300)" units_comment="kg/m3" />
        </parameters>
    </execution>
</case>
```

XML file

How to create new cases?

1. Following existing examples
2. XML Guide
3. Help in the forum or email

…

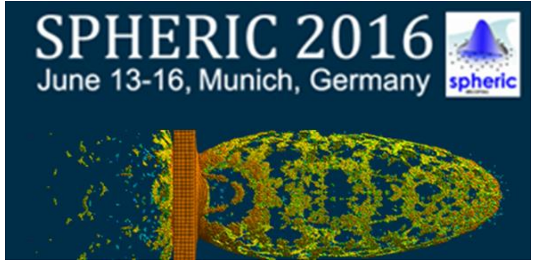
Creation of new XML is difficult for new users and takes time

Since SPH-based simulations are growing in applicability to solve industrial problems…
How to encourage new users to work with SPH code?

GRAPHICAL USER INTERFACE IS KEY

Graphical User Interface

June 2016: SPHERIC Worskhop 2016 in Munich



Panel discussion on “Industrial needs”:
Need of friendly user interface

December 2016:
SPHERIC Steering Committee meeting in Manchester

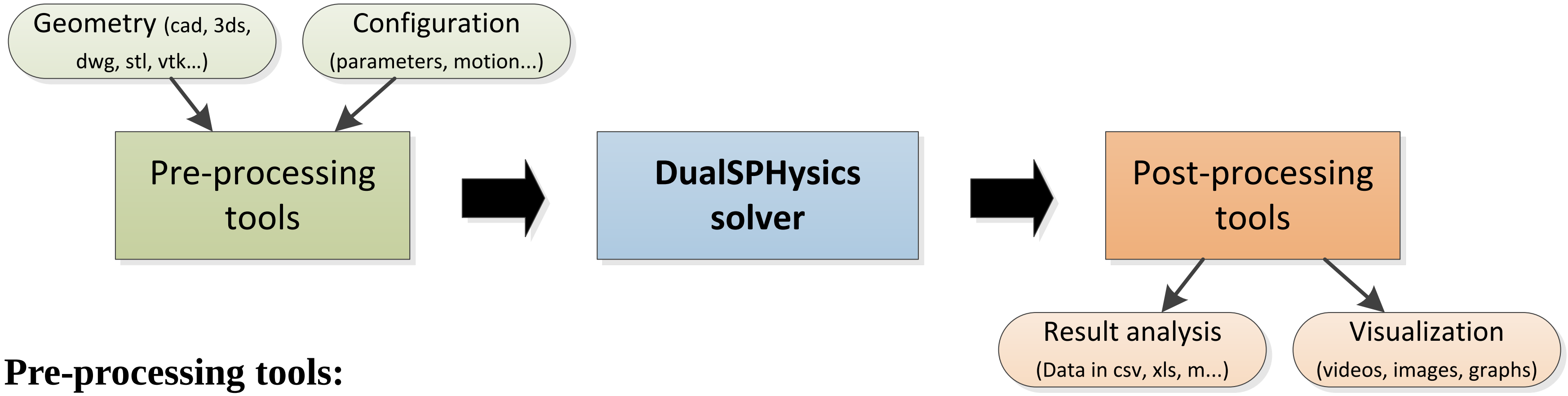
New Grand Challenges (GCs) are defined by the SPHERIC Steering Committee:
GC#5: Applicability to industry

December 2016:
2nd DualSPHysics Users Workshop in Manchester

Presentation about:
New Graphical User Interface for DualSPHysics

DesignSPHysics: Graphical User Interface for DualSPHysics

DSPH project includes:



Geometry (cad, 3ds, dwg, stl, vtk...)
Configuration (parameters, motion...)
Pre-processing tools
DualSPHysics solver
Post-processing tools
Result analysis (Data in csv, xls, m...)
Visualization (videos, images, graphs)

Pre-processing tools:

- Converts geometry into particles.
- Provides configuration for simulation.

DualSPHysics solver:

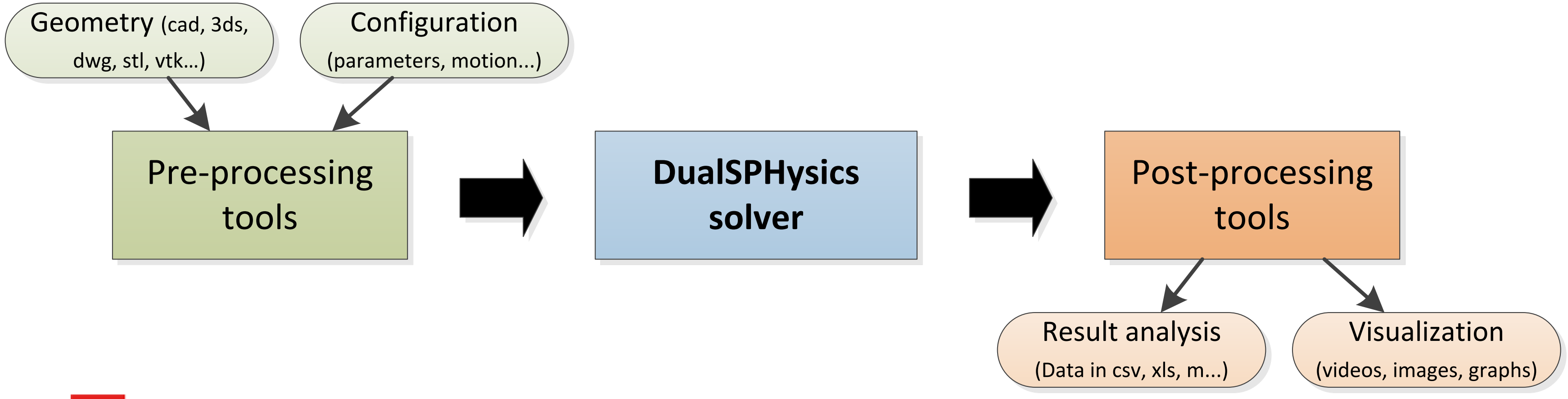
- Runs simulation using SPH particles.
- Obtains data simulation for time intervals.

Post-processing tools:

- Calculates magnitudes using particle data.
- Visualisation starting form SPH particles.

DesignSPHysics: Graphical User Interface for DualSPHysics

DSPH project includes:



Geometry (cad, 3ds, dwg, stl, vtk...)

Configuration (parameters, motion...)

Pre-processing tools

DualSPHysics solver

Post-processing tools

Result analysis (Data in csv, xls, m...)

Visualization (videos, images, graphs)

FreeCAD

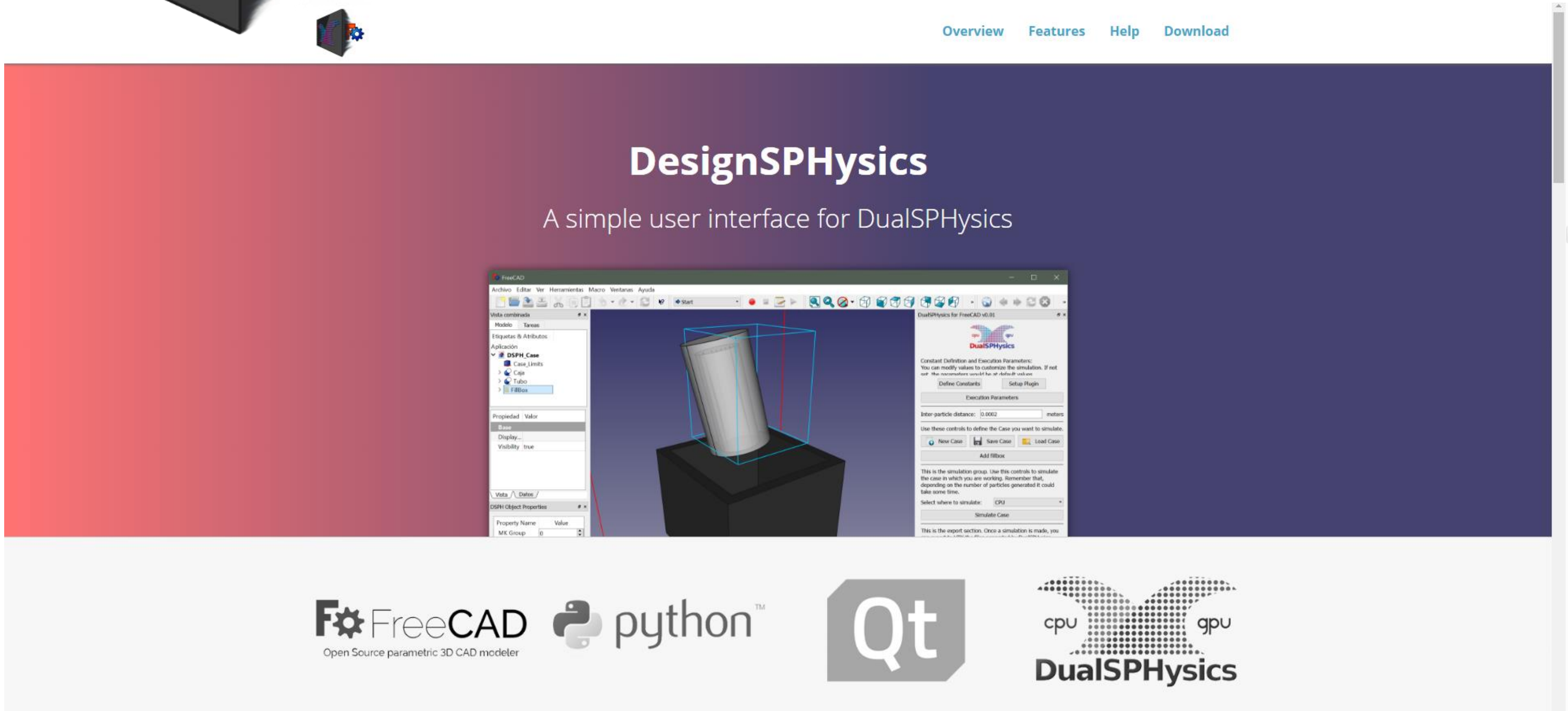
Open Source parametric 3D CAD modeler

python™

DesignSPHysics

DesignSPHysics: Graphical User Interface for DualSPHysics

Main developer: Andrés Vieira



DesignSPHysics

A simple user interface for DualSPHysics

Constant Definition and Execution Parameters:
You can modify values to customize the simulation. If not set, the parameters would be at default values

Inter-particle distance: 0.0002 meters

Use these controls to define the Case you want to simulate.

This is the simulation group. Use this controls to simulate the case in which you are working. Remember that, depending on the number of particles generated it could take some time.

Select where to simulate: CPU

This is the export section. Once a simulation is made, you

http://design.sphysics.org

DesignSPHysics: Graphical User Interface for DualSPHysics

FreeCAD:

open-source general-purpose parametric 3D CAD modeler
developed mostly in C++ and uses Python as a scripting language for tools, plug-ins and modules

Python:

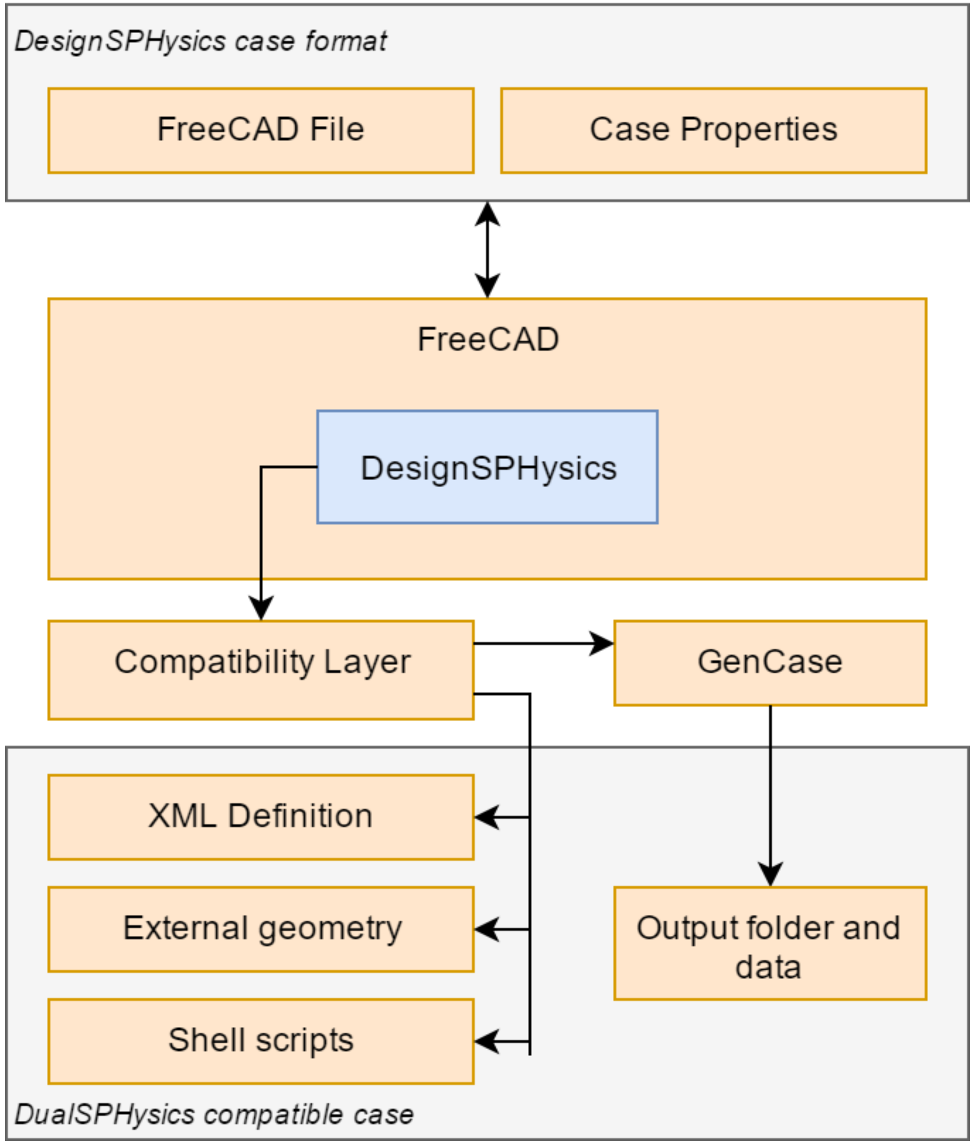
high-level programming and scripting language for general-purpose uses

Qt framework (PySide):

cross-platform application framework used for developing applications that can be executed in different software and hardware platforms with minor changes in the underlying codebase

PySide is an open source library for Python that provides bindings for the QT framework.

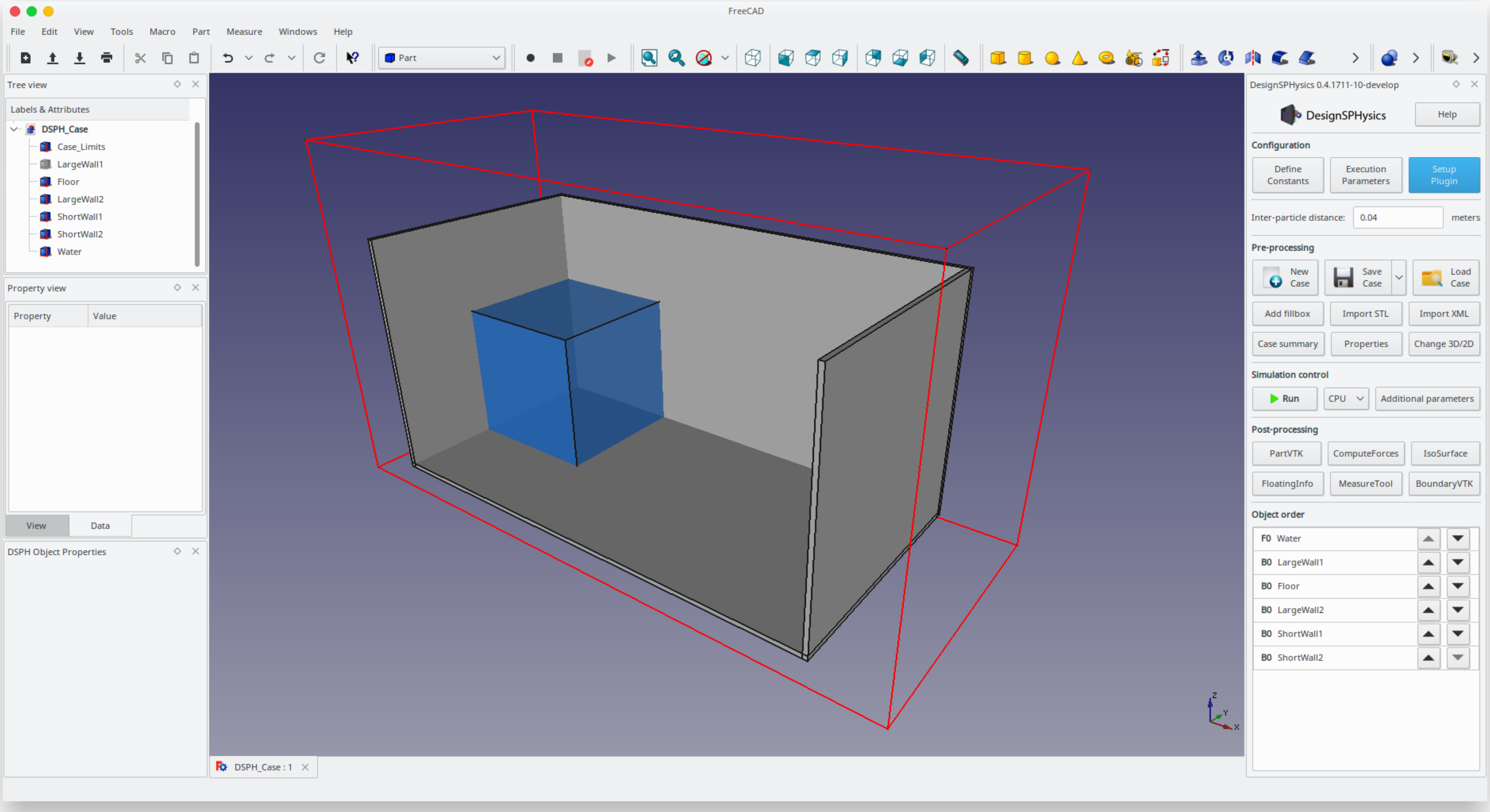
DesignSPHysics: Graphical User Interface for DualSPHysics



Data workflow

DesignSPHysics: Graphical User Interface for DualSPHysics

DesignSPHysics is presented as a plug-in (or macro) for FreeCAD



Inter-particle distance: 0.04 meters

CPU

F0 Water
B0 LargeWall1
B0 Floor
B0 LargeWall2
B0 ShortWall1
B0 ShortWall2

DesignSPHysics: Graphical User Interface for DualSPHysics

MAIN FEATURES

- **Automatic installer**

- **External executables support**

- **Case creation**
  - Managed as a FreeCAD document.
  - Already includes predefined objects (cubes, spheres, cylinders, cones…)

- **Case loading and saving**
  - By using a linking structure and a 3-D FreeCAD standard document

- **Geometry creation and adaptation**
  - It helps to create the XML input file of DualSPHysics
  - Complex objects are exported as an STL and will be loaded as an STL file

- **STL importing**
  - DesignSPHysics includes a STL importer wrapper to rescale

DesignSPHysics: Graphical User Interface for DualSPHysics

MAIN FEATURES

- **Constant and parameter settings**
  - The execution parameters include the simulation time, the kernel function, the time step algorithm, etc.
  - The constants to be defined are gravity, reference density of the fluid, CFL number, the smoothing length, etc.

- **XML importing**
  - An existing DualSPHysics XML case file can be interpreted by DesignSPHysics and imported into FreeCAD

- **2D and 3D support**
  - DualSPHysics supports a 2-D simulations ignoring the Y-axis interactions and reducing the resources needed to execute the simulation

DesignSPHysics: Graphical User Interface for DualSPHysics

MAIN FEATURES

- **Object properties edition**
  - Object type: if the object will be converted into fluid particles or boundary
  - Object MK: The MK number acts as a label or identifier
  - Fill mode: creation of particles as wireframe, face, solid or full

- **Initial fluid velocity**
  - This will be soon replaced by inlet / outlet open boundaries

- **Floating configuration**
  - Define mass or density of the body, the centre of gravity, an initial inertia and an initial linear and/or angular velocity vector

- **Moving objects**
  - Different types of motion (rectilinear, circular, sinusoidal and rotational)

- **Wave generation**: monochromatic (regular) and random (irregular) waves
  - Define wave height, wave period, depth…

DesignSPHysics: Graphical User Interface for DualSPHysics

MAIN FEATURES

- Integrated simulation execution

- Integrated postprocessing

DesignSPHysics is a complete software that allows the user to

1) create a new case,
2) execute the simulation and then
3) analyse the results
    3.1) by visualising the particles
    3.2) by computing physical magnitudes of interest

DesignSPHysics: Graphical User Interface for DualSPHysics

New Features

- Automatic opening with Paraview →

- Save without GenCase

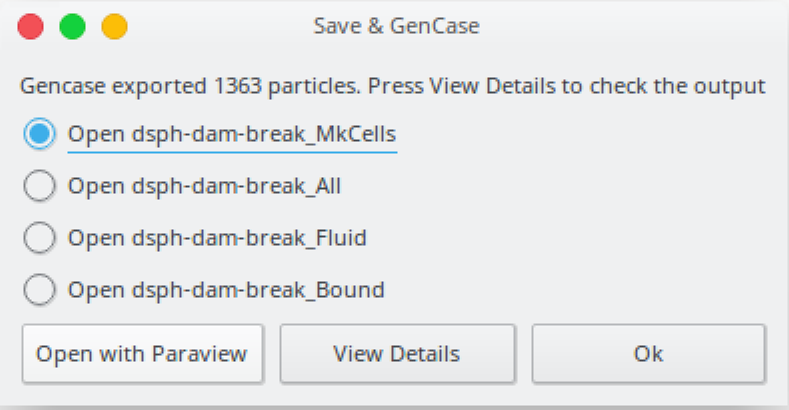
- Details in post-processing tools

- Per version configuration

- Default configuration with the latest dualsphysics

- Reworked some UI elements to fit most screens

- Improved stability and fixed a lot of errors, typos, wrong units, etc



Gencase exported 1363 particles. Press View Details to check the output

● Open dsph-dam-break_MkCells
○ Open dsph-dam-break_All
○ Open dsph-dam-break_Fluid
○ Open dsph-dam-break_Bound

DesignSPHysics: Graphical User Interface for DualSPHysics

Future Improvements

- Toolbox oriented UI with retractable sections, to better organize the space

- External server support with SSH: Run cases in a remote machine

- Multi-case support: Multiple cases in different tabs of FreeCAD

DesignSPHysics: Graphical User Interface for DualSPHysics

CONCLUSIONS

DesignSPHysics is a solution developed in Python as a FreeCAD plugin.

FreeCAD is an existing tool to create complex geometries.

DesignSPHysics uses already implemented SPH code, there is no need to change the source files of the SPH solver, only to adapt the interface to the code.

Users of DesignSPHysics do not have to edit a text file, only follow the GUI

Good practice is to use FreeCad and Python (open-source and multiplatform)

DesignSPHysics is a complete software that allows the user:

- to create a new case
- to execute the simulation and then
- to analyse the results by visualising the particles and by computing physical magnitudes of interest

Thanks for your attention

http://design.sphysics.org

Previous releases: github (send issue)
Mail to dualsphysics@gmail.com
Mail to anvieiravazquez@gmail.com

Graphical User Interface for SPH codes: DesignSPHysics

A. Vieira, O. García-Feal, J.M. Domínguez,
A.J.C. Crespo, M. Gómez-Gesteira
EPHYSLAB, Environmental Physics Laboratory
Universidade de Vigo