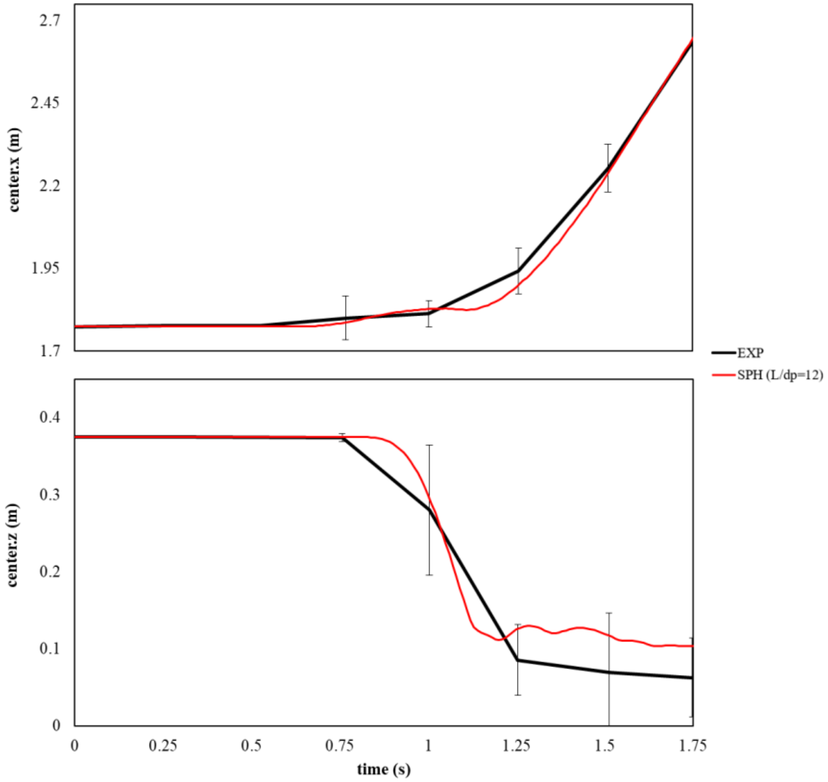
Updates about coupling with Project Chrono

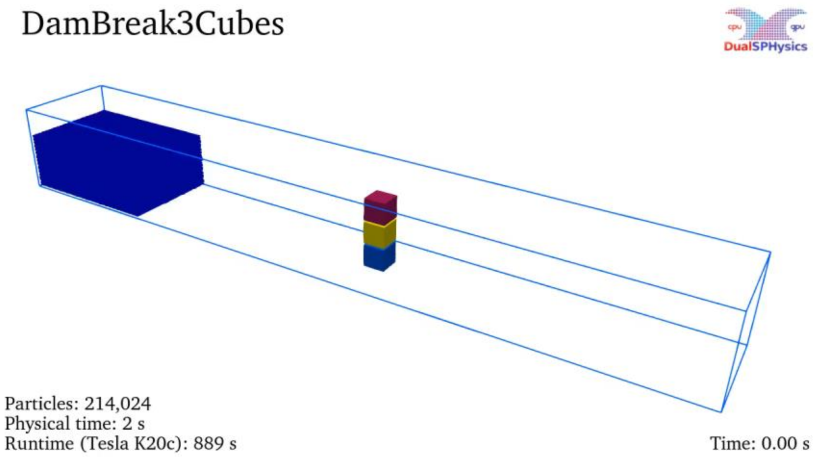
IVÁN MARTÍNEZ-ESTÉVEZ

Aim & Motivation

Coupling between models

Canelas et al., 2018





- Upgrade the version of Chrono
- Coupling with DualSPHysics (v5.2)
- Features presented in Canelas et al., 2018
- New features were implemented
- New coupling strategy

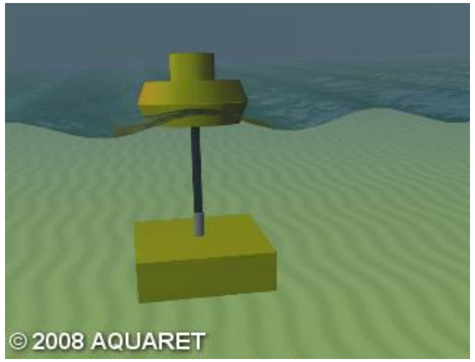
Aim & Motivation

UniversidadeVigo

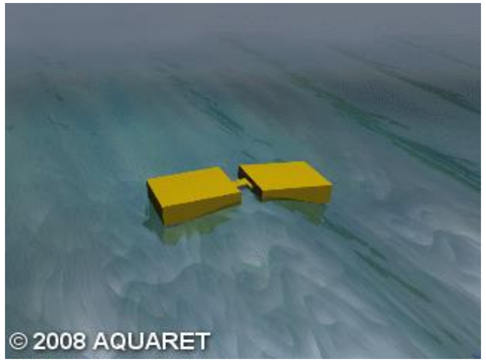
Environmental Physics Laboratory
EPhysLab

Numerical modelling to study the efficiency and survivability of:

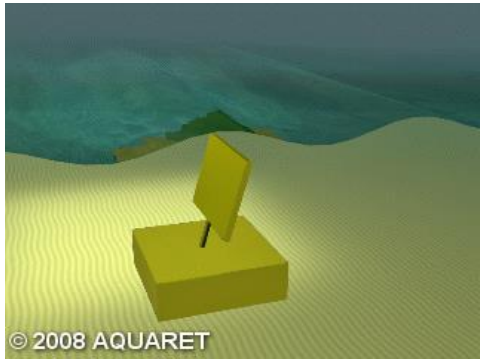
i) Wave Energy Converters (WECs)



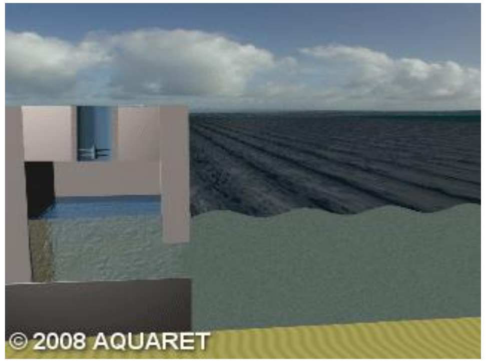
Point absorber



Attenuator



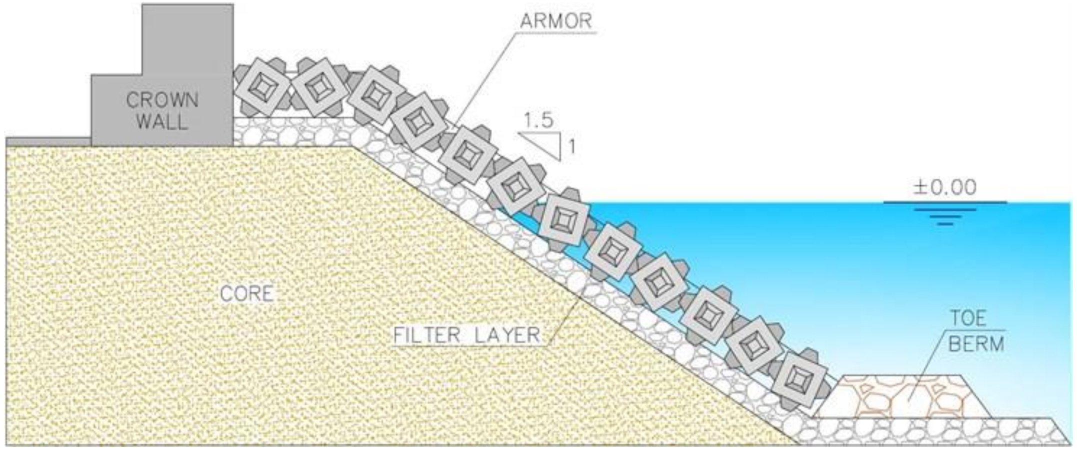
Oscillating wave surge converter (OWSC)



Oscillating water column (OWC)

ii) Coastal protection

https://www.cubipod.com/



Outline

Project Chrono

Coupling SPH-Chrono

Collision detection

Multibody dynamics

Validation

Applications

Conclusions & Future Work

Outline

Project Chrono

Coupling SPH-Chrono

Collision detection

Multibody dynamics

Validation

Applications

Conclusions & Future Work

PROJECT CHRONO

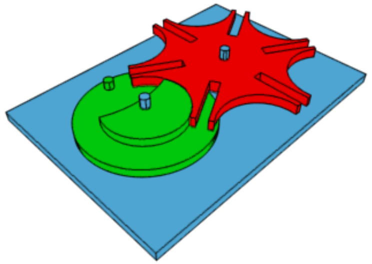
https://projectchrono.org/

Project Chrono

Open-source **multi-physics** simulation engine

Tasora et al., 2016. **Chrono: An Open Source Multi-physics Dynamics Engine**. High Performance Computing in Science and Engineering. https://doi.org/10.1007/978-3-319-40361-8_2

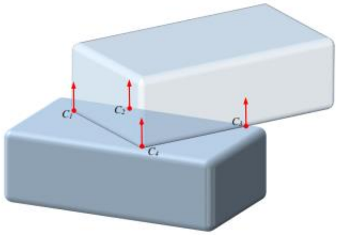
BSD



https://projectchrono.org/

Collision detection

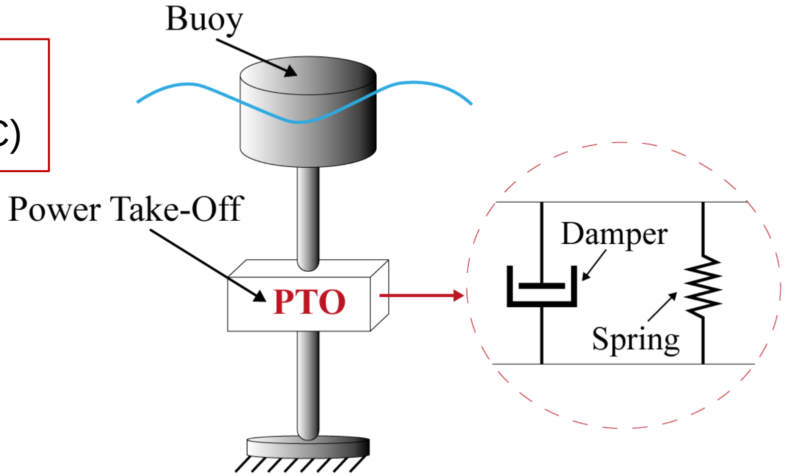
Compute frictional contact forces
Collision detection algorithms.
Define material properties.



Multibody dynamics

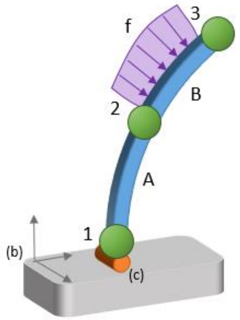
Mechanical constraints between rigid and flexible objects.
Add motors, linear actuators, springs and dampers.
Apply forces and torques.

Power Take-Off (PTO) of
Wave Energy Converters (WEC)



Finite elements

Use the FEA module to create finite elements and model flexible parts.
Beams, cables, shells, solid tetrahedrons and hexahedrons.



Outline

DualSPHysics ⇄ PROJECT CHRONO

Project Chrono

Coupling SPH-Chrono

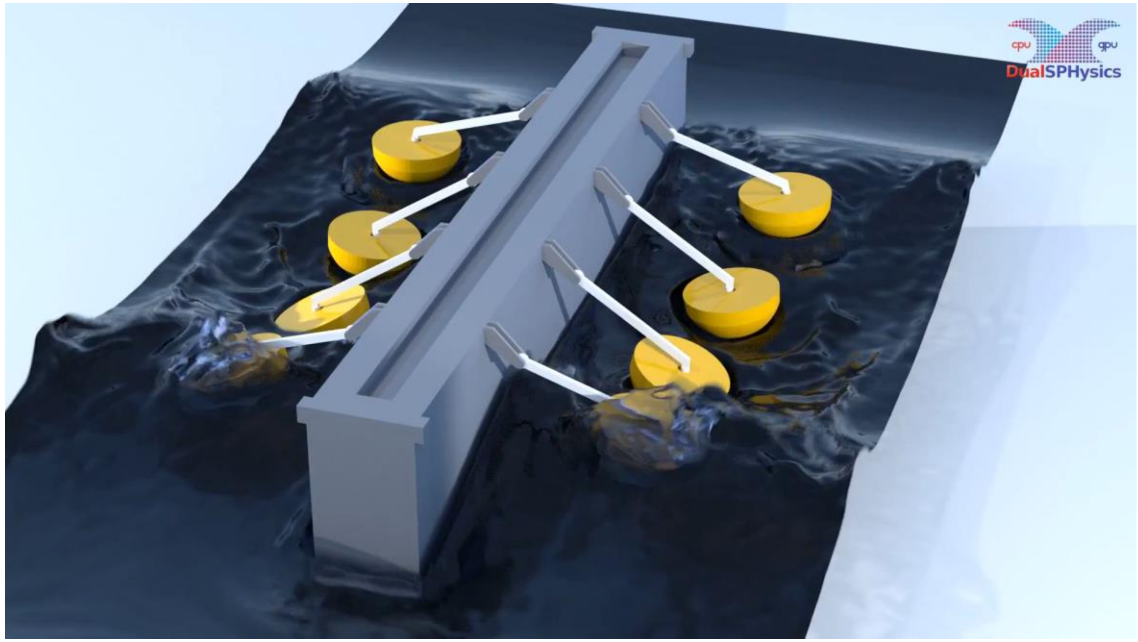
Collision detection

Multibody dynamics

Validation

Applications

Conclusions & Future Work



Coupling SPH-Chrono

Martínez-Estévez, I., Domínguez, J. M., Tagliafierro, B., Canelas, R., García-Feal, O., Crespo, A. J. C. & Gómez-Gesteira, M. (**2022**). **Coupling of an SPH-based solver with a multiphysics library**. *Computer Physics Communications. In Press*

Software components

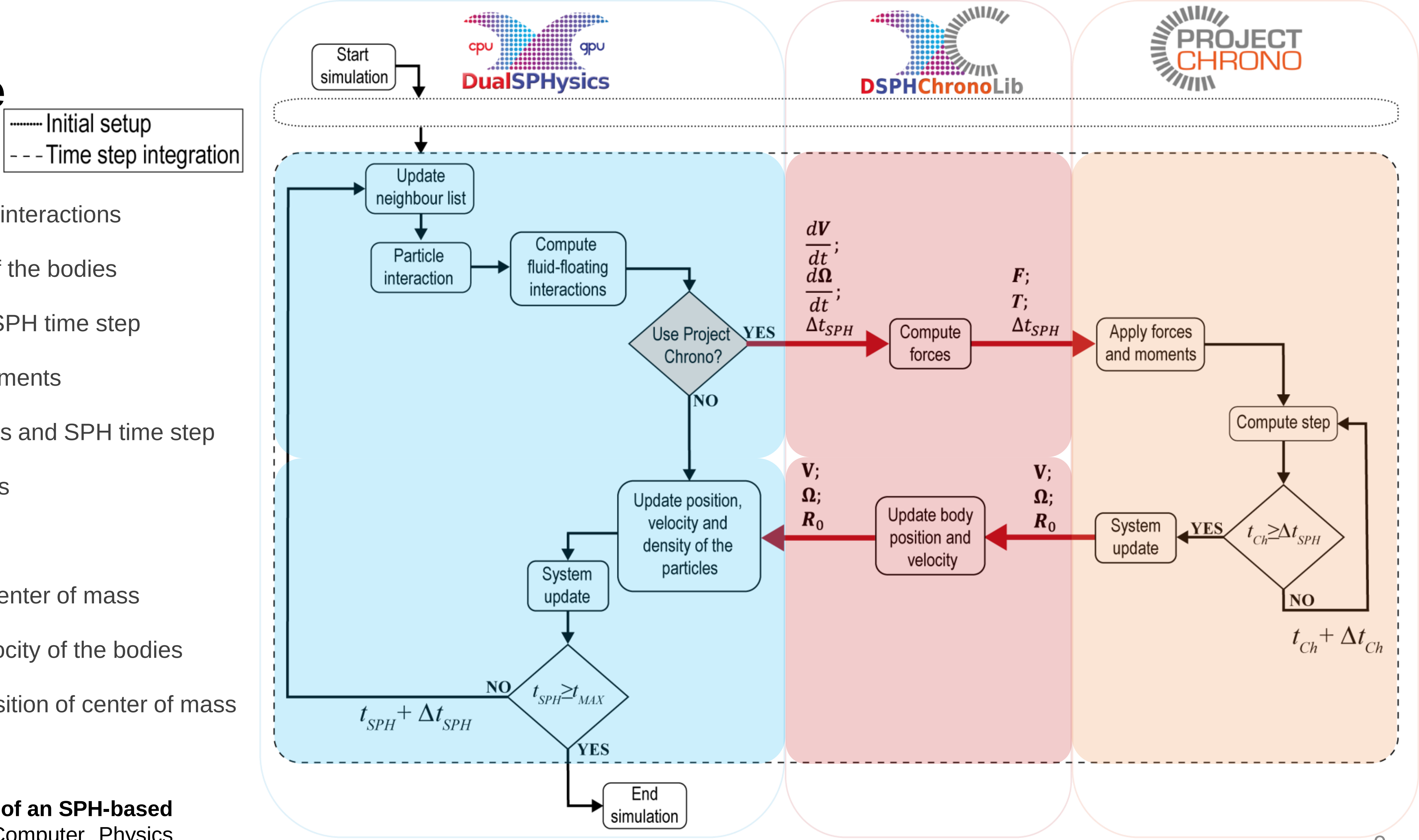
- Discretizes the domain into particles
- Solves the fluid
- Solves the fluid-solid interaction

- Dynamic library (.dll / .so)
- Controls the communication process
- Handles the information transfer
- <u>Abstraction</u>: only changes in DSPHChronoLib

- Dynamic library (.dll / .so)
- Solves the mechanical constraints
- Solves solid-solid interaction

Coupling SPH-Chrono

Coupling scheme



DualSPHysics computes the SPH particle interactions

DualSPHysics obtains the accelerations of the bodies

DualSPHysics exports accelerations and SPH time step

DSPHChronoLib computes forces and moments

DSPHChronoLib transfers forces, moments and SPH time step

Chrono applies forces and solve the bodies

Chrono solves the rigid objects

Chrono returns velocities and position of center of mass

DSPHChronoLib updates position and velocity of the bodies

DSPHChronoLib returns velocities and position of center of mass

DualSPHysics updates the SPH system

Martínez-Estévez et al., 2022. **Coupling of an SPH-based solver with a multiphysics library.** Computer Physics Communications. In Press

Coupling SPH-Chrono

JSphCpuSingle & JSphGpuSingle — Included in *src/source* of v5.2 BETA

Implementation

RunFloating() function to process floating objects

ChronoObjects pointer to manage the data using Chrono

XML to enable the use of Chrono

Export the linear and angular accelerations

Solve Chrono during a Δt_{SPH}

Read position of center of mass and velocities to update the system

JSphCpuSingle::**RunFloating()**

```
//==============================================================================
/// Process floating objects
/// Procesa floating objects.
//==============================================================================
void JSphCpuSingle::RunFloating(double dt,bool predictor){
 ...
    //-Run floating with Chrono library.
    if(ChronoObjects){
      TmcStop(Timers,TMC_SuFloating);
      TmcStart(Timers,TMC_SuChrono);
      //-Export data / Exporta datos.
      for(unsigned cf=0;cf<FtCount;cf++)if(FtObjs[cf].usechrono){
        ChronoObjects->SetFtData(
          FtObjs[cf].mkbound,
          FtoForces[cf].face,
          FtoForces[cf].fomegaace);
      }
      //-Applies the external velocities to each floating body of Chrono.
      if(FtLinearVel!=NULL)ChronoFtApplyImposedVel();
      //-Calculate data using Chrono / Calcula datos usando Chrono.
      ChronoObjects->RunChrono(Nstep,TimeStep,dt,predictor);
      //-Load calculated data by Chrono / Carga datos calculados por Chrono.
      for(unsigned cf=0;cf<FtCount;cf++)if(FtObjs[cf].usechrono)
        ChronoObjects->GetFtData(
          FtObjs[cf].mkbound,
          FtoForcesRes[cf].fcenterres,
          FtoForcesRes[cf].fvelres,
          FtoForcesRes[cf].fomegares);
      TmcStop(Timers,TMC_SuChrono);
      TmcStart(Timers,TMC_SuFloating);
    }
 ...
}
```

$$\frac{d\mathbf{V}}{dt};\frac{d\mathbf{\Omega}}{dt}$$

$$\Delta t_{SPH}$$

$$\mathbf{R}_0;\mathbf{V};\mathbf{\Omega};$$

Case_Def.xml

```
<case>
  ...
  <execution>
    ...
    <special>
      ...
      <chrono>
        ...
      </chrono>
      ...
    </special>
    ...
  </execution>
</case>
```

Coupling SPH-Chrono

XML file: *case-execution-special-chrono*

```
<case>
    ...
    <execution>
        <special>
            <chrono>
                <savedata value="0.01" />
                <schemescale value="1"/>
                <scaleforce>
                    <body mkbound="1" x="0.01" y="1" z="1" />
                </scaleforce>
                <collision activate="true"/>
                <bodyfloating id="box1" mkbound="1" modelfile="box.obj" />
                <bodyfixed id="domain" mkbound="0" modelfile="domain.obj" />
                <link_linearspring idbody1="domain" idbody2="box1" />
                <link_coulombdamping  idbody1="domain" idbody2="box1" />
                <link_hinge idbody1="box1" idbody2="box2" />
                <link_spheric idbody1="box2" idbody2="box3" />
                <link_pointline idbody1="box4" />
                <link_pointline idbody1="box4" idbody2="box5"/>
                <link_spheric idbody1="box5"/>
            </chrono>
        </special>
        ...
    </execution>
</case>
```

savedata: Saves CSV with data exchange
ChronoBody_forces.csv,
ChronoExchange_mkbound_XX.csv

schemescale: Creates VTK file with the initial scheme of Chrono objects using the given scale
CfgChrono_Scheme.vtk

scaleforce: Defines a force scaling factor

collision: Section to activate collisions

bodyfloating & **bodyfixed**: Indicates the rigid objets to be solved by Chrono

link_xxxx: Section to define mechanical constraints.

Outline

Project Chrono

Coupling SPH-Chrono

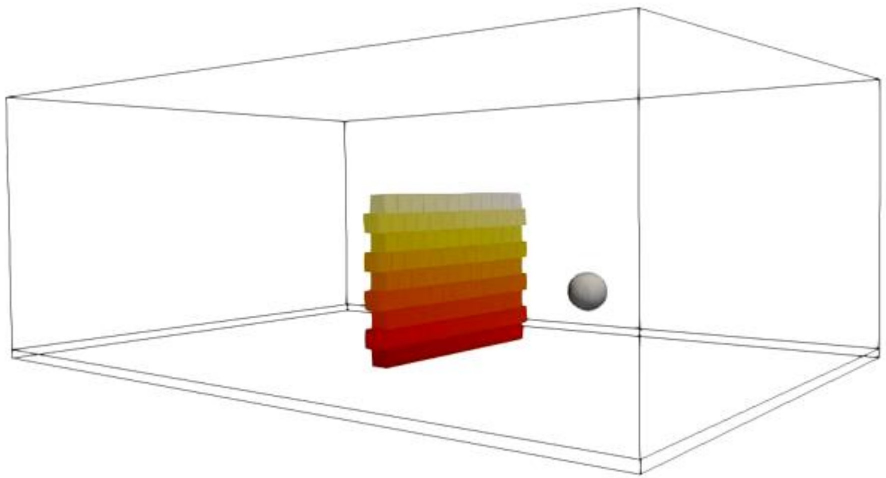
Collision detection

Multibody dynamics

Validation

Applications

Conclusions & Future Work



https://dual.sphysics.org/

Collision detection

Methods

Two approaches based on Discret Element Method (DEM)

Non-Smooth Contacts (NSC)

DEM using complementarity (DEM-C)

Enforce non-penetration of the discrete elements

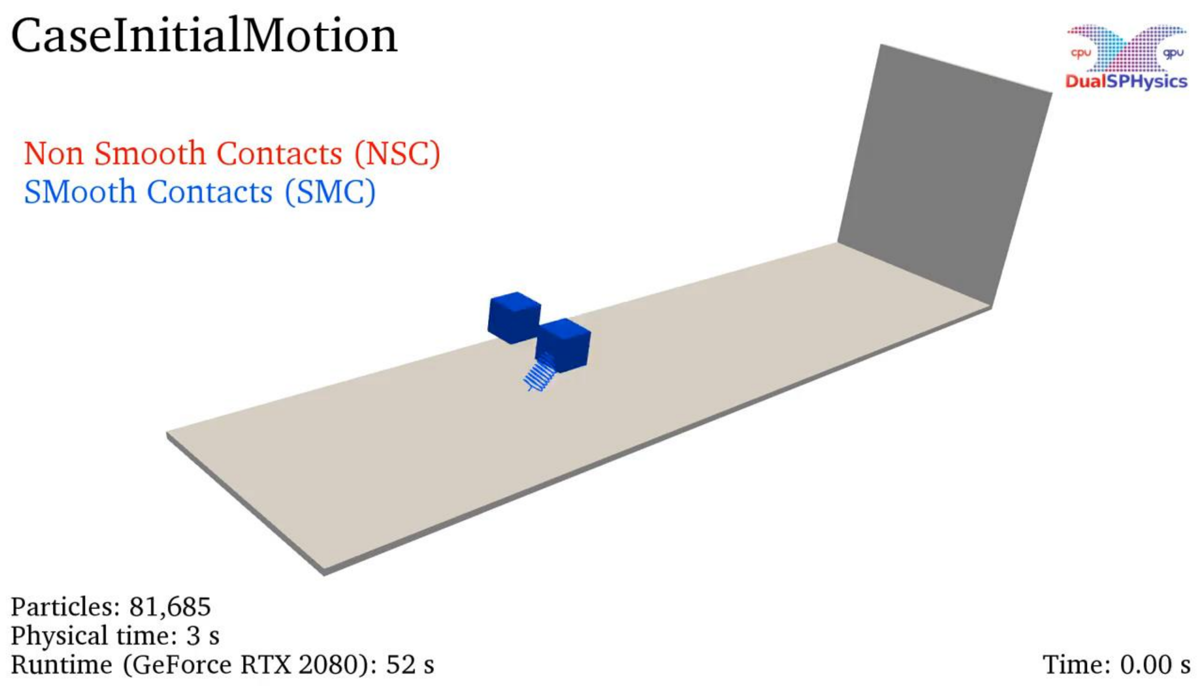
Friction, Restitution coefficients can be defined

SMooth Contacts (SMC)

DEM with penalties (DEM-P) or soft-body approach

Penetration to experience deformation

Young’s Modulus, Poisson’s Ratio can be defined (in addition)



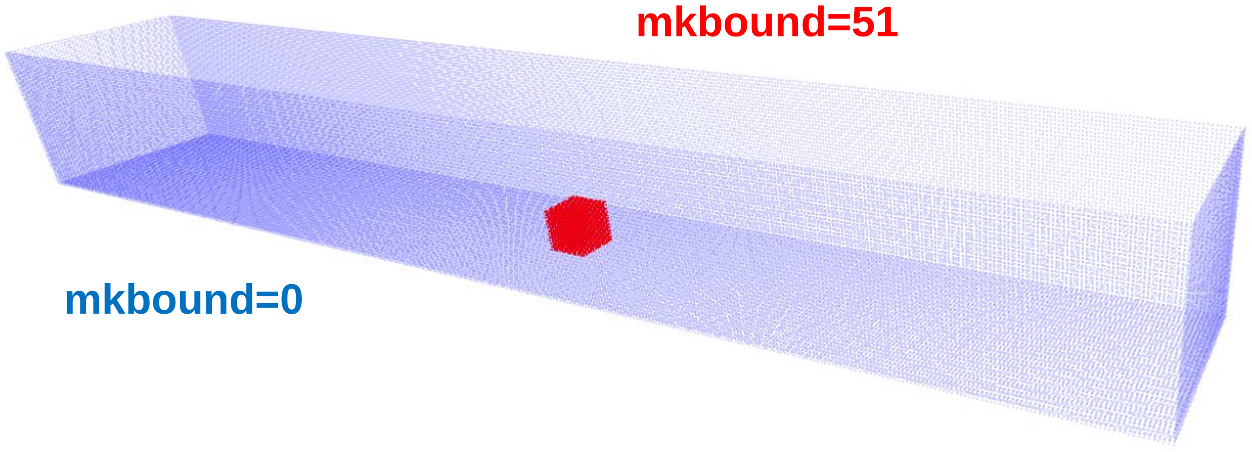
Collision detection

XML file

***Case_Def.xml:** case-casedef-geometry-commands-mainlist*

```
<case>
    <casedef>
        ...
        <geometry>
            <commands>
                <mainlist>
                    ...
                    <setmkbound mk="51" />
                    <drawbox>
                        <boxfill>solid</boxfill>
                        <point x="1.7" y="0.275" z="0.0125" />
                        <size x="0.15" y="0.15" z="0.15" />
                    </drawbox>
                    <shapeout file="cube1" />

                    <setmkbound mk="0" />
                    <drawbox>
                        <boxfill>all^top</boxfill>
                        <point x="-1.0" y="0" z="0" />
                        <size x="5" y="0.70" z="0.60" />
                    </drawbox>
                    <shapeout file="tank" />
                    ...
                </mainlist>
            </commands>
        ...
    </casedef>
    ...
</case>
```



Case_cube1_Actual.vtk
Case_tank_Actual.vtk

Collision detection

XML file

Material properties	NSC	SMC
Restitution	YES	YES
Kinetic friction	YES	YES
Young's modulus	NO	YES
Poisson ratio	NO	YES

***Case_Def.xml:** case-casedef-properties*

```xml
<case>
    <casedef>
        ...
        <properties>
            <propertyfile file="Floating_Materials.xml" path="materials" />
            <links>
                <link mkbound="0" property="steel" comment="Property for the tank" />
            </links>
        </properties>
        ...
    </casedef>
    ...
</case>
```

tank

***Case_Def.xml:** case-casedef-floatings*

```xml
<case>
    <casedef>
        ...
        <floatings>
            <floating mkbound="51-53" property="pvc">
                <massbody value="2.7" />
            </floating>
        </floatings>
        ...
    </casedef>
    ...
</case>
```

cube

Floating_Materials.xml

```xml
<materials>
    ...
    <property name="aluminium">
        <Young_Modulus value="69000000000.0" comment="Young Modulus (N/m2)" />
        <PoissonRatio value="0.33" comment="Poisson Ratio (-)" />
        <Restitution_Coefficient value="0.70" comment="Restitution Coefficient (-)" />
        <Kfric value="0.65" comment="Kinetic friction coefficient" />
    </property>
    <property name="steel">
        <Young_Modulus value="210000000000.0" comment="Young Modulus (N/m2)" />
        <PoissonRatio value="0.35" comment="Poisson Ratio (-)" />
        <Restitution_Coefficient value="0.80" comment="Restitution Coefficient (-)" />
        <Kfric value="0.35" comment="Kinetic friction coefficient" />
    </property>
    <property name="pvc">
        <Young_Modulus value="3000000000.0" comment="Young Modulus (N/m2)" />
        <PoissonRatio value="0.30" comment="Poisson Ratio (-)" />
        <Restitution_Coefficient value="0.60" comment="Restitution Coefficient (-)" />
        <Kfric value="0.15" comment="Kinetic friction coefficient" />
    </property>
    ...
</materials>
```

Collision detection

XML file

Case_Def.xml: *case-execution-special-chrono*

```
<case>
    ...
    <execution>
        ...
        <special>
            ...
            <chrono>
                <savedata value="0.01" comment="Saves CSV with data exchange for each time interval (0=all steps)" />
                <schemescale value="1" comment="Scale used to create the initial scheme of Chrono objects (default=1)" />
                <collision activate="true">
                    <distancedp value="0.5" comment="Allowed collision overlap according Dp (default=0.5)" />
                    <contactmethod value="0"
                        comment="Contact method type. 0:NSC (Non Smooth Contacts), 1:SMC (SMooth Contacts). (default=0)" />
                </collision>
                <bodyfloating id="cube1" mkbound="51" modelfile="AutoActual" />
                <bodyfixed id="tank" mkbound="0" modelfile="AutoActual" modelnormal="invert" />
            </chrono>
            ...
        </special>
        ...
    </execution>
</case>
```

Enables the collision detection

contactmethod: Defines the contact method type (NSC or SMC)

distancedp: Maximum allowed overlap $f(dp)$

Case_Def.xml: *case-execution-parameters*

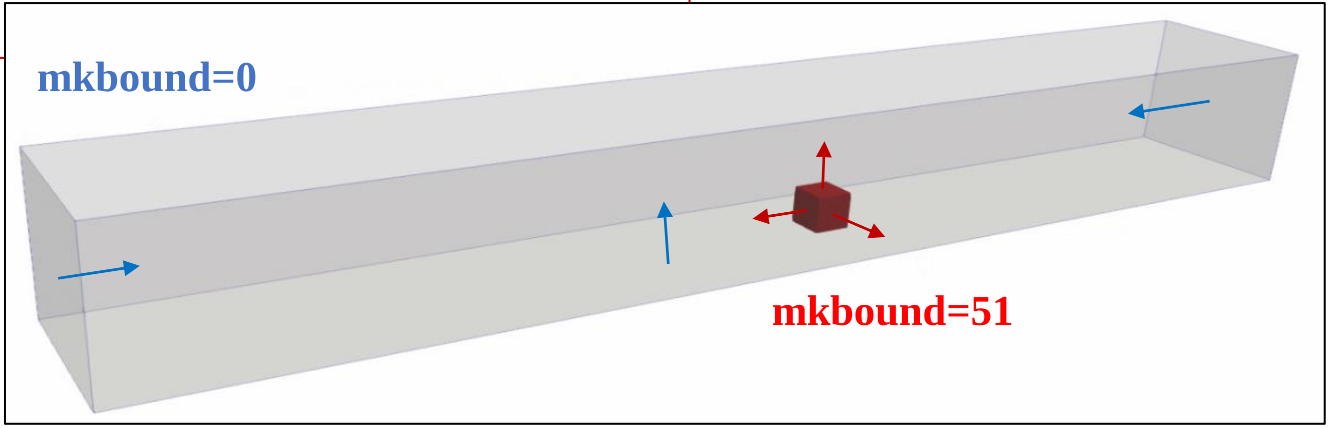
```
<case>
    ...
    <execution>
        ...
        <parameters>
            ...
            <parameter key="RigidAlgorithm" value="3"
                comment="Rigid Algorithm 0:collision-free,
                    1:SPH, 2:DEM, 3:Chrono (default=1)" />
            ...
        </parameters>
    </execution>
</case>
```

Creates the objects

cube1_mkb0051.obj
tank_mkb0000.obj

modelfile: Indicates the source file to generate the shape
AutoActual:
Case_cube1_Actual.vtk
Case_tank_Actual.vtk

modelnormal: Defines de direction of the normals



Collision detection

Initialisation

DualSPHysics

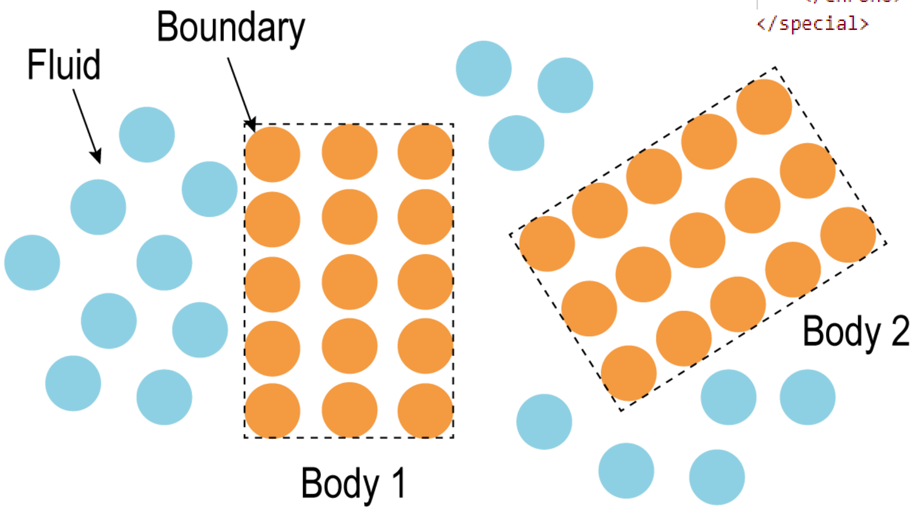
Shape (.obj file)

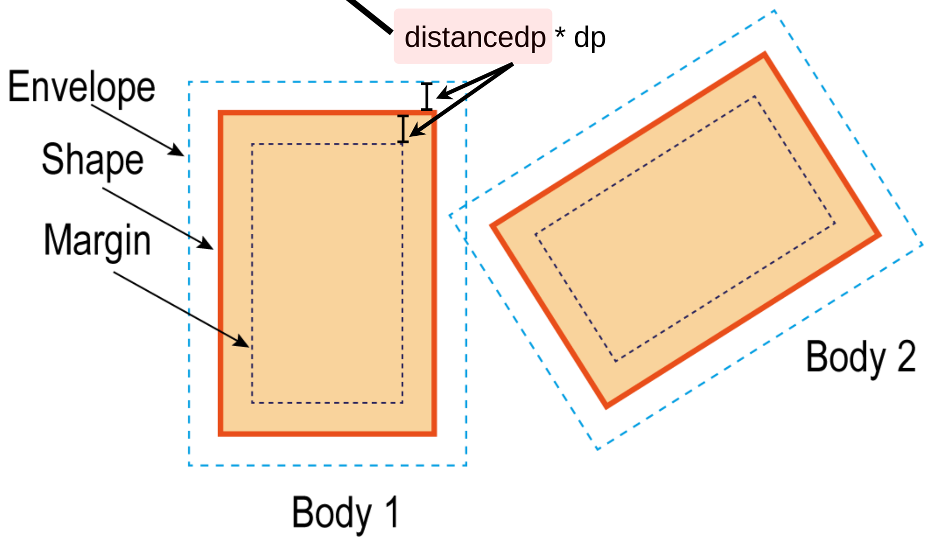
body1_mkbXXXX.obj
body2_mkbXXXX.obj

DSPHChronoLib

PROJECT CHRONO

```
<special>
    <chrono>
        <collision activate="true">
            <distancedp value="0.1"
                comment="Allowed collision overlap according Dp (default=0.5)" />
        </collision>
        ...
    </chrono>
</special>
```





Outline

Project Chrono

Coupling SPH-Chrono

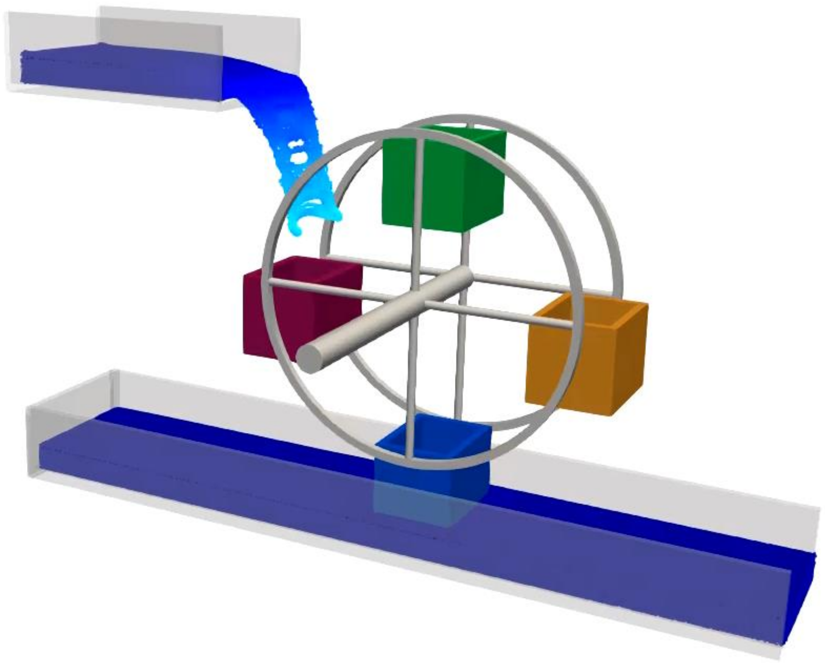
Collision detection

Multibody dynamics

Validation

Applications

Conclusions & Future Work



https://dual.sphysics.org/

Multibody dynamics

Mechanical constraints

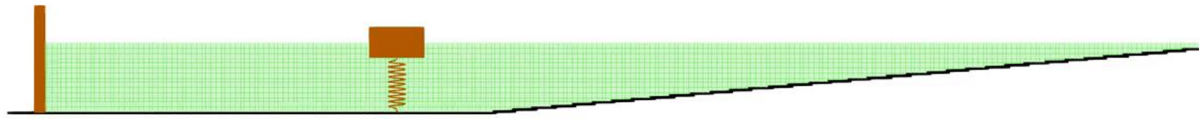
Translational Spring-Damper-Actuator (TSDA)

Spring with stiffness and damping

Spring using Coulomb damping

link_linearspring

CasePointAbsorberSpring



Runtime (GeForce RTX 2080): 497 s

```
<chrono>
    ...
    <bodyfixed id="Bottom" mkbound="0" />
    <bodyfloating id="Floater" mkbound="50" />
    <link_linearspring idbody1="Bottom" idbody2="Floater">
        <point_fb1 x="2.0" y="0.0" z="0.0" comment="Point in body 1" />
        <point_fb2 x="2.0" y="0.0" z="0.325" comment="Point in body 2" />
        <stiffness value="100.0" comment="Stiffness [N/m]" />
        <damping value="500.0" comment="Damping [Ns/m]" />
        <rest_length value="0.325" comment="Spring equilibrium length [m]" />
        <savevtk>
            ...
        </savevtk>
    </link_linearspring>
</chrono>
```

$$F_{PTO} = k(d - l) + c \cdot v$$

d: distance between the two origins
l: equilibrium length
v: velocity that the spring experiences

point_fb1: Point in body 1

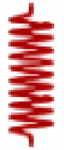
point_fb2: Point in body 2

k: Stiffness [N/m]

c: Damping [Ns/m]

rest_length: Spring equilibrium length [m]

savevtk: VTK for visualisation
SpringVtk/Chrono_Springs_xxxx.vtk



Multibody dynamics

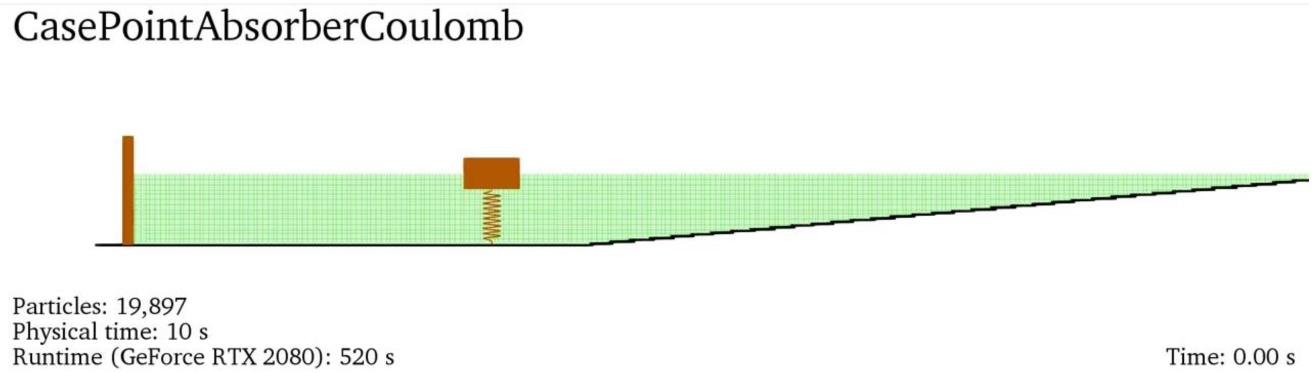
Mechanical constraints

link_coulombdamping

Translational Spring-Damper-Actuator (TSDA)

Spring with stiffness and damping

Spring using Coulomb damping



$$F_{PTO} = -sing(v)F_b$$

v : velocity that the spring experiences

```
<chrono>
    ...
    <bodyfixed id="Bottom" mkbound="0" />
    <bodyfloating id="Floater" mkbound="50" />
    <link_coulombdamping  idbody1="Bottom" idbody2="Floater">
        <point_fb1 x="2.0" y="0.0" z="0.0" comment="Point in body 1" />
        <point_fb2 x="2.0" y="0.0" z="0.325" comment="Point in body 2" />
        <rest_length value="0.325" comment="Spring equilibrium length [m]" />
        <damping value="10.0" comment="Coulomb force [N]" />
        <savevtk>
            ...
        </savevtk>
    </link_coulombdamping>
</chrono>
```

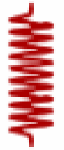
point_fb1: Point in body 1

point_fb2: Point in body 2

F_b: Coulomb force (damping) [N]

rest_length: Spring equilibrium length [m]

savevtk: VTK for visualisation
SpringVtk/Chrono_Springs_xxxx.vtk



Multibody dynamics

Mechanical constraints

Mechanical joints (Lock Formulation)

$$\boldsymbol{j} = \{\boldsymbol{q}, \boldsymbol{e}\}$$

Vector of translation in 3-D

$$\boldsymbol{q} = \{x, y, z\}$$

Quaternion for rotations

$$\boldsymbol{e} = \{e_0, e_1, e_2, e_3\}$$

0: unlocked (free movement)

1: locked (constrained movement)

$$\boldsymbol{F}_c = k_r\boldsymbol{\theta} + c_r\dot{\boldsymbol{\theta}}$$

$\boldsymbol{\theta}$: relative angle of rotation

$\dot{\boldsymbol{\theta}}$: angular velocity

k_r: rotational stiffness

c_r: rotational damping

Multibody dynamics

link_hinge

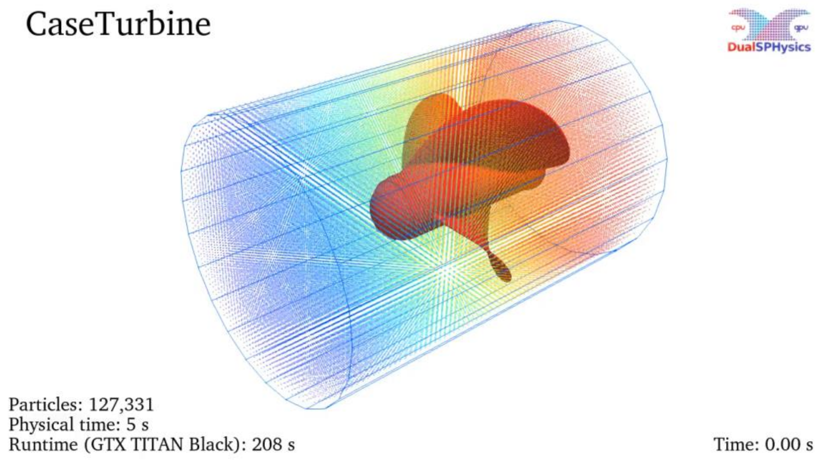
Mechanical constraints

Mechanical joints (Lock Formulation)

Hinge along an axis

Spherical hinge on a point

Sliding along an axis



$$F_c = k_r \theta + c_r \dot{\theta}$$

```
<chrono>
    ...
    <bodyfloating id="rotor" mkbound="1" />
    <link_hinge idbody1="rotor">
        <rotpoint x="0" y="0" z="0" comment="Point for rotation" />
        <rotvector x="1" y="0" z="0" comment="Vector direction for rotation" />
        <stiffness value="0" comment="Torsional stiffness [Nm/rad]" />
        <damping value="0" comment="Torsional damping [Nms/rad]" />
    </link_hinge>
</chrono>
```

rotpoint: Point for rotation

rotvector: Vector direction for rotation

k_r: Torsional stiffness [Nm/rad]

c_r: Torsional damping [Nms/rad]

Multibody dynamics

link_spheric

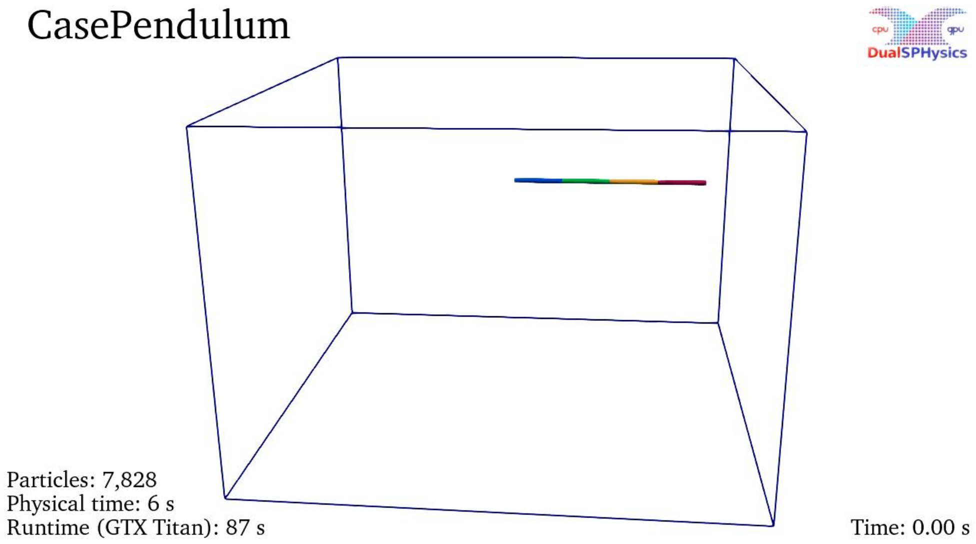
Mechanical constraints

Mechanical joints (Lock Formulation)

Hinge along an axis

Spherical hinge on a point

Sliding along an axis



$$\boldsymbol{F}_c = k_r\boldsymbol{\theta} + c_r\dot{\boldsymbol{\theta}}$$

```
<chrono>
    ...
    <link_spheric idbody1="arm0" idbody2="arm1">
        <rotpoint x="5" y="6" z="5" comment="Point for rotation" />
        <stiffness value="0" comment="Torsional stiffness [Nm/rad]" />
        <damping value="0" comment="Torsional damping [Nms/rad]" />
    </link_spheric>
    ...
</chrono>
```

rotpoint: Point for rotation

k_r: Torsional stiffness [Nm/rad]

c_r: Torsional damping [Nms/rad]

Multibody dynamics

Mechanical constraints

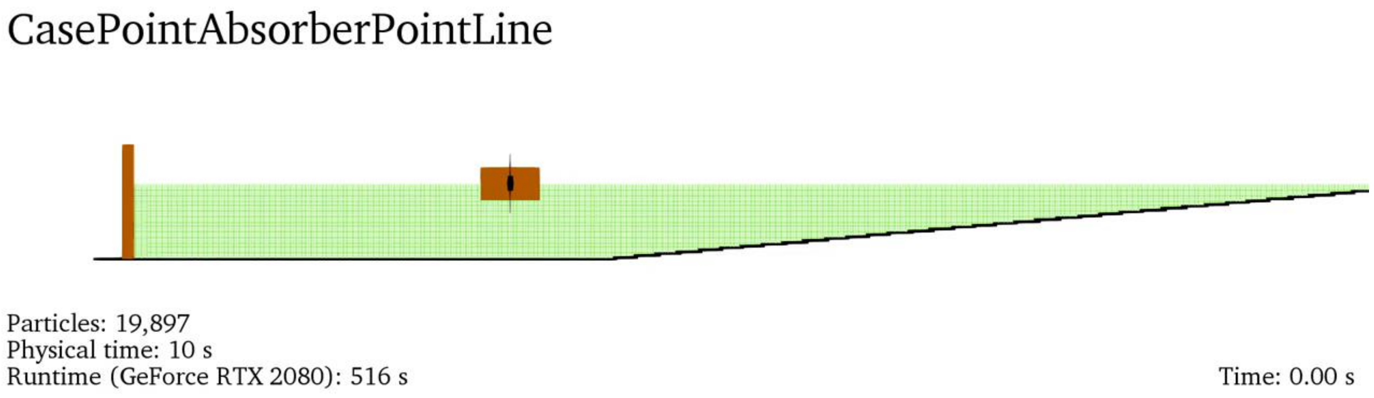
link_pointline

Mechanical joints (Lock Formulation)

Hinge along an axis

Spherical hinge on a point

Sliding along an axis



$$F_c = k_r\boldsymbol{\theta} + c_r\dot{\boldsymbol{\theta}}$$

```
<chrono>
    ...
    <link_pointline idbody1="Floater">
        <slidingvector x="0" y="0" z="1" comment="Sliding vector direction" />
        <rotpoint x="2" y="0" z="0.4" comment="Sliding point in the body" />
        <rotvector x="0" y="0" z="1" comment="Vector direction for rotation" />
        <stiffness value="0" comment="Torsional stiffness [Nm/rad]" />
        <damping value="0" comment="Torsional damping [Nms/rad]" />
    </link_pointline>
</chrono>
```

slidingvector: Sliding vector direction

rotpoint: Point for rotation

rotvector: Vector direction for rotation

k_r: Torsional stiffness [Nm/rad]

c_r: Torsional damping [Nms/rad]

Outline

Project Chrono

Coupling SPH-Chrono

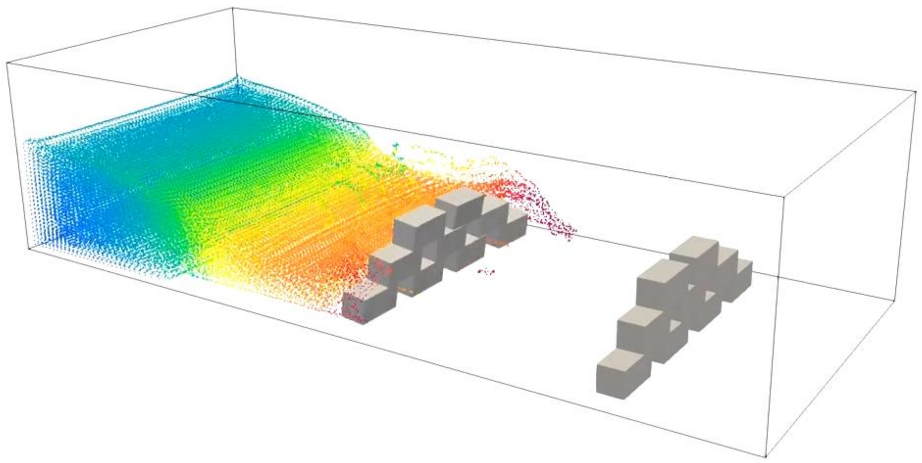
Collision detection

Multibody dynamics

Validation

Applications

Conclusions & Future Work



https://dual.sphysics.org/

Validation

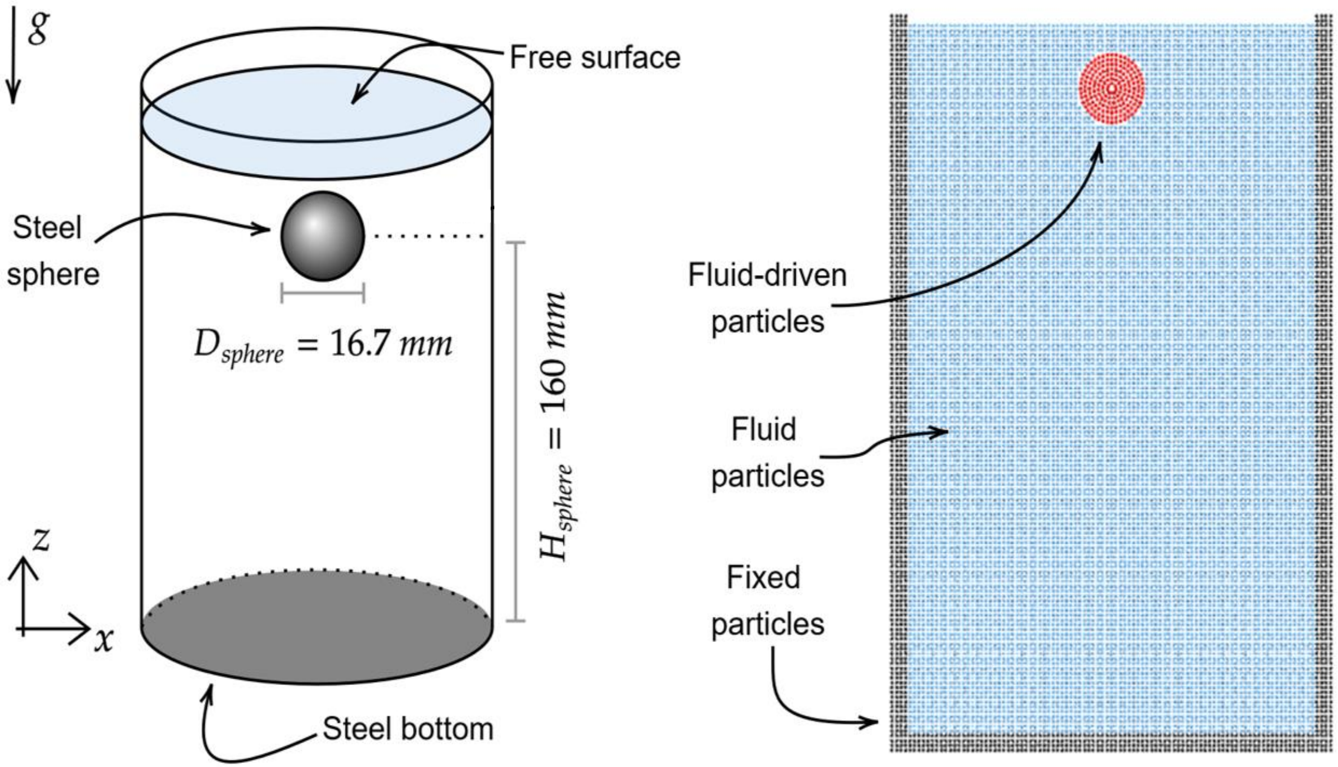
Collision detection

Exp. data in **Hagemeier et al., 2020**.

D=16.7 mm

Using SMC

Steel sphere and bottom



Martínez-Estévez et al., 2022. **Coupling of an SPH-based solver with a multiphysics library.** Computer Physics Communications. *In Press*

Validation

Collision detection

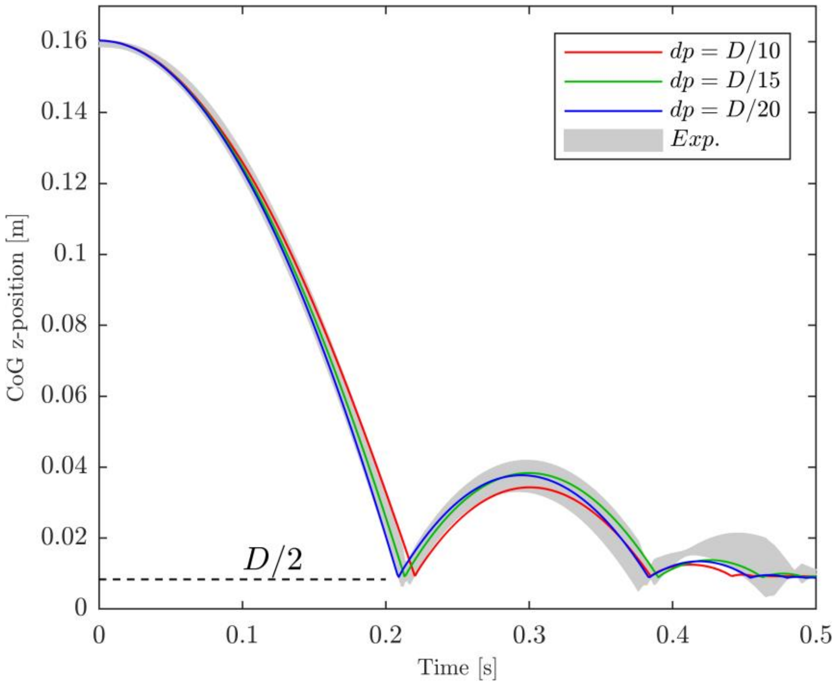
Exp. data in **Hagemeier et al., 2020**.

D=16.7 mm

Using SMC

Steel sphere and bottom

Comparison between numerical and experimental (grey) of the center of gravity (CoG) of the falling ball



Martínez-Estévez et al., 2022. **Coupling of an SPH-based solver with a multiphysics library.** Computer Physics Communications. *In Press*

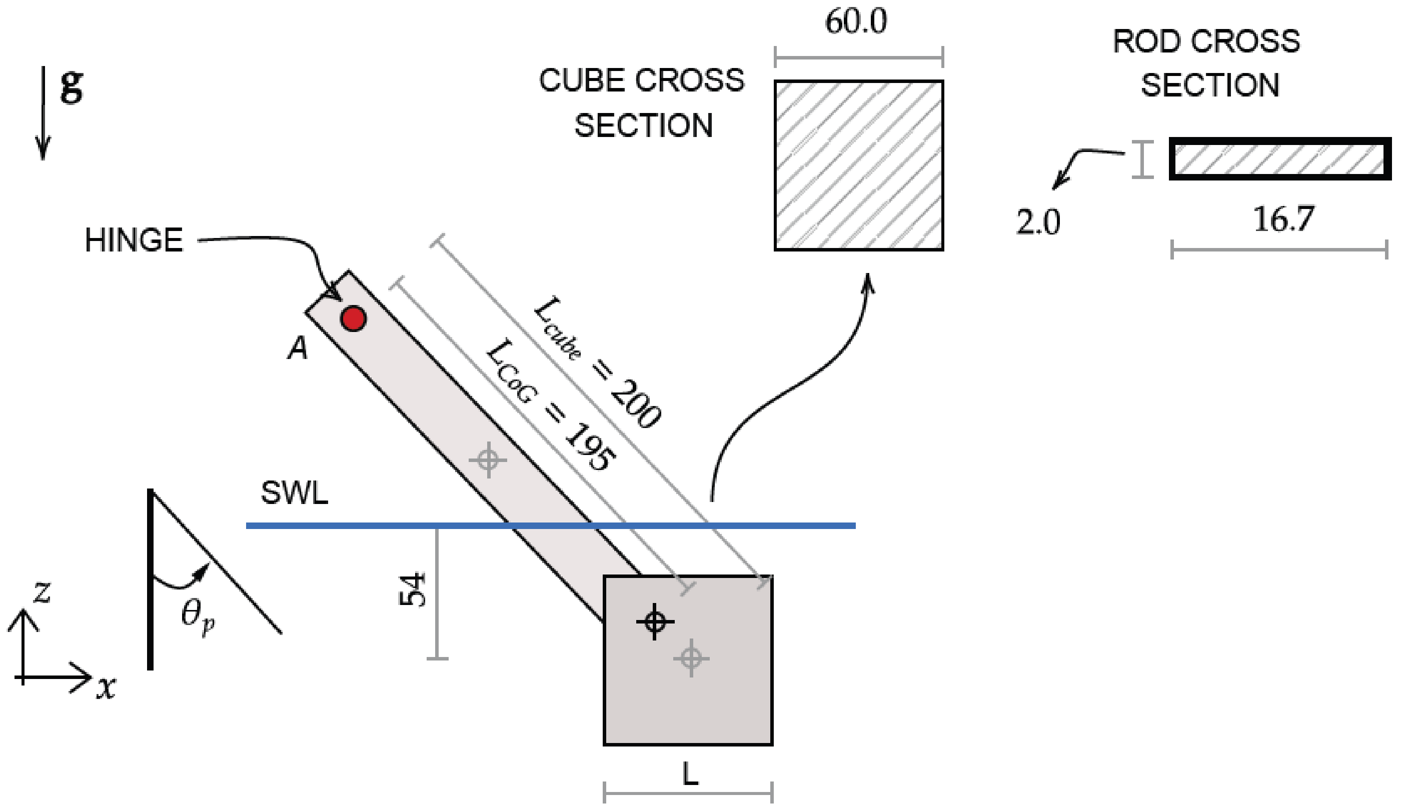
Validation

Multibody dynamics

Exp. data in **Arnold et al., 2015**.

L=60 mm

Gravity pendulum (Hinge)



GRAVITY PENDULUM

Martínez-Estévez et al., 2022. **Coupling of an SPH-based solver with a multiphysics library.** Computer Physics Communications. *In Press*

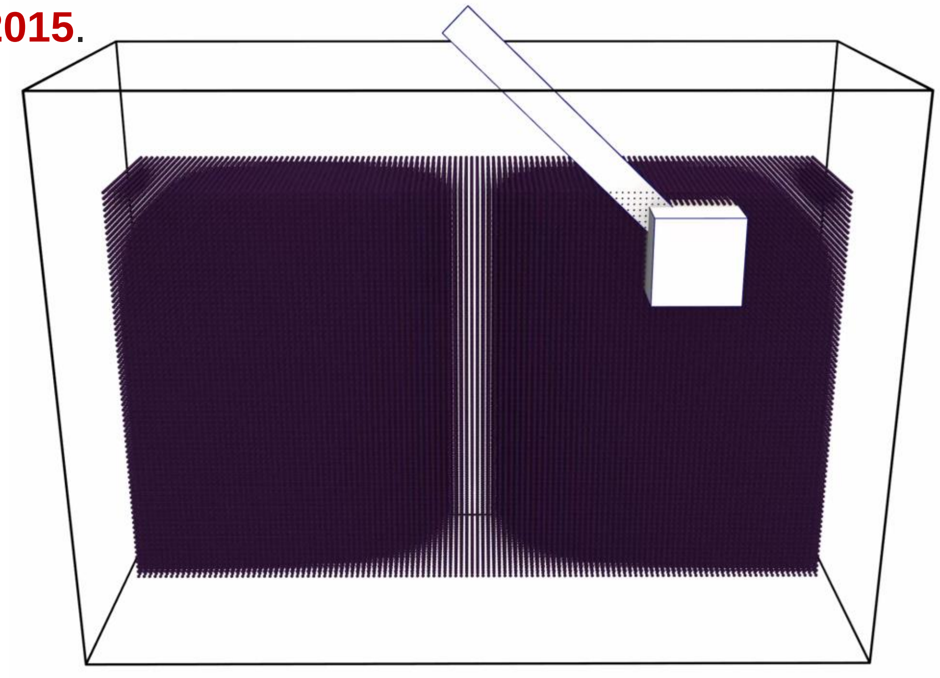
Validation

Multibody dynamics

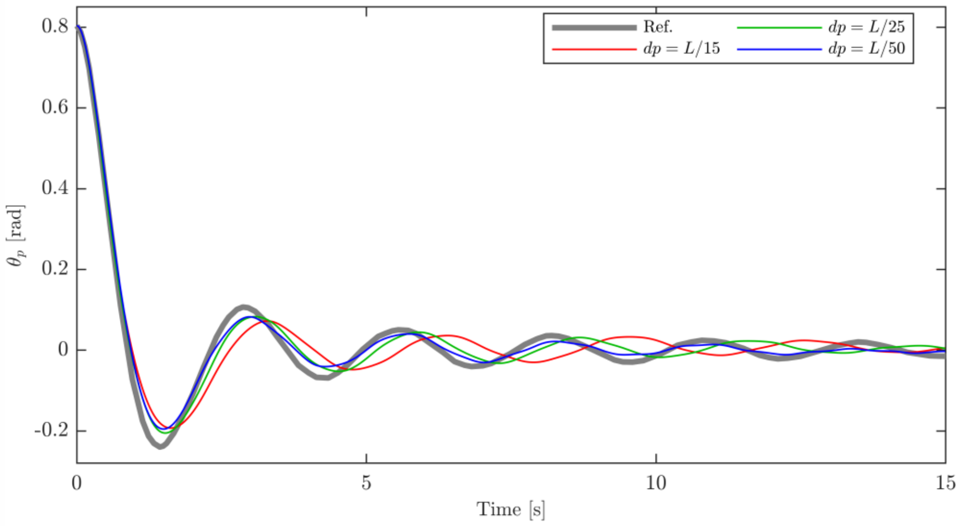
Exp. data in **Arnold et al., 2015**.

L=60 mm

Gravity pendulum (Hinge)



Comparison between numerical and experimental (grey) amplitude for the gravity pendulum in water



Martínez-Estévez et al., 2022. Coupling of an SPH-based solver with a multiphysics library. Computer Physics Communications. *In Press*

Outline

Project Chrono

Coupling SPH-Chrono

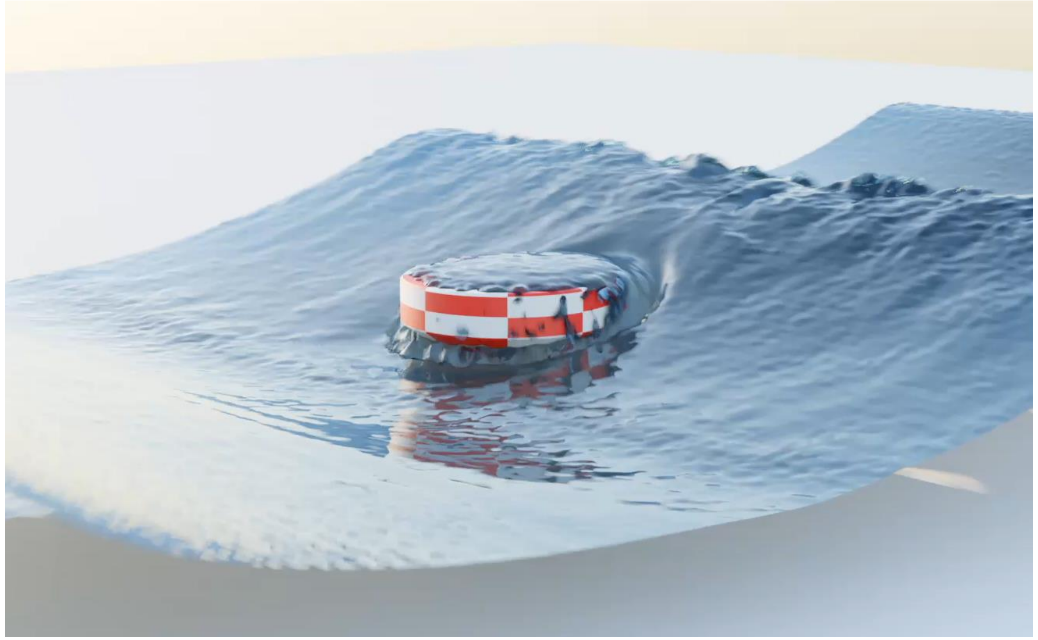
Collision detection

Multibody dynamics

Validation

Applications

Conclusions & Future Work



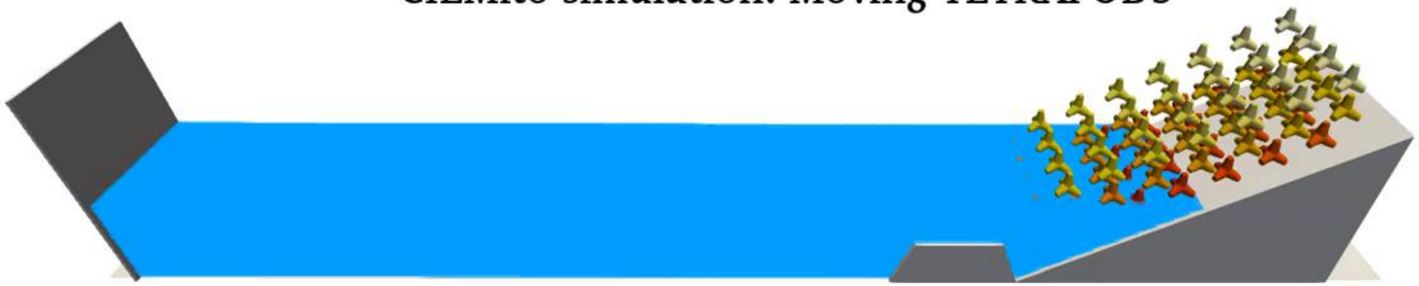
Dr. Tagliafierro

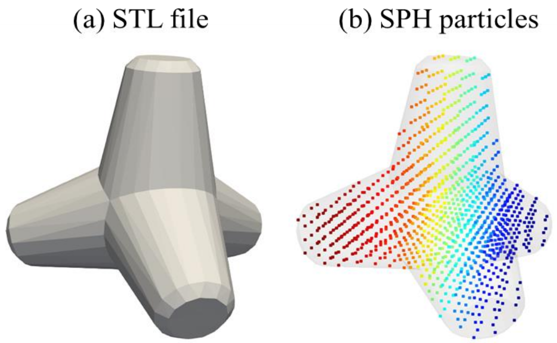
Applications

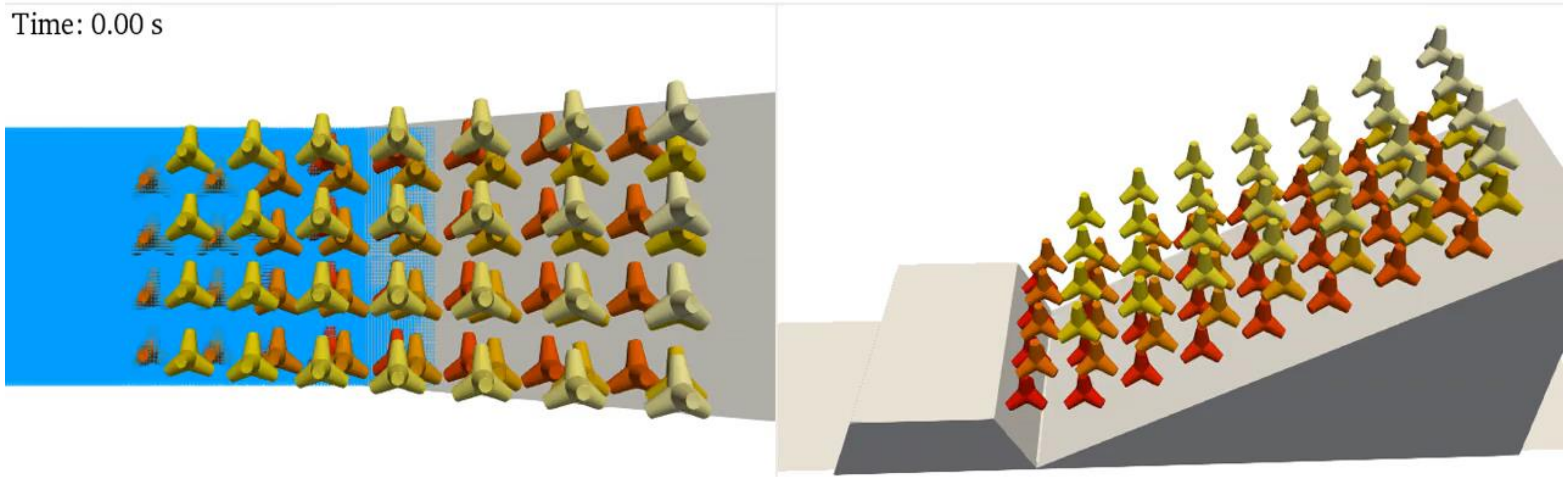
Collision detection

TETRAPODS

CIEMito simulation: Moving TETRAPODS







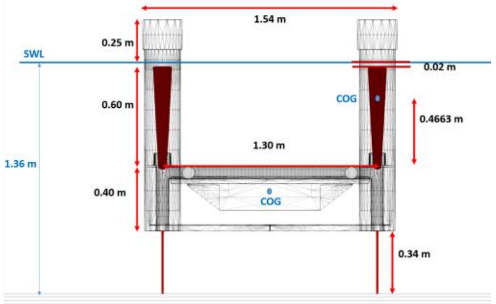
MItsui J, Altomare C, Crespo AJC, Domínguez JM, Suzuki T, Kubota S, Gómez-Gesteira M. 2021. **SPH simulation of hydrodynamic behaviour of Tetrapods against solitary wave**. In: Proceedings of the 15th International SPHERIC Workshop - SPHERIC 2021, New Jersey, US

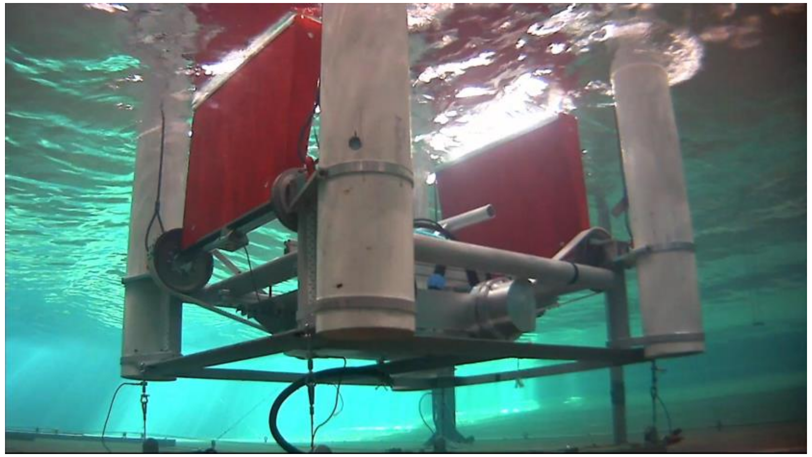
Applications

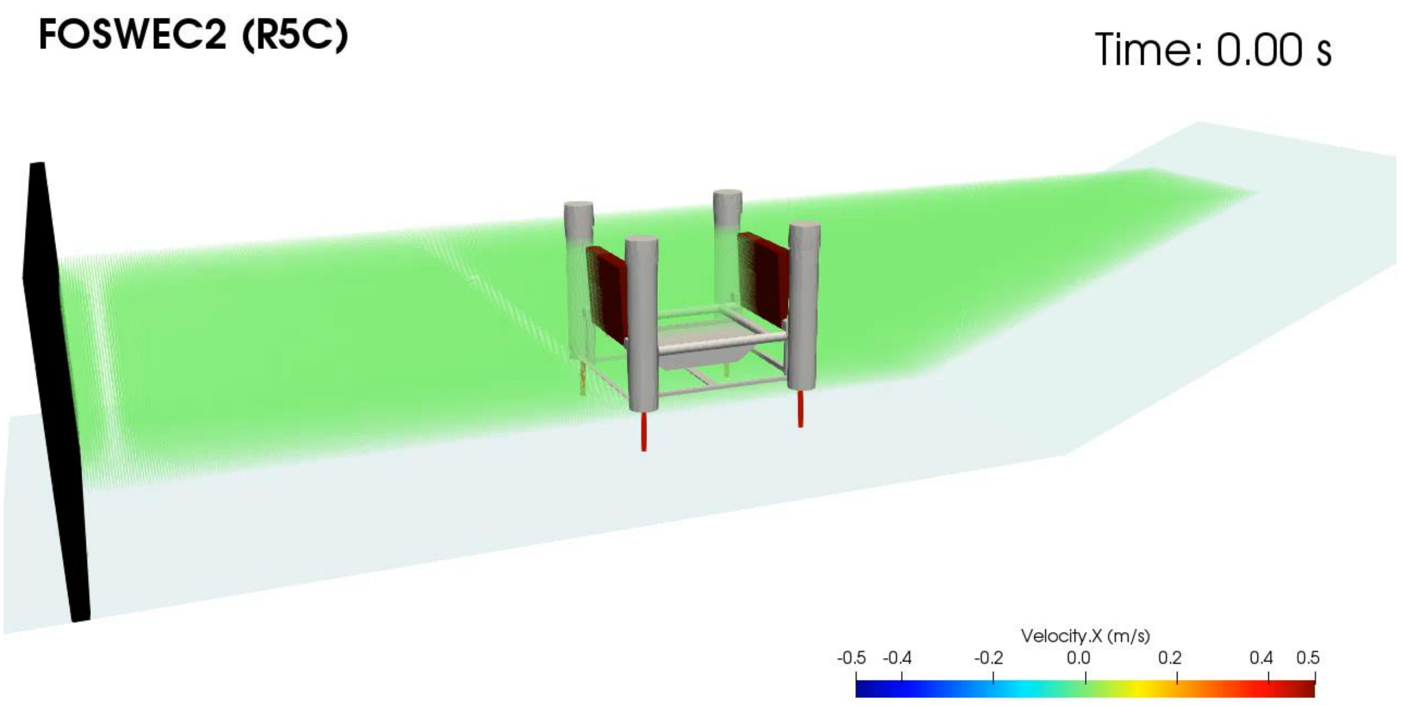
Multibody dynamics

FOSWEC2

Sandia National Laboratories





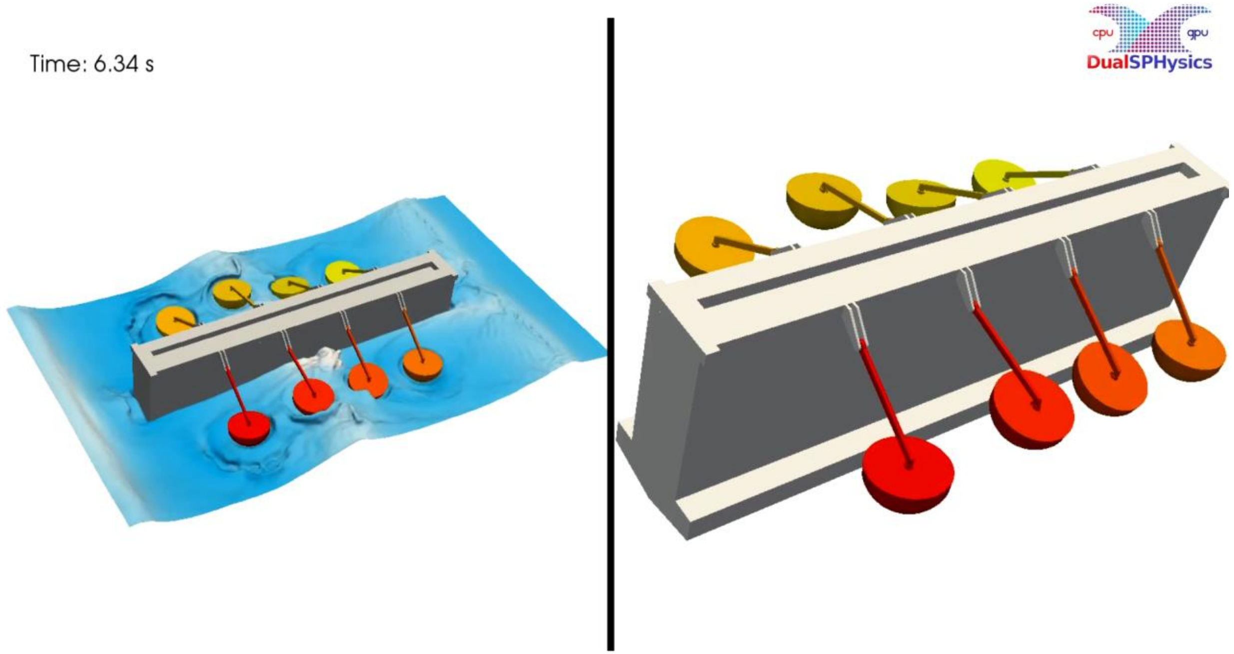


Tagliafierro B, Martínez-Estévez I, Crego-Loureiro C, Domínguez JM, Crespo AJC, Coe RG, Bacelli G, Vicione G, Gómez-Gesteira M. 2022. **Numerical Modeling of Moored Floating Platforms for Wave Energy Converters Using DualSPHysics**. In: Proceedings of the 41st International Conference on Ocean, Offshore & Arctic Engineering OMAE 2022, Hamburg, Germany.

Applications

Multibody dynamics

WAVE STAR



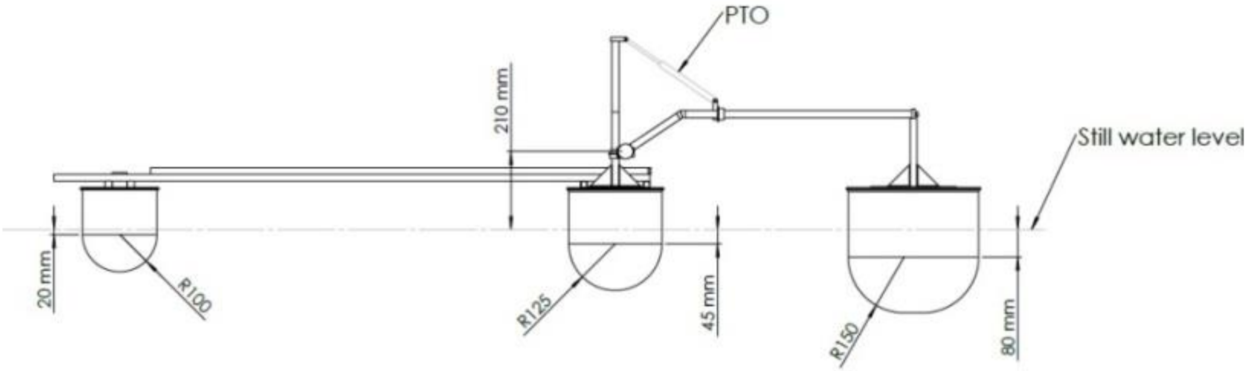
Canelas RBC, Crespo AJC, Brito M, Domínguez JM, García-Feal O. 2018. **Extending DualSPHysics with a Differential Variational Inequality: modeling fluid-mechanism interaction**. Applied Ocean Research, 76: 88-97.

Applications

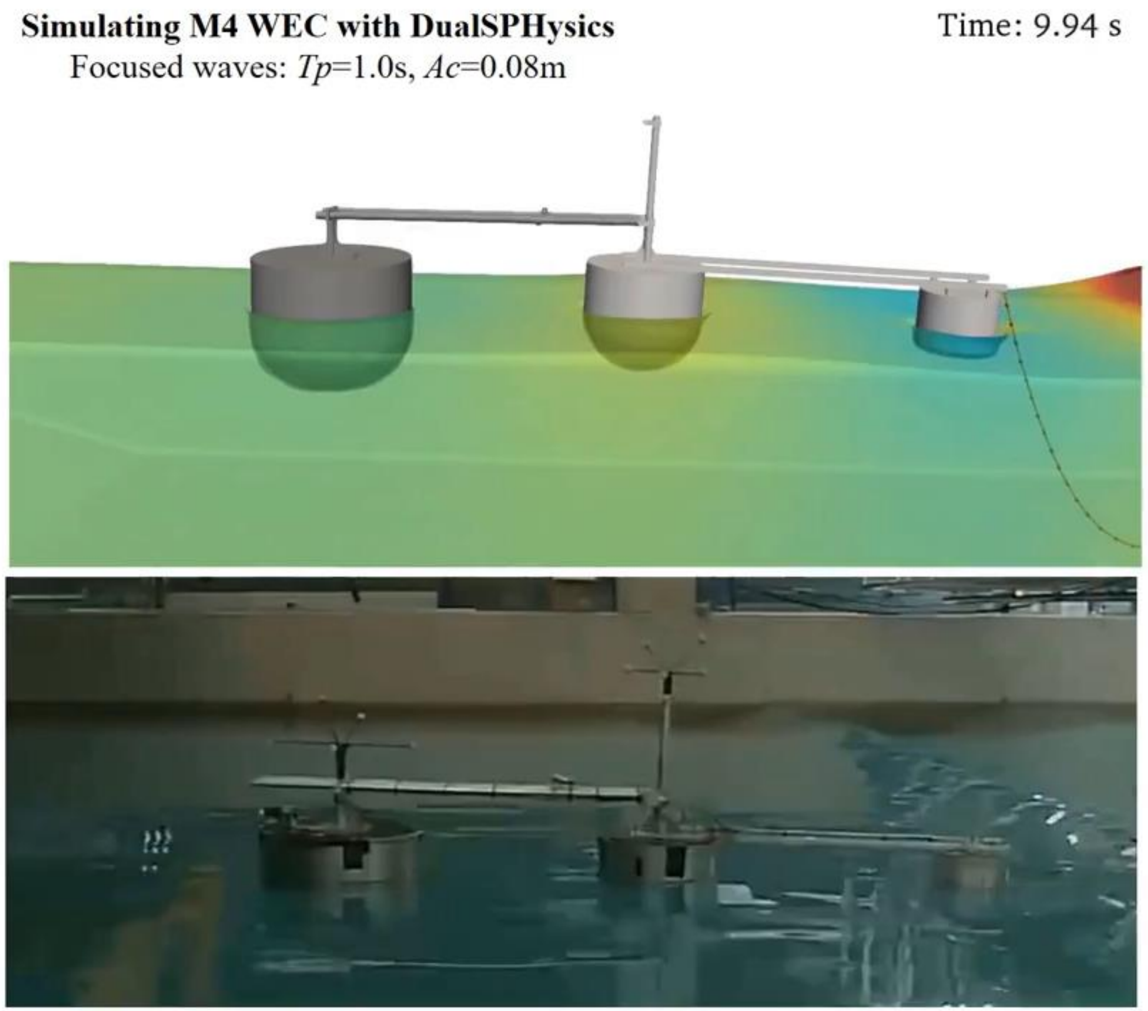
Multibody dynamics

MULTIFLOATER M4

The University of Manchester



Stansby et al., 2017



Carpintero Moreno E, Fourtakas G, Crespo AJC, Stansby PK. 2020. **Response of the multi-float WEC M4 in focussed waves using SPH**. In: 4th International Conference on Renewable Energies Offshore, Lisbon, Portugal.

Outline

Project Chrono

Coupling SPH-Chrono

Collision detection

Multibody dynamics

Validation

Applications

Conclusions & Future Work

CONCLUSIONS & FUTURE WORK

Conclusions

Coupling between solvers makes the code more versatile.

DualSPHysics-Chrono can handle multiphysics simulations.

DualSPHysics-Chrono can be used to study WECs and coastal protections.

Future Work

Additional mechanical constrains such as rotational springs.

Use the GPU and multi-core version of Project Chrono.

REFERENCES

Domínguez, J. M., Fourtakas, G., Altomare, C., Canelas, R. B., Tafuni, A., García-Feal, O., Martínez-Estévez, I., Mokos, A., Vacondio, R., Crespo, A. J. C., Rogers, B. D., Stansby, P. K., & Gómez-Gesteira, M. (2022). DualSPHysics: From fluid dynamics to multiphysics problems. Computational Particle Mechanics, 9(5), 867–895. https://doi.org/10.1007/s40571-021-00404-2

Canelas, R. B., Brito, M., Feal, O. G., Domínguez, J. M., & Crespo, A. J. C. (2018). Extending DualSPHysics with a Differential Variational Inequality: Modeling fluid-mechanism interaction. *Applied Ocean Research*, *76*, 88-97. https://doi.org/10.1016/j.apor.2018.04.015

Tagliafierro, B., Martínez-Estévez, I., Domínguez, J. M., Crespo, A. J. C., Göteman, M., Engström, J., & Gómez-Gesteira, M. (2022). A numerical study of a taut-moored point-absorber wave energy converter with a linear power take-off system under extreme wave conditions. *Applied Energy*, *311*, 118629. https://doi.org/10.1016/j.apenergy.2022.118629

Tasora, A., Serban, R., Mazhar, H., Pazouki, A., Melanz, D., Fleischmann, J., Taylor, M., Sugiyama, H., & Negrut, D. (2016). Chrono: An Open Source Multi-physics Dynamics Engine. En T. Kozubek, R. Blaheta, J. Šístek, M. Rozložník, & M. Čermák (Eds.), High Performance Computing in Science and Engineering (pp. 19-49). Springer International Publishing. https://doi.org/10.1007/978-3-319-40361-8_2

Martínez-Estévez, I., Domínguez, J. M., Tagliafierro, B., Canelas, R., García-Feal, O., Crespo, A. J. C. & Gómez-Gesteira, M. (2022). Coupling of an SPH-based solver with a multiphysics library. *Computer Physics Communications*. *In Press*

T. Hagemeier, D. Thévenin, T. Richter, Settling of spherical particles in the transitional regime, (2020). https://doi.org/10.48550/arXiv.2009.02250

Arnold, M., Kretschmer, M., Koch, J., Cheng, P. W., & Biskup, F. (2015). A Validation Method for Fluid–Structure Interaction Simulations Based on Submerged Free Decay Experiments. Journal of Ocean and Wind Energy, 2. https://doi.org/10.17736/jowe.2015.mkr02

Updates about coupling with Project Chrono

IVÁN MARTÍNEZ-ESTÉVEZ