Coupling with MoorDyn library and with Project Chrono

ALEJANDRO J. C. CRESPO
IVÁN MARTÍNEZ-ESTÉVEZ
UNIVERSIDADE DE VIGO

OUTLINE

Motivation

Coupling of DualSPHysics with MoorDyn

- MoorDyn library
- Formulation/Functionalities/Implementation
- Validation & examples

Coupling of DualSPHysics with Project Chrono

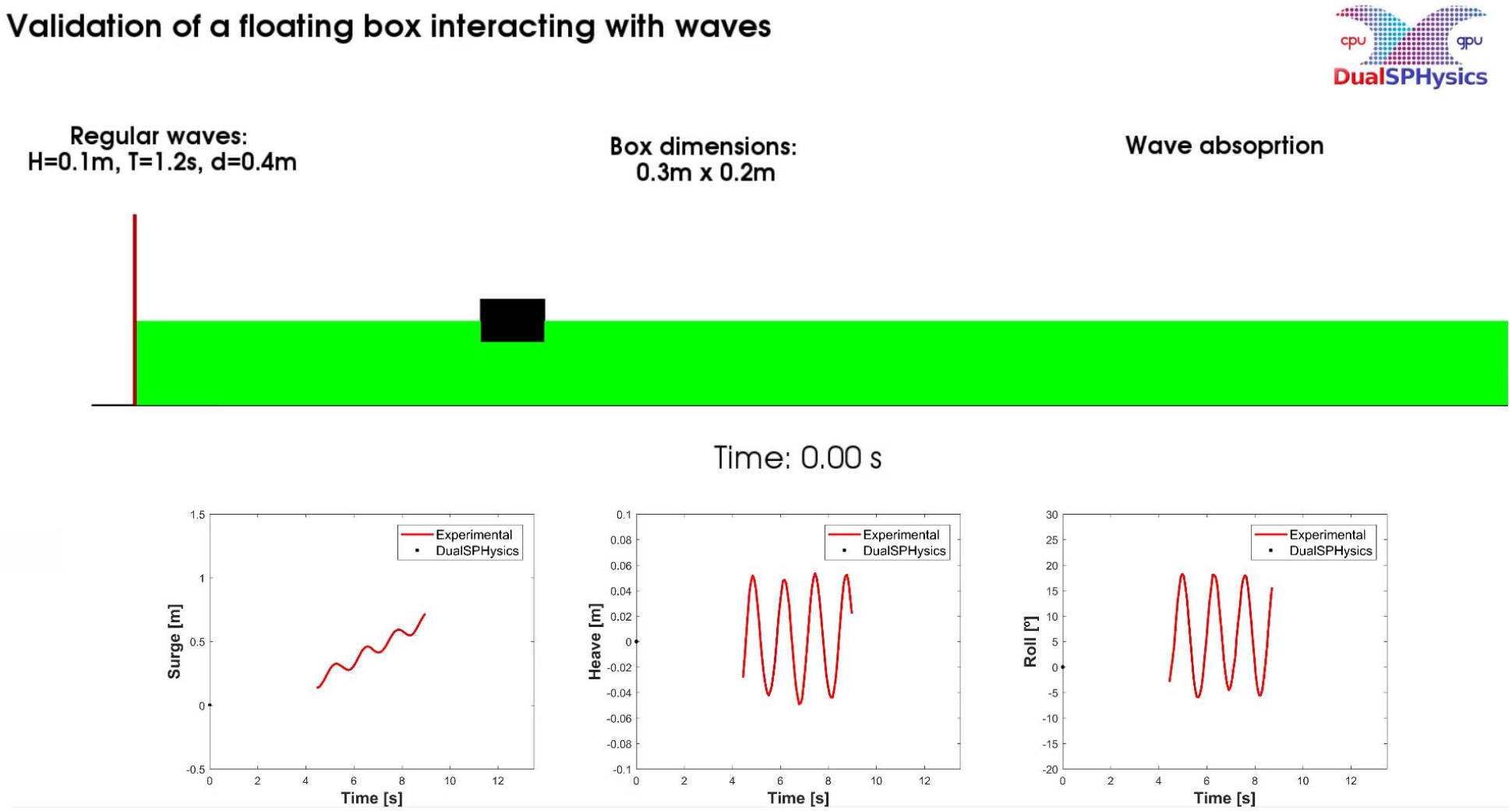
- Project Chrono library
- Formulation/Functionalities/Implementation
- Validation & examples

Main developers

Work in progress

MOTIVATION

DualSPHysics has proven to simulate with accuracy wave-floater interactions



OUTLINE

Motivation

Coupling of DualSPHysics with MoorDyn
- MoorDyn library
- Formulation/Functionalities/Implementation
- Validation & examples

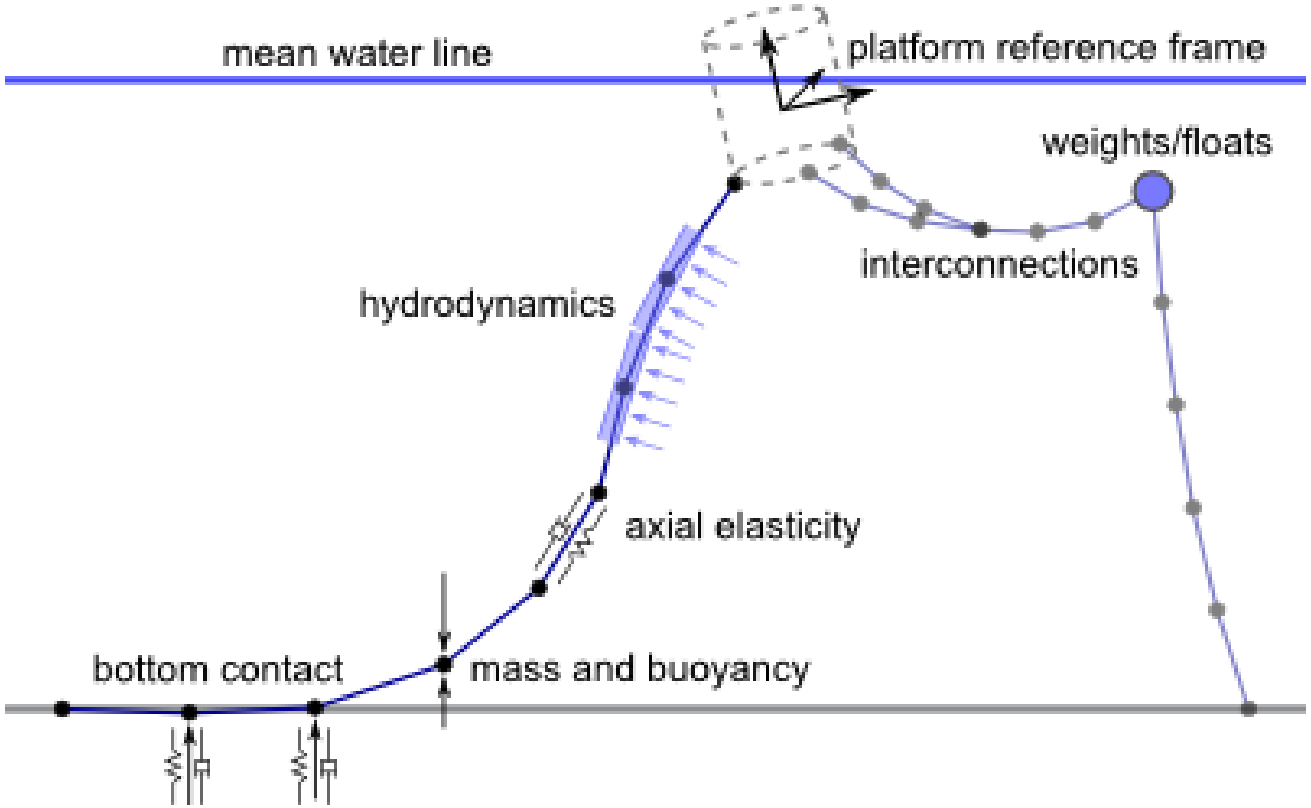
Coupling of DualSPHysics with Project Chrono
- Project Chrono library
- Formulation/Functionalities/Implementation
- Validation & examples

Main developers

Work in progress

MOORDYN LIBRARY

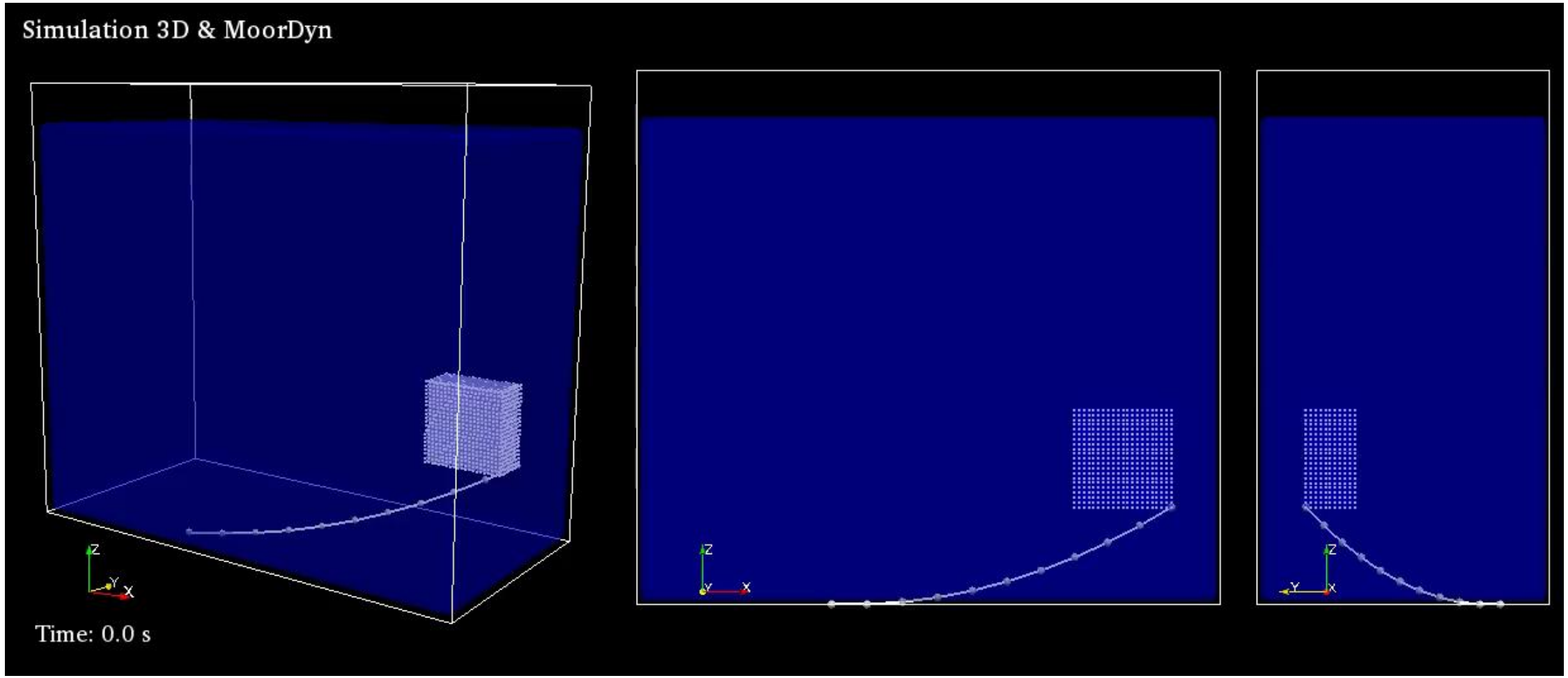
MoorDyn is an open-source dynamic mooring line model that uses a lumped-mass formulation for modelling axial elasticity, hydrodynamics, and bottom contact (Hall, 2018).



http://www.matt-hall.ca/moordyn/

MOORDYN LIBRARY

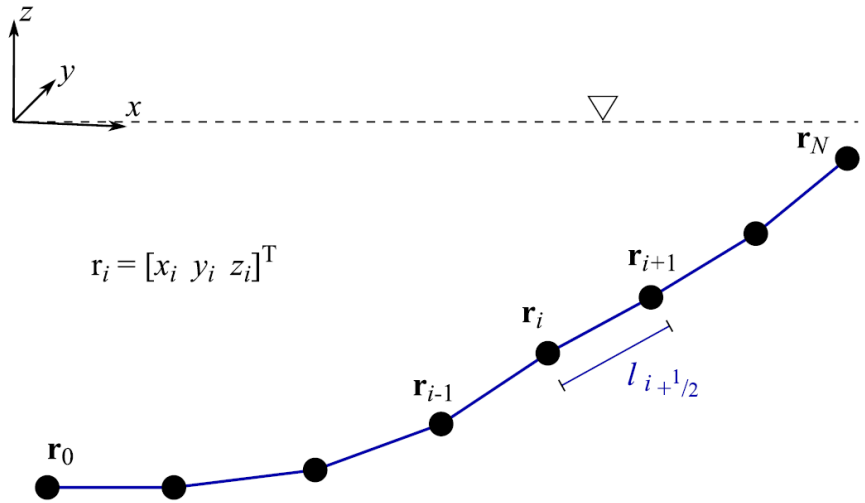
MoorDyn is an open-source dynamic mooring line model that uses a lumped-mass formulation for modelling axial elasticity, hydrodynamics, and bottom contact (Hall, 2018).

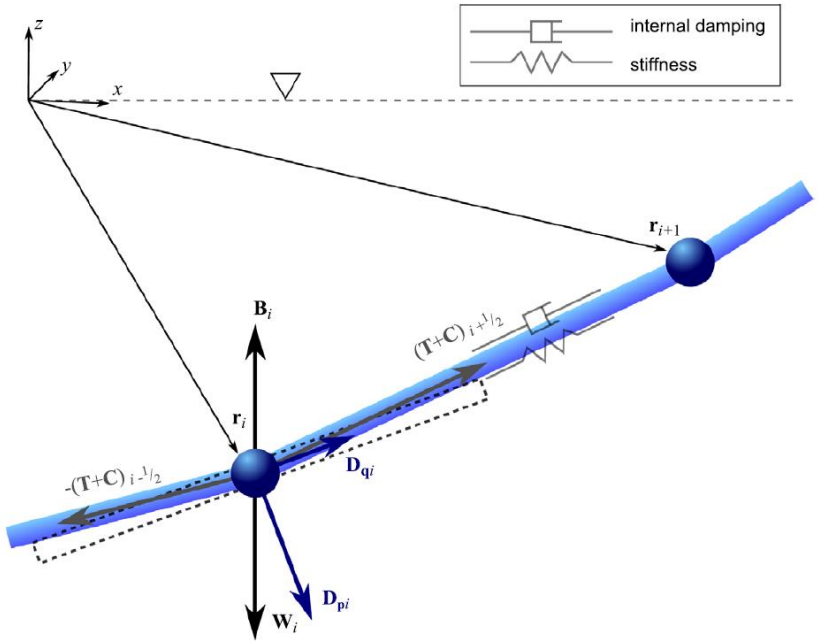


MOORDYN LIBRARY

Formulation

It discretizes mooring lines as point masses (nodes *i*) connected by linear spring-damper segments to provide elasticity in the axial direction.





If $\mathbf{r}_i$ and $\mathbf{r}_{i+1}$ represent the absolute position vectors of two adjacent nodes, the strain in the segment connecting them (i+1/2) is calculated as

$$e_{i+½} = \left(\frac{\|\mathbf{r}_{i+1} + \mathbf{r}_i\|}{l} - 1 \right)$$

where l is the segment unstretched length.

MOORDYN LIBRARY

Formulation

The total equation of motion for each node along a **mooring line *i***:

$$(\mathbf{m}_{\mathbf{node},i} + \mathbf{a}_i)\frac{\partial^2 \mathbf{r}_i}{\partial t^2} = \mathbf{T}_{i+½} - \mathbf{T}_{i-½} + \mathbf{C}_{i+½} - \mathbf{C}_{i-½} + \mathbf{W}_{\mathbf{sub},i} + \mathbf{B}_i + \mathbf{D}_{ni} + \mathbf{D}_{ti}$$

$\mathbf{m}_{\mathbf{node}}$	node mass
$\mathbf{a}$	added mass on each node
$\mathbf{W}_{\mathbf{sub}}$	submerged weight
$\mathbf{T}$	tension force due to material stiffness
$\mathbf{C}$	tension force due to internal damping
$\mathbf{B}$	vertical seabed contact forces
$\mathbf{D}_n$	drag force in the transverse direction
$\mathbf{D}_t$	drag force in the tangential direction

MOORDYN LIBRARY

Functionalities

MoorDyn+

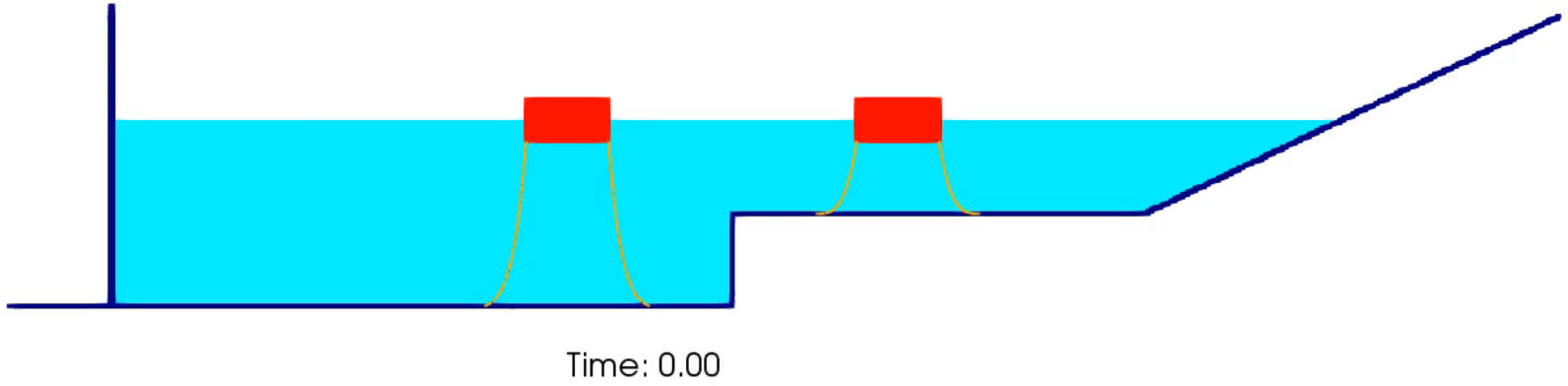
- Solves catenary equation
- Considers axial stiffness and friction with the bottom
- Bugs in MoorDyn are solved
- Robust control of exceptions
- Option to include more than one moored floating objects
- Use of different water depths in the same simulation
- Mooring connected to more than one floating object
- Define a maximum value of tension for the mooring lines

https://github.com/imestevez/MoorDynPlus

MOORDYN LIBRARY

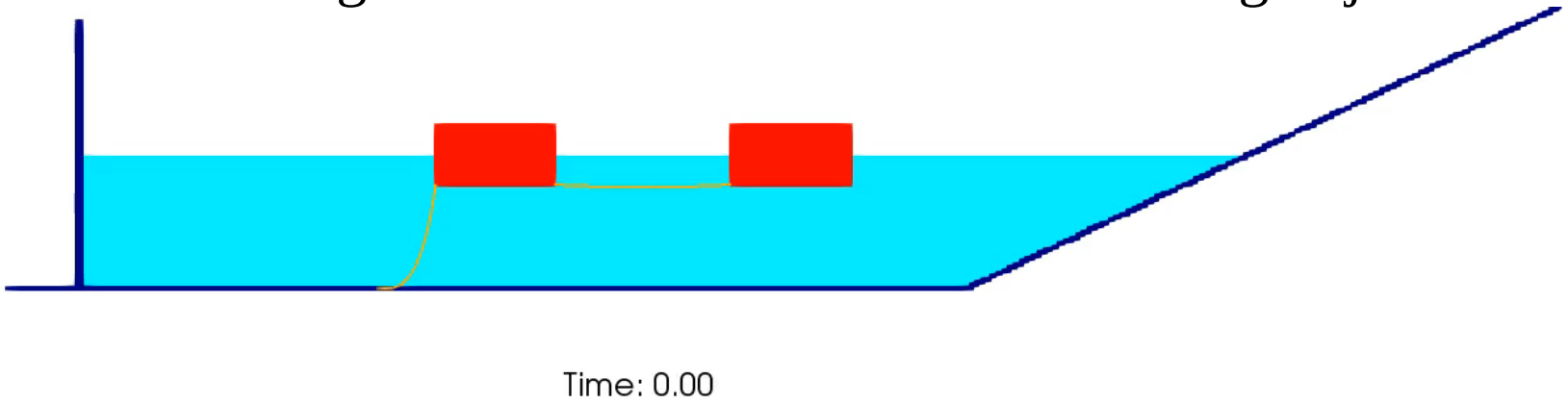
Functionalities

Use of different water depths in the same simulation



MoorDyn+

Mooring connected to more than one floating object



https://github.com/imestevez/MoorDynPlus

MOORDYN LIBRARY

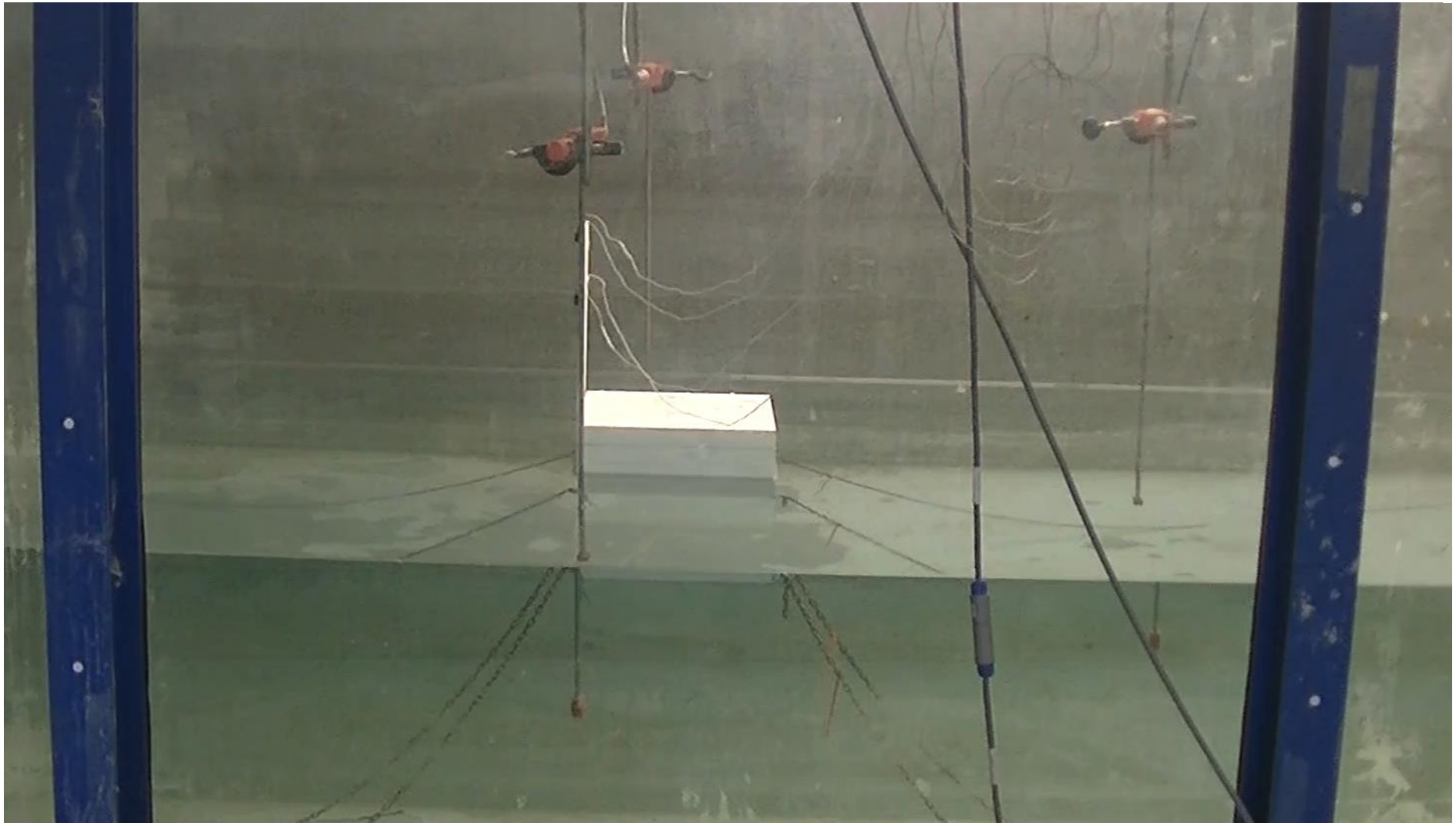
Functionalities

MoorDyn+

https://github.com/imestevez/MoorDynPlus

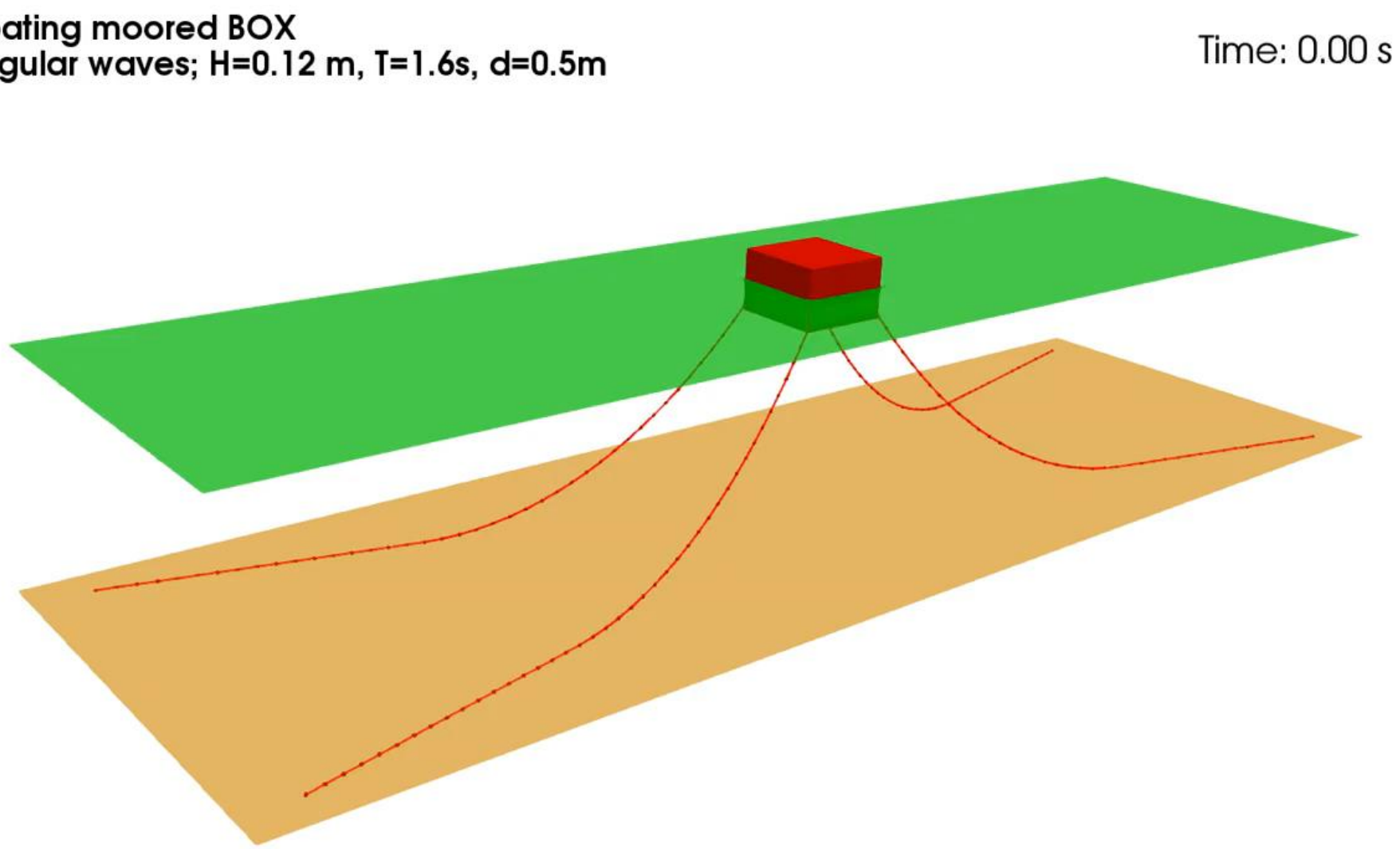
COUPLING DUALSPHYSICS+MOORDYN

Validation



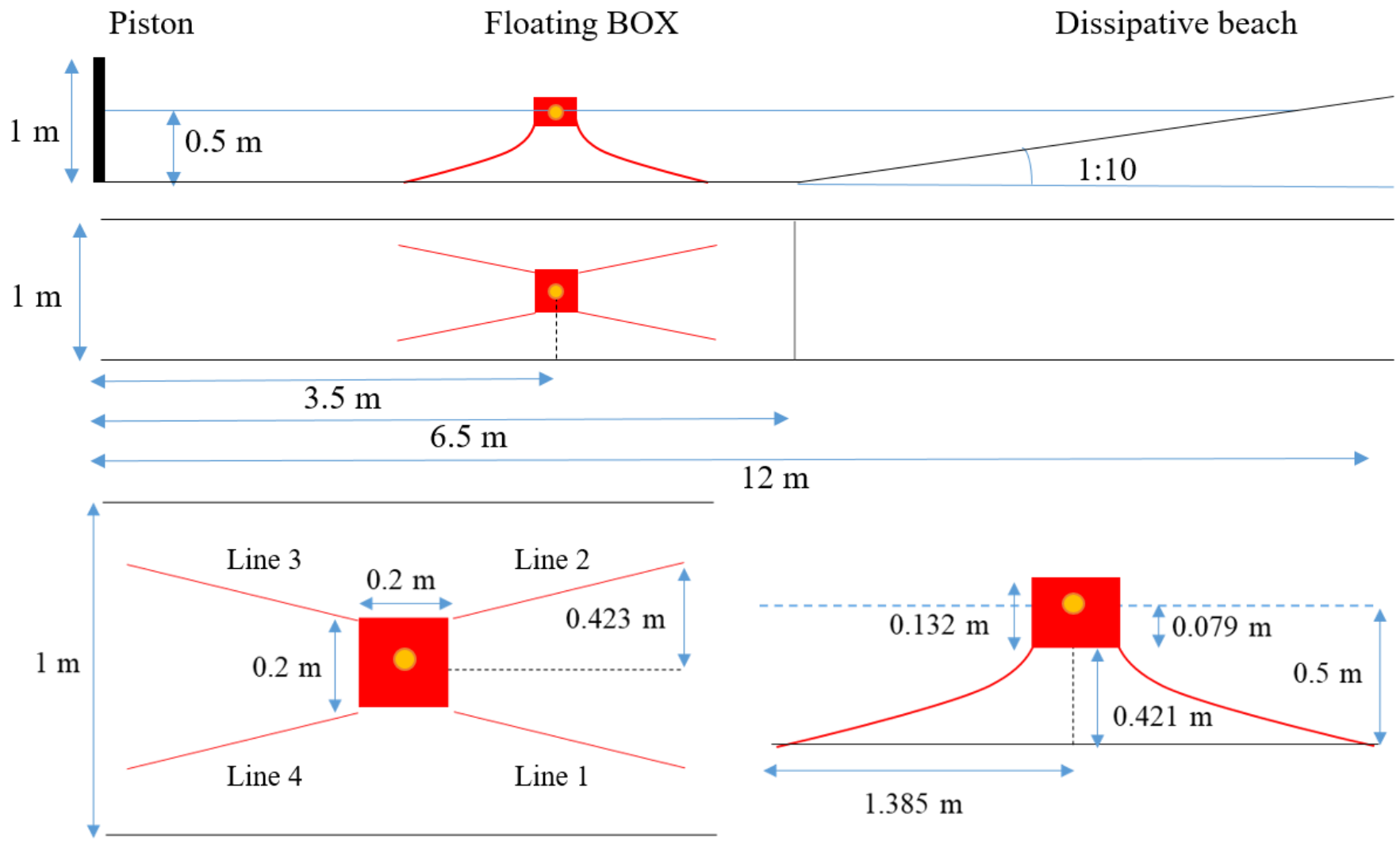
COUPLING DUALSPHYSICS+MOORDYN

Validation



COUPLING DUALSPHYSICS+MOORDYN

Validation



COUPLING DUALSPHYSICS+MOORDYN

Validation

WAVES

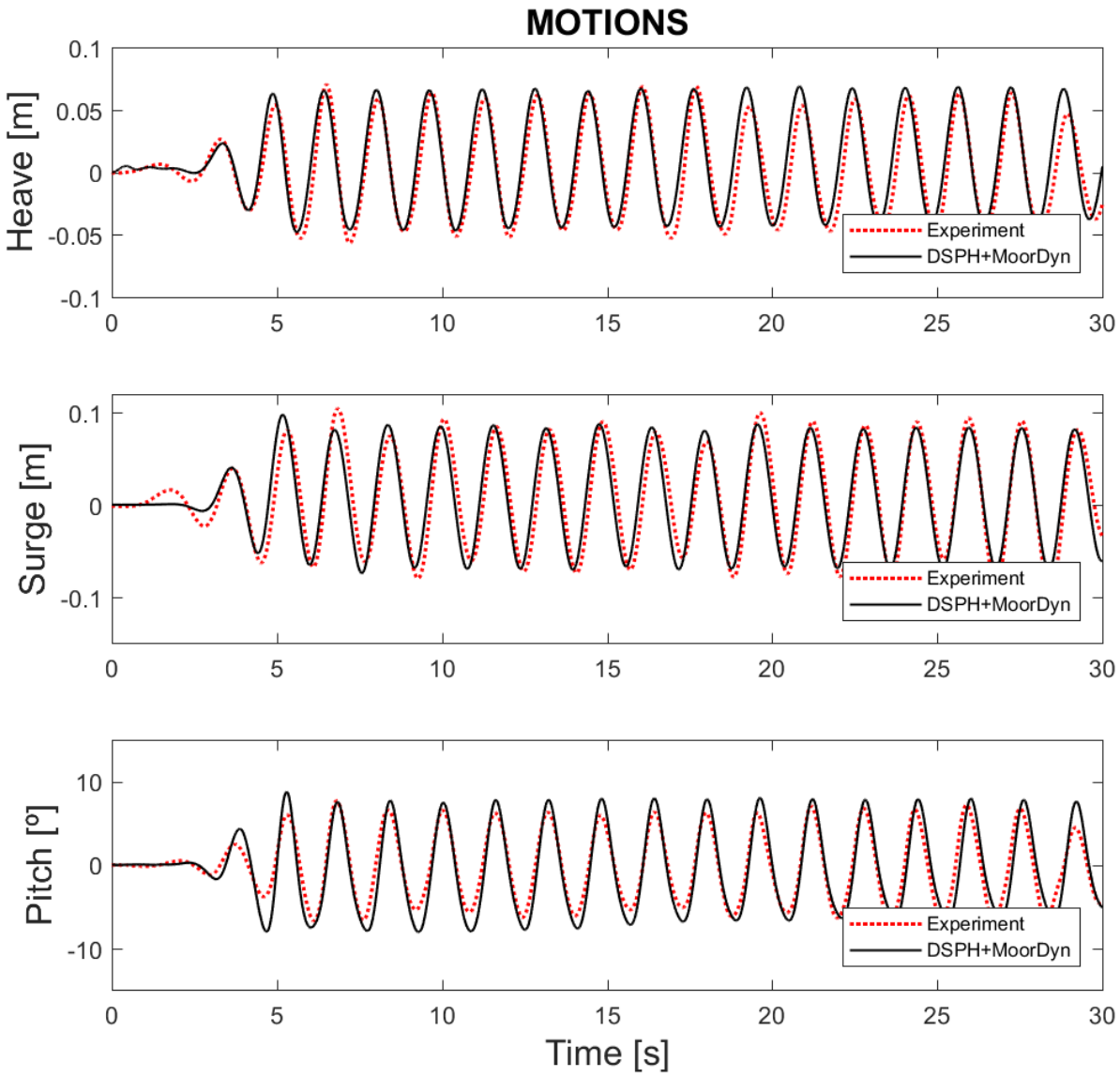
ID name	*T* (s)	*H* (m)	*d* (m)	*L* (m)	*H/L*
Case1	1.60	0.12	0.5	3.100	0.039
Case2	1.80	0.12	0.5	3.615	0.033
Case3	2.00	0.12	0.5	4.116	0.029

INPUT DATA

Parameter	Value
box length	20 cm
box width	20 cm
box height	13.2 cm
box weight (+ connections)	3.6 kg
centre of gravity of the box (X,Y,Z)	(0, 0, -1.26) cm
box lip draught	7.86 cm
mooring diameter	3.656 mm
mooring weight (per length)	0.607 g/cm
mooring length	145.5 cm

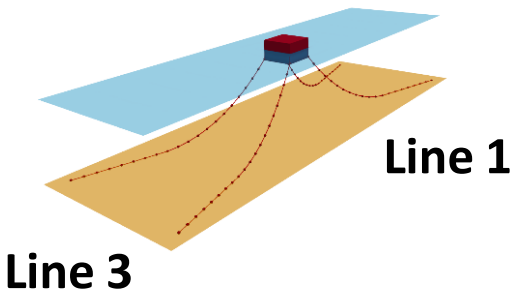
COUPLING DUALSPHYSICS+MOORDYN

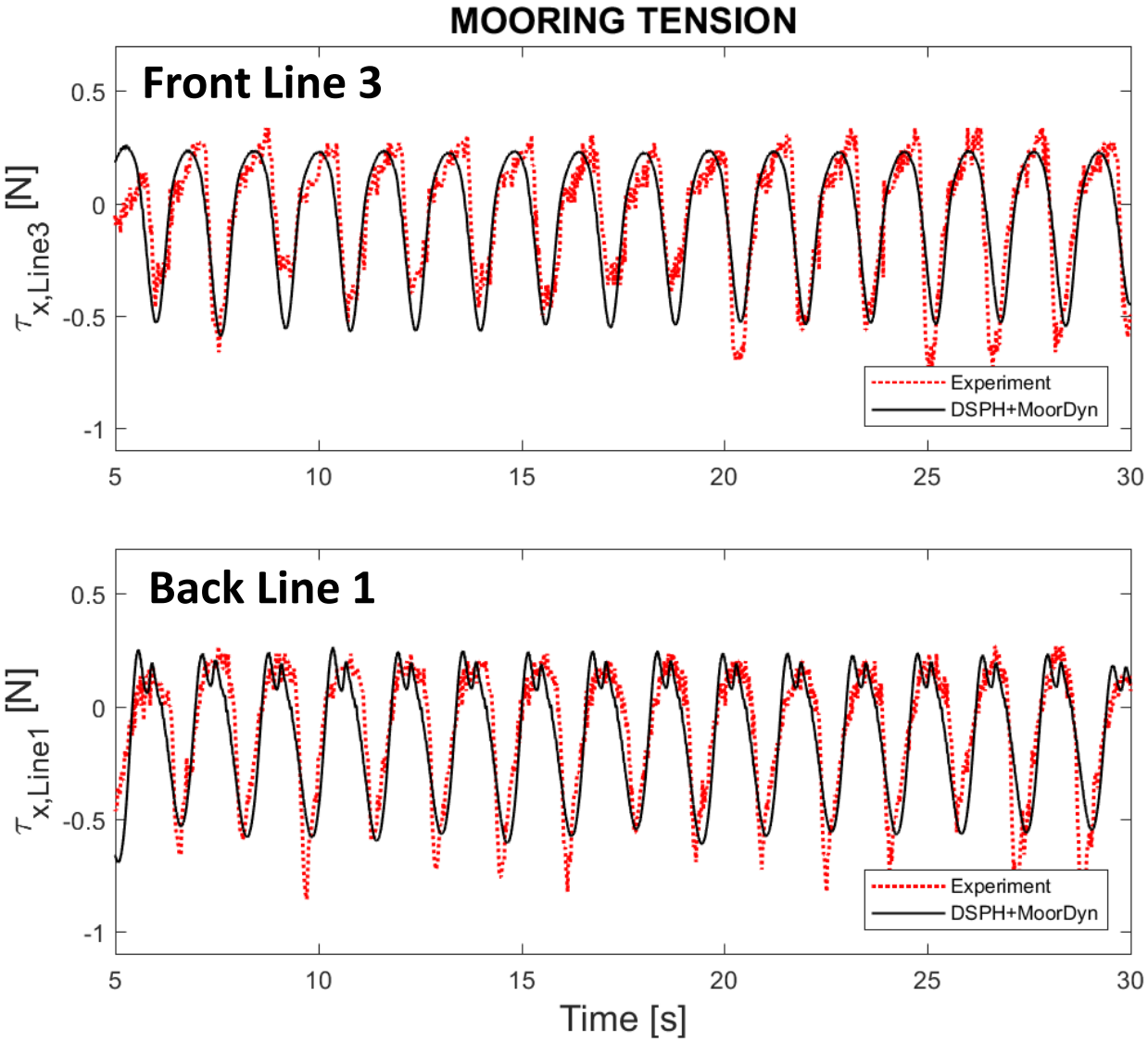
Validation



COUPLING DUALSPHYSICS+MOORDYN

Validation





COUPLING DUALSPHYSICS+MOORDYN

Validation

Domínguez et al., 2019 CENG

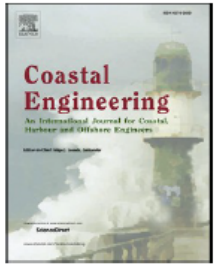
Domínguez JM, Crespo AJC, Hall M, Altomare C, Wu M, Stratigaki V, Troch P, Cappietti L, Gómez-Gesteira M. 2019. SPH simulation of floating structures with moorings. Coastal Engineering, 153, 103560. doi: 10.1016/j.coastaleng.2019.103560.

Coastal Engineering 153 (2019) 103560

Contents lists available at ScienceDirect

Coastal Engineering

journal homepage: http://www.elsevier.com/locate/coastaleng



SPH simulation of floating structures with moorings

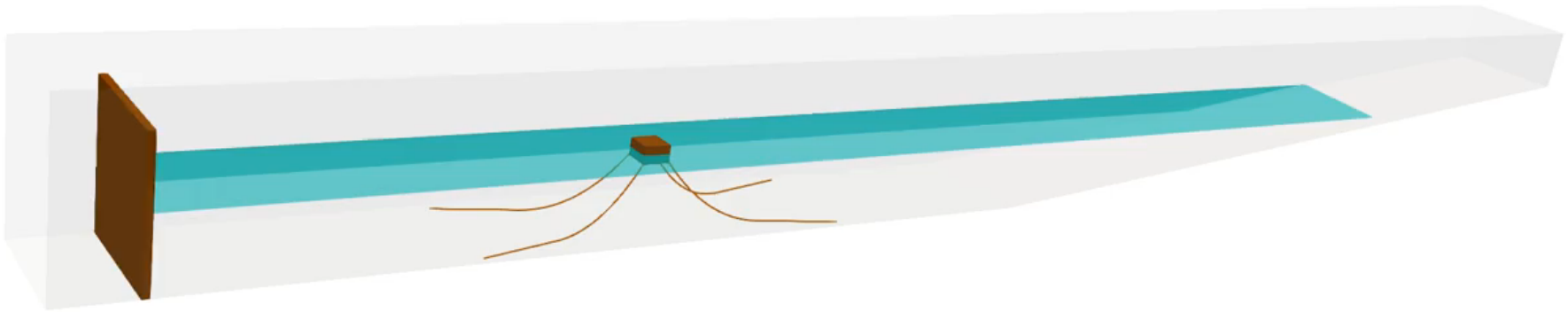
José M. Domínguez [a], Alejandro J.C. Crespo [a,*], Matthew Hall [b], Corrado Altomare [c,d], Minghao Wu [d], Vasiliki Stratigaki [d], Peter Troch [d], Lorenzo Cappietti [e], Moncho Gómez-Gesteira [a]

COUPLING DUALSPHYSICS+MOORDYN

Validation

Examples included in DualSPHysics_v5.0

CaseMoorings3D



Particles: 4,833,204
Physical time: 10 s
Runtime (GeForce RTX 2080): 4.63 hours

Time: 0.00 s

COUPLING DUALSPHYSICS+MOORDYN

XML FILE

```
<special>
    <moorings>
        <savevtk_moorings value="true" />
        <savecsv_points value="true" />
        <savevtk_points value="false" />
        <mooredfloatings>
            <floating mkbound="45" />
            <floating mkbound="50" />
        </mooredfloatings>
        <moordyn file="moordyn.xml" />
    </moorings>
</special>
```

mkbound of the moored objects

Configuration for the MoorDyn+ library can be defined in:

A) a new separated XML file

```
<moordyn file="moordyn.xml" comment="MoorDyn configuration"/>
```

B) in the same XML including this sections:

```
<moordyn>
    <solverOptions>
    <bodies>
    <lines>
    <output>
</moordyn>
```

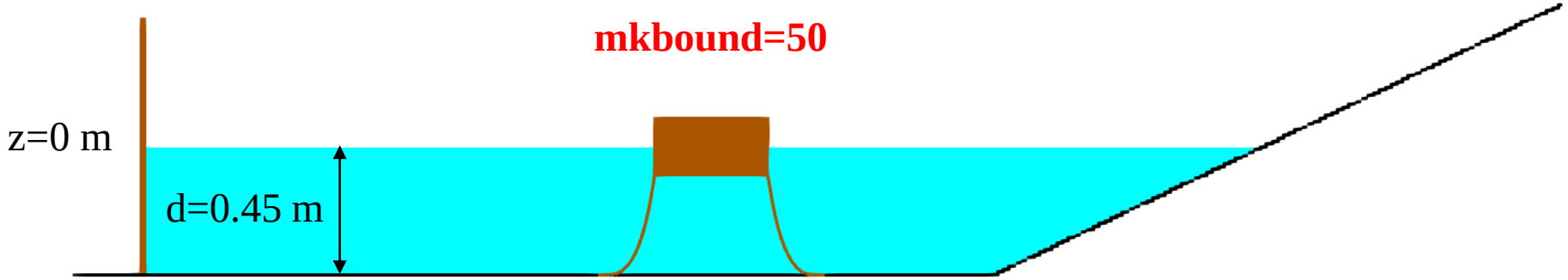
COUPLING DUALSPHYSICS+MOORDYN

XML FILE

```
<moordyn comment="MoorDyn configuration">
    <solverOptions>
    <bodies>
        <body ref="50" />
    </bodies>
    <lines>
    <output>
</moordyn>
```

ref indicates which fluid-driven object will be moored
ref==mkbound

Fluid-driven object to attach mooring lines:
mkbound=50



COUPLING DUALSPHYSICS+MOORDYN

XML FILE

```
<moordyn comment="MoorDyn configuration">
    <solverOptions>
    <bodies>
    <lines>
        <linedefault >
            <ea value="2.9e3" />
            <diameter value="3.656e-3" />
            <massDenInAir value="0.0607" />
            <ba value="-0.8" />
            <can value="1.0" />
            <cat value="0.0" />
            <cdn value="1.6" />
            <cdt value="0.05" />
            <breaktension value="500" />
            <outputFlags value="pv" />
        </linedefault>
        <line>  %line 0
        <line>  %line 1
    </lines>
    <output>
</moordyn>
```

Shared properties for each line

ea: line stiffness (N)

diameter: volume-equivalent diameter (m)

massDenInAir: mass per unit length (kg/m)

ba: internal damping (Ns)

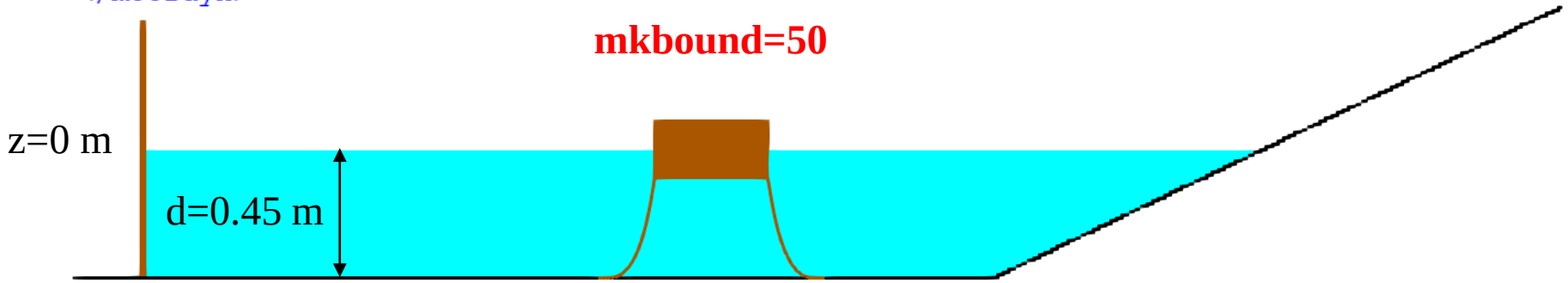
can: transverse added mass coefficient

cat: tangential added mass coefficient

cdn: transverse drag coefficient

cdt: tangential drag coefficient

breaktension: Maximum value of tension (N)



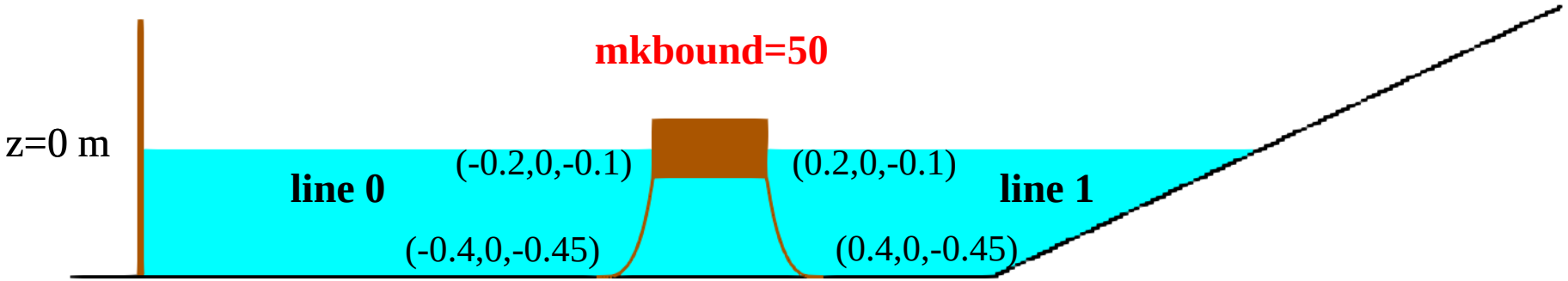
COUPLING DUALSPHYSICS+MOORDYN

XML FILE

Connects the line to a fluid-driven object

```
<line>  %line 0
    <vesselconnection bodyref="50" x="-0.2" y="0.0" z="-0.1" />
    <fixconnection x="-0.4" y="0.0" z="-0.45" />
    <length value="0.45" />
    <segments value="40" />
    <breaktension value="300" />
</line>
<line>  %line 1
    <vesselconnection bodyref="50"  x="0.2" y="0.0" z="-0.1" />
    <fixconnection x="0.4" y="0.0" z="-0.45" />
    <length value="0.45" />
    <segments value="40" />
    <breaktension value="350" />
</line>
```

vesselconnection is attached to the body with ref=50

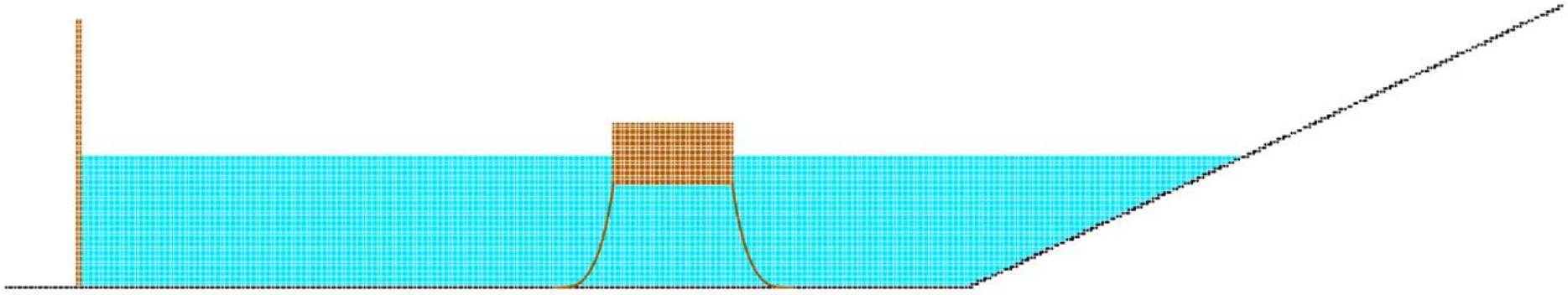


COUPLING DUALSPHYSICS+MOORDYN

XML FILE

Examples included in DualSPHysics_v5.0

CaseMoorings2D



Particles: 16,423
Physical time: 10 s
Runtime (GeForce RTX 2080): 916 s

Time: 0.00 s

OUTLINE

Motivation

Coupling of DualSPHysics with MoorDyn

- MoorDyn library
- Formulation/Functionalities/Implementation
- Validation & examples

Coupling of DualSPHysics with Project Chrono

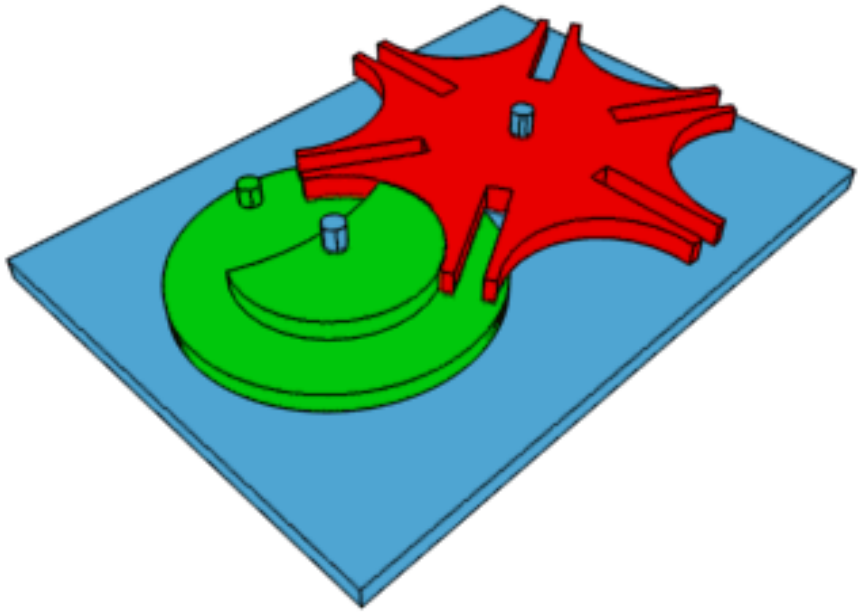
- Project Chrono library
- Formulation/Functionalities/Implementation
- Validation & examples

Main developers

Work in progress

PROJECT CHRONO LIBRARY

Project Chrono is an open-source **multi-physics** simulation engine



http://projectchrono.org

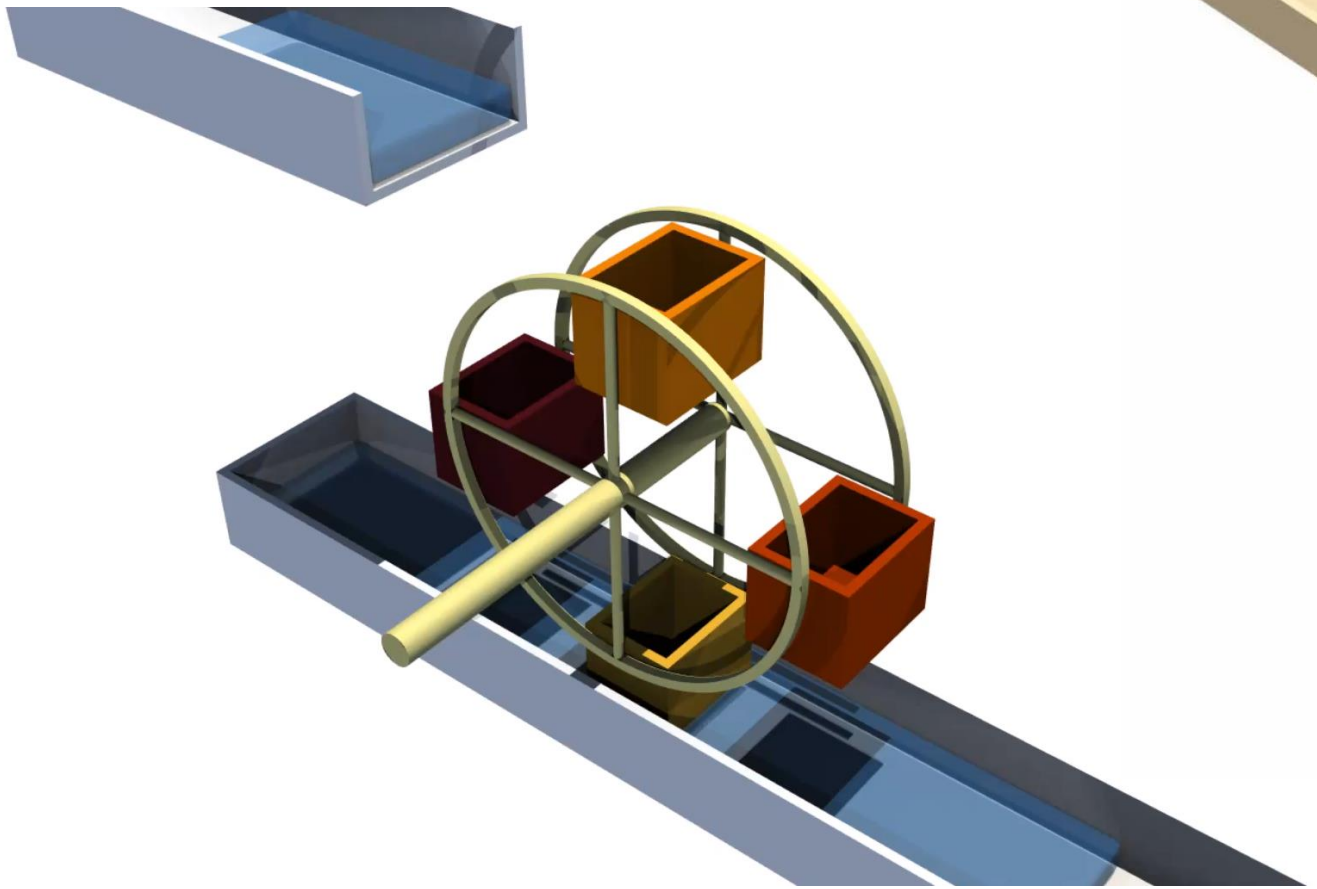
COLLISIONS:

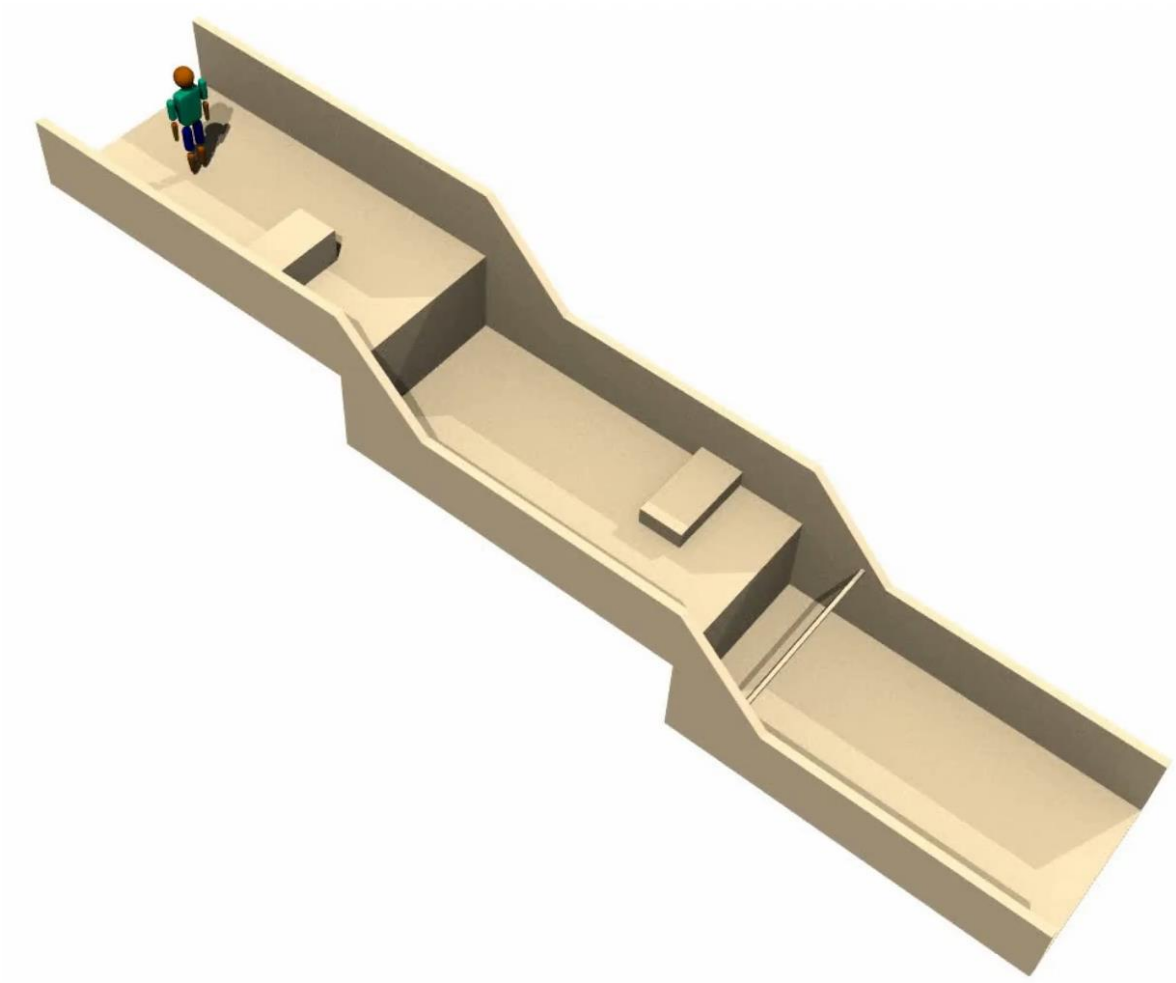
- Non-smooth contacts (NSC)
- Smooth contacts (SMC)

RESTRICTIONS:

- Spring with stiffness and damping
- Spring using Coulomb damping
- Hinge along an axis
- Spherical hinge on a point
- Sliding along an axis
- Pulley

PROJECT CHRONO LIBRARY





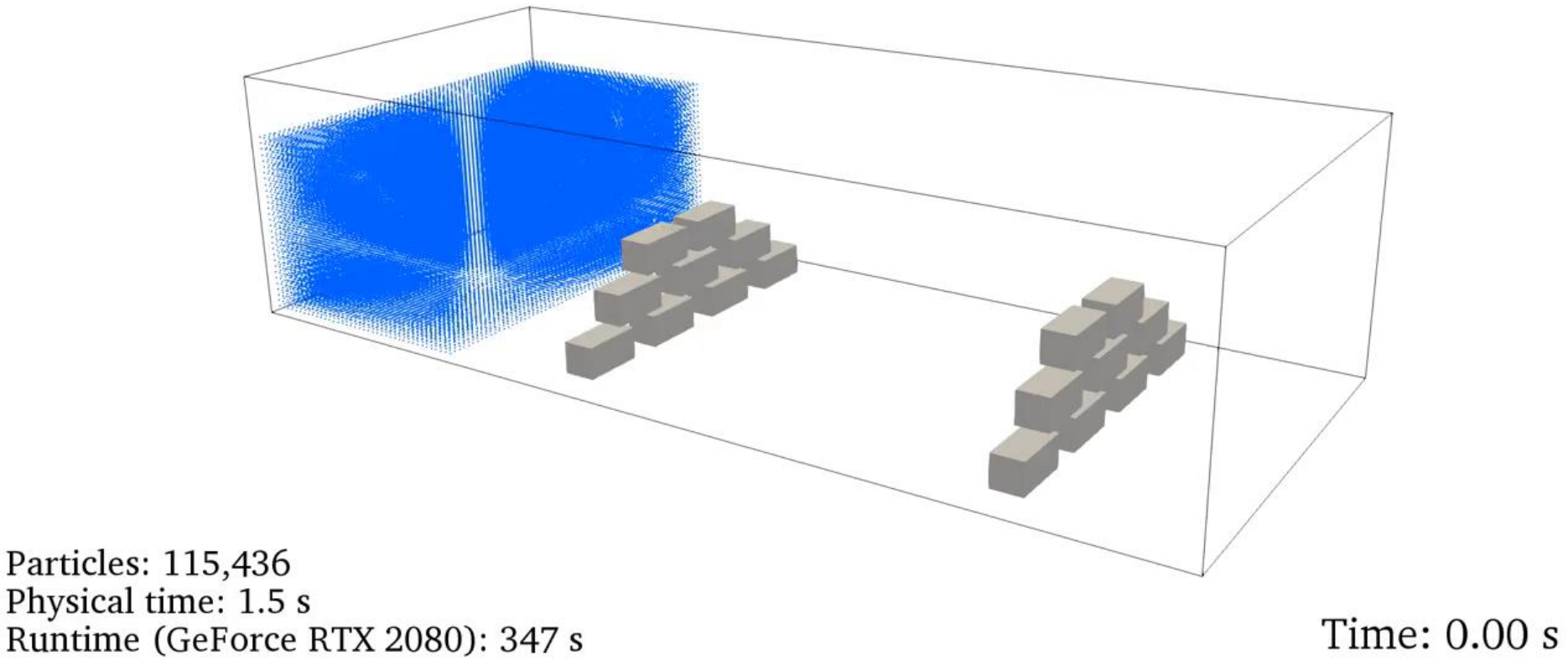
PROJECT CHRONO LIBRARY

COLLISIONS

Functionalities

Examples included in DualSPHysics_v5.0

CaseSolidsCHRONO



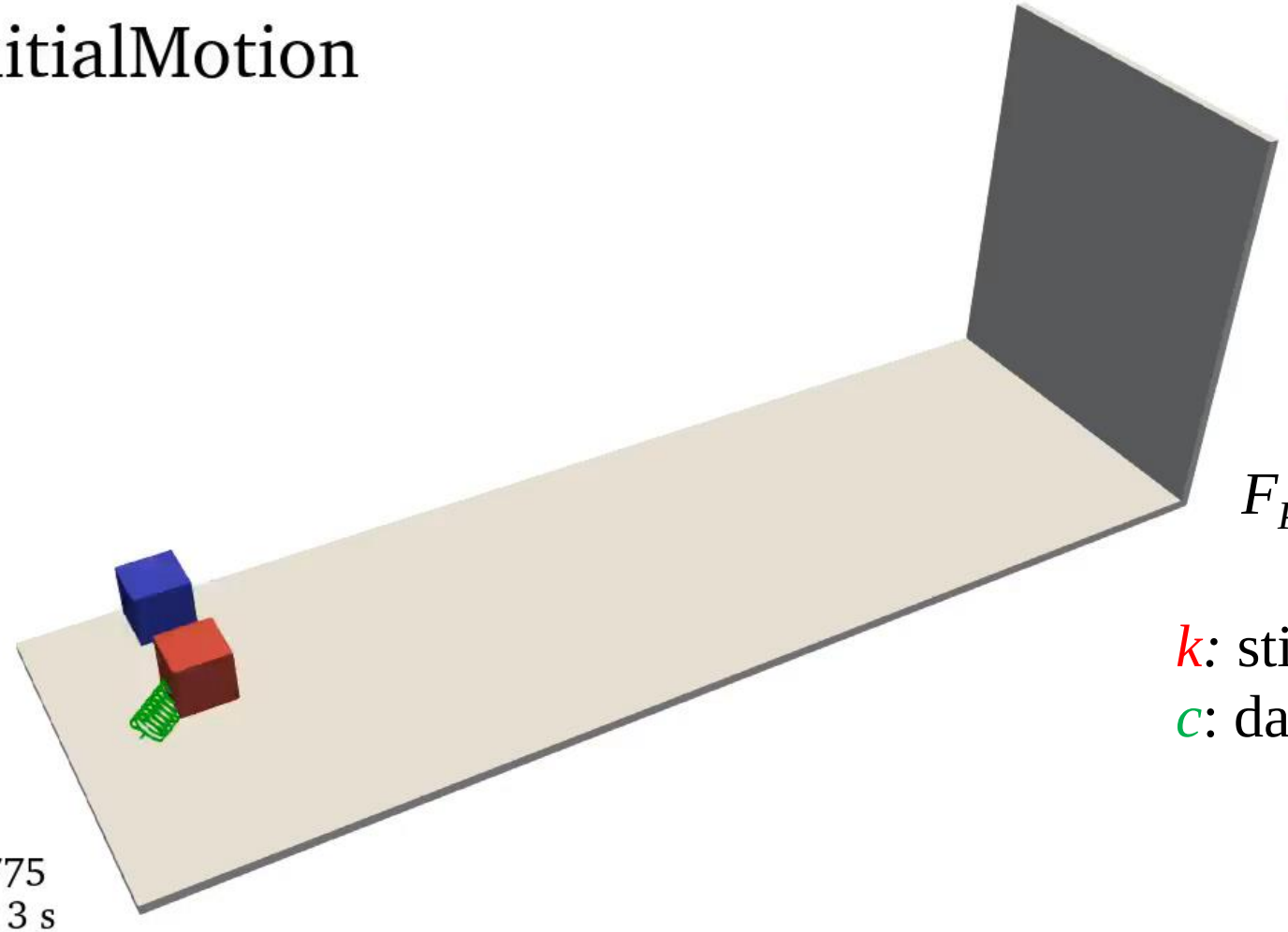
PROJECT CHRONO LIBRARY

LINEAR SPRING

Functionalities

Examples included in DualSPHysics_v5.0

CaseInitialMotion



Particles: 66,775
Physical time: 3 s
Runtime (RTX 2080 Ti): 22 s

Time: 0.00 s

$$F_{PTO}=k\cdot z+c\cdot vel.z$$

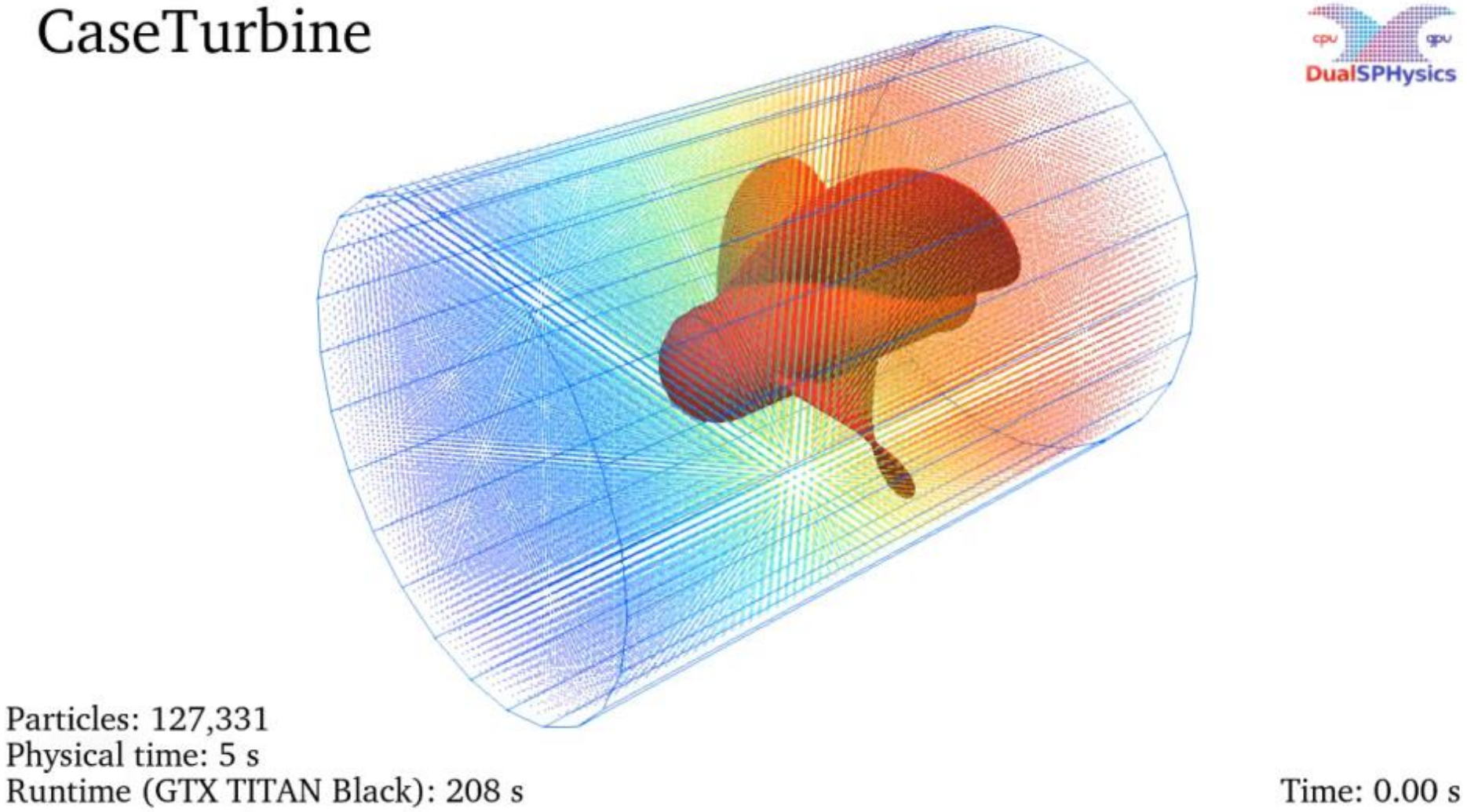
k: stiffness coefficient
c: damping coefficient

PROJECT CHRONO LIBRARY

HINGE

Functionalities

Examples included in DualSPHysics_v5.0

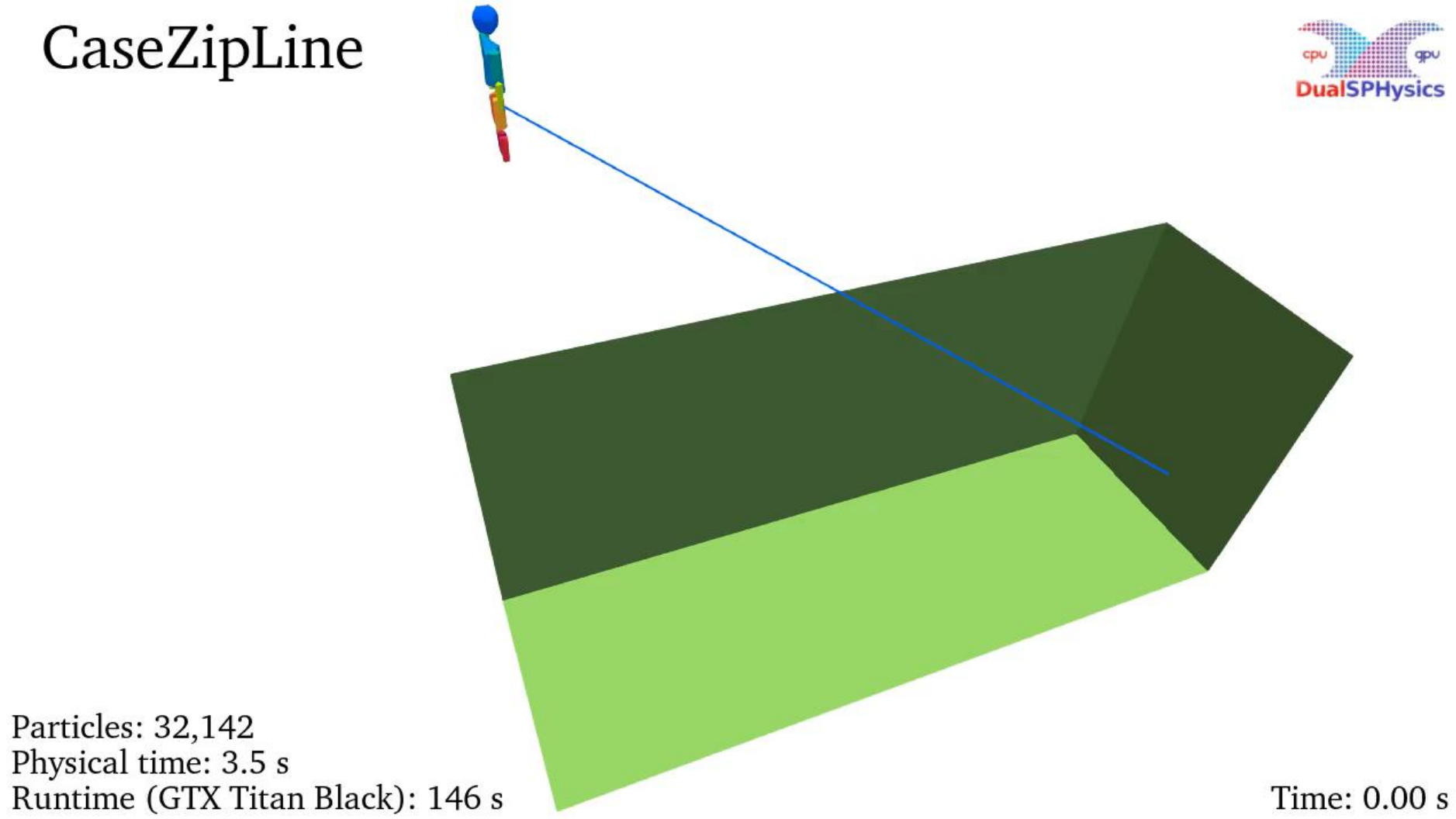


PROJECT CHRONO LIBRARY

POINT-LINE

Functionalities

Examples included in DualSPHysics_v5.0

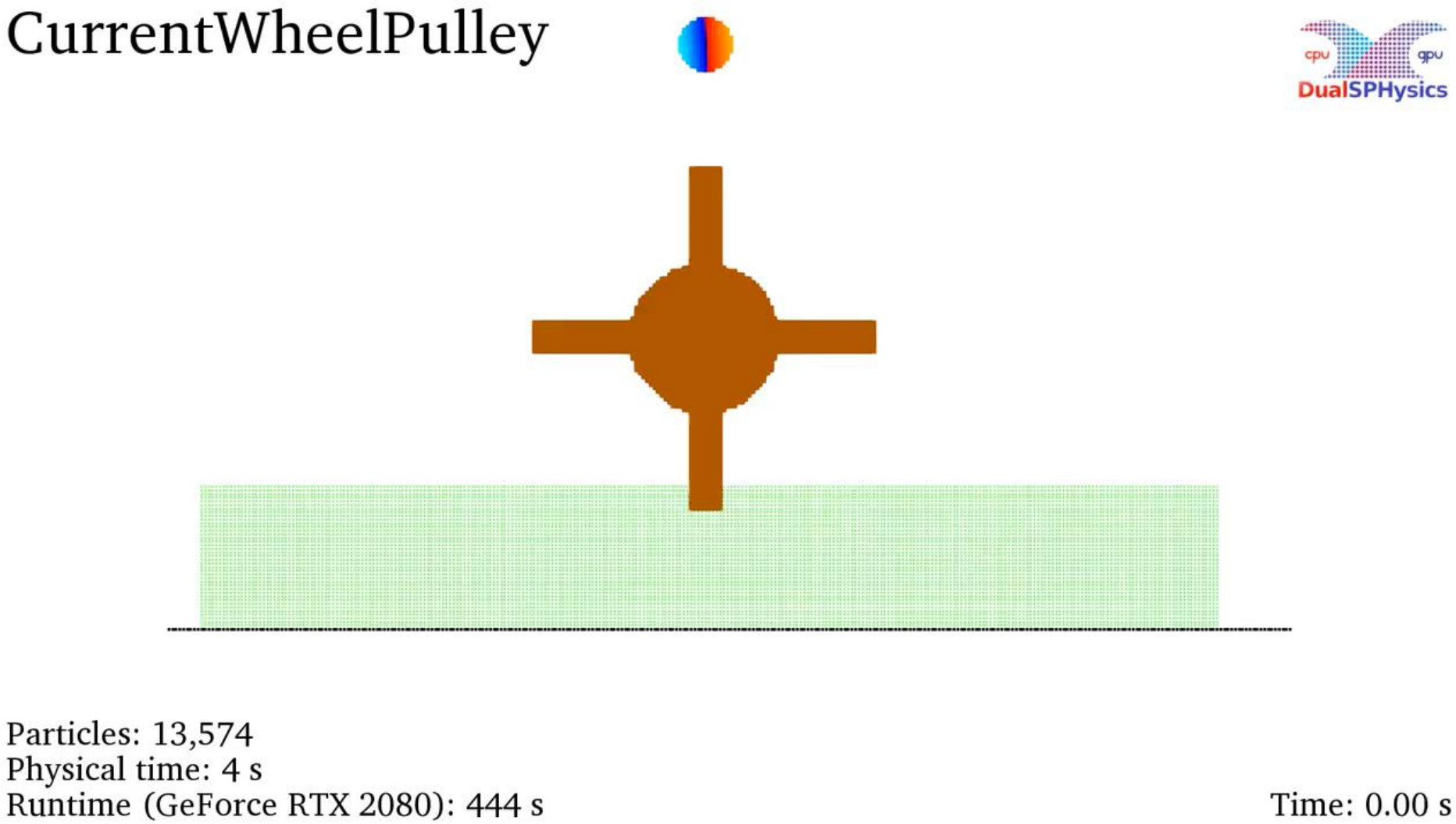


PROJECT CHRONO LIBRARY

PULLEY

Functionalities

Examples included in DualSPHysics_v5.0

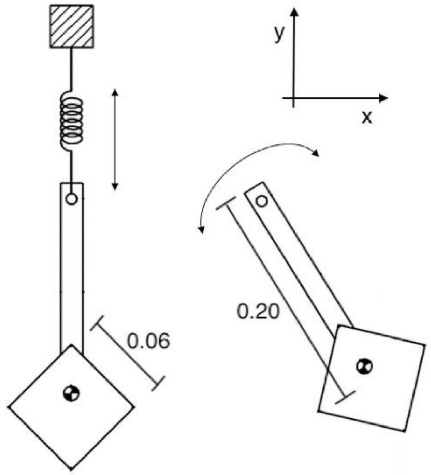


COUPLING DUALSPHYSICS+CHRONO

Canelas et al., 2018 APOR

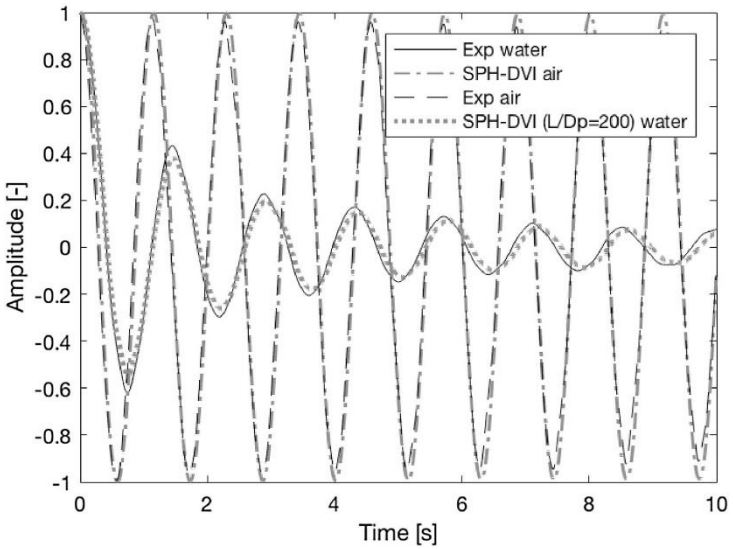
Validation

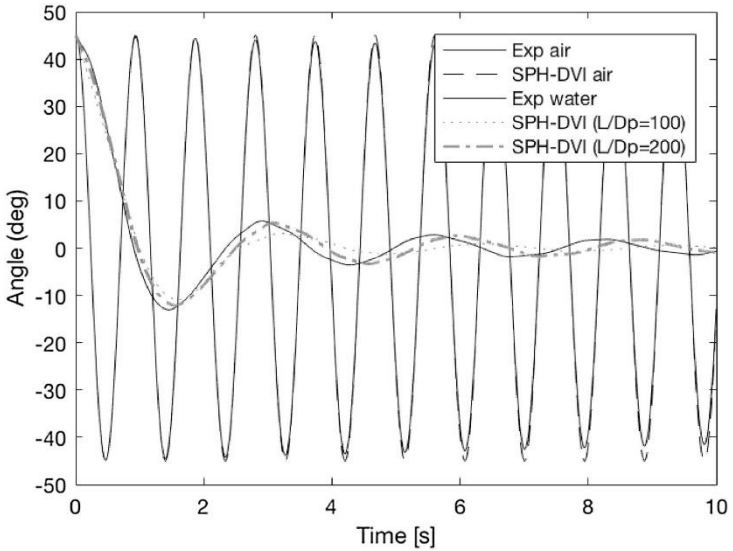
Spring pendulum



Gravity pendulum

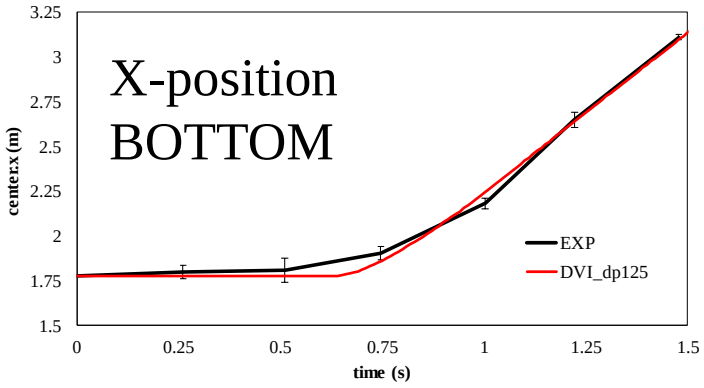
Comparison between numerical and experimental rotation angle of **spring pendulum** in air and water and **gravity pendulum** in air and water

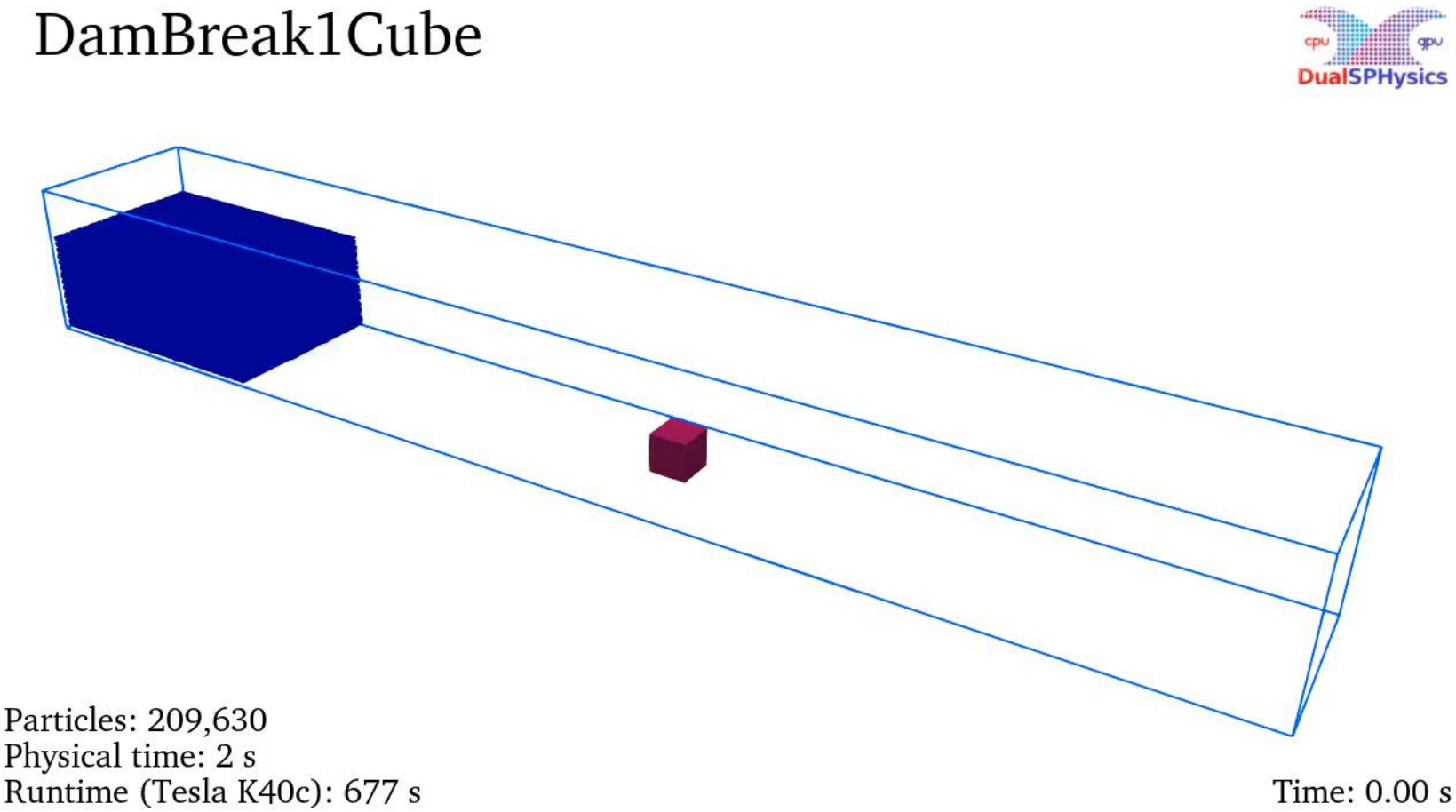




COUPLING DUALSPHYSICS+CHRONO

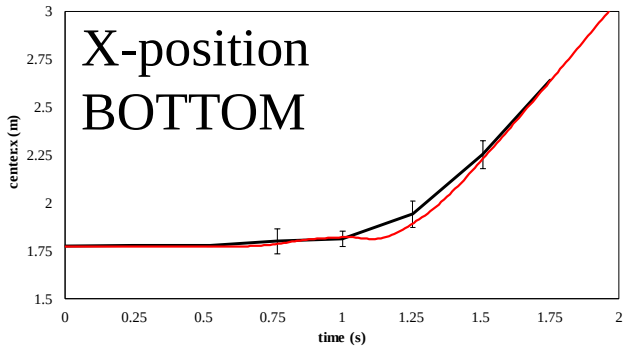
Validation

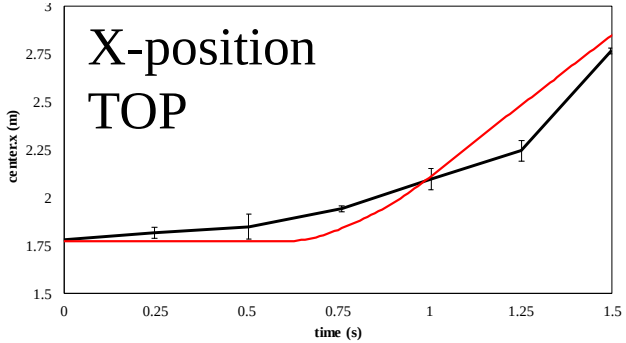


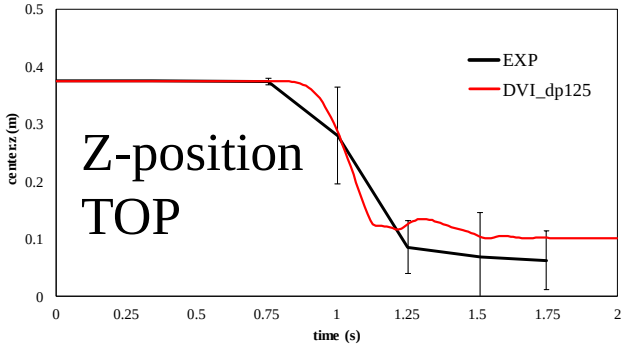


COUPLING DUALSPHYSICS+CHRONO

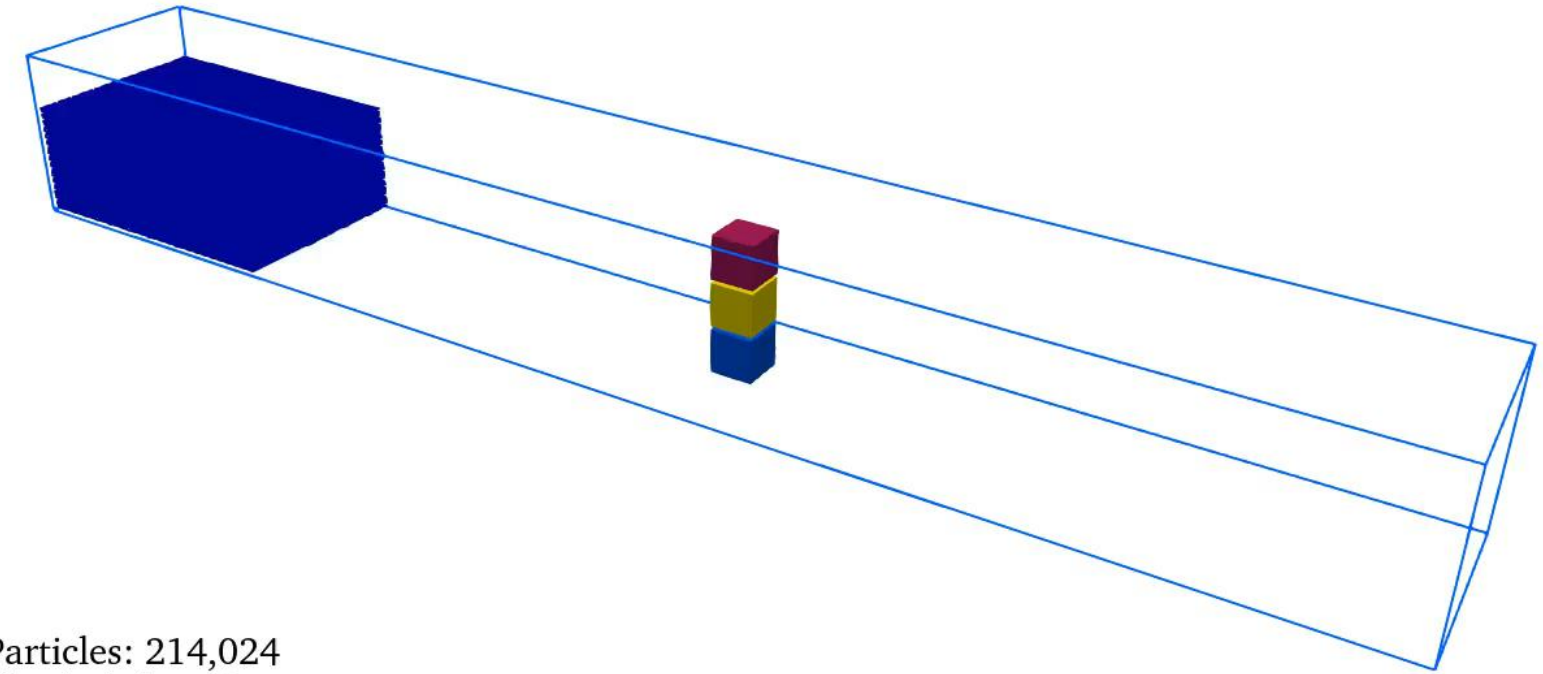
Validation







DamBreak3Cubes



Particles: 214,024
Physical time: 2 s
Runtime (Tesla K20c): 889 s

Time: 0.00 s

PROJECT CHRONO LIBRARY

COLLISIONS

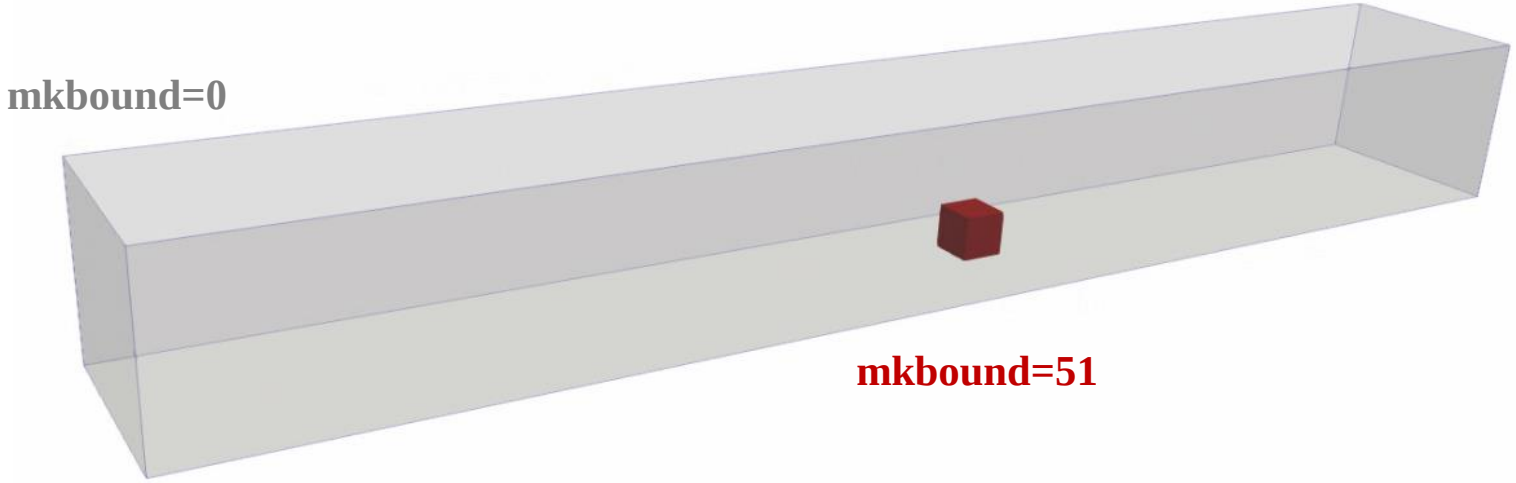
XML FILE

mkbound=51 is a floating object made of “pvc”
mkbound=0 is a fixed tank made of “steel”

```
<floatings>
    <floating mkbound="51" property="pvc" >
        <massbody value="2.7" />
    </floating>
</floatings>

<properties>
    <propertyfile file="Floating_Materials.xml" path="materials" />
    <links>
        <link mkbound="0" property="steel" />
    </links>
</properties>
```

```
<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>
```



PROJECT CHRONO LIBRARY

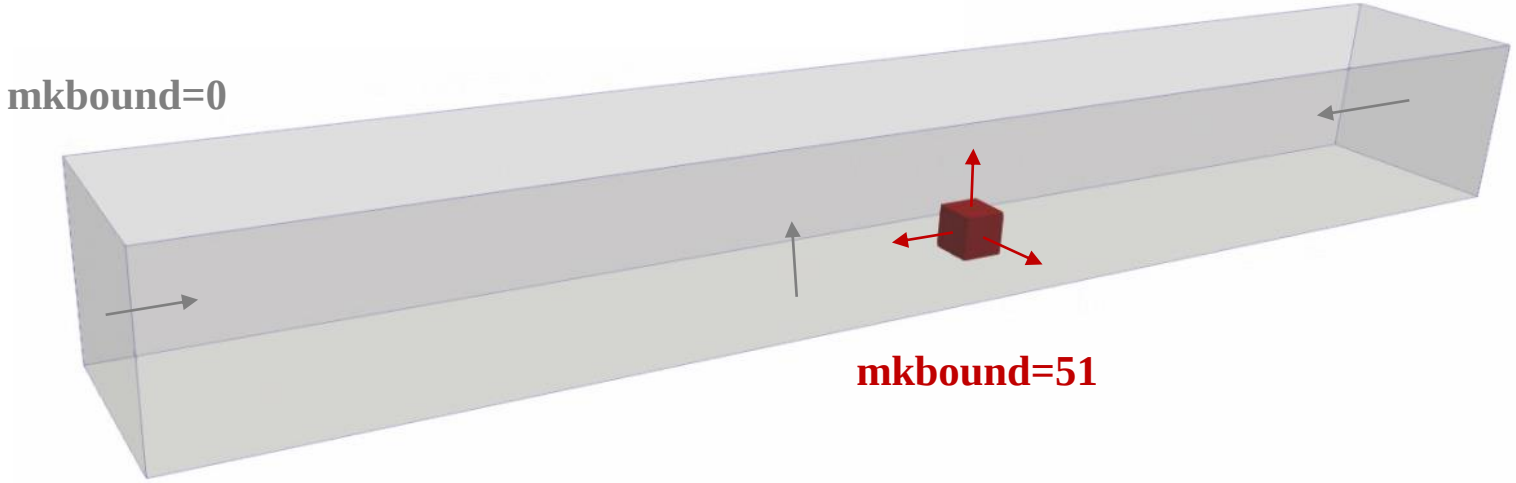
COLLISIONS

XML FILE

activate:true To activate the functionality to solve collisions
distancedp:0.5 Allowed collision overlap according to “dp” (0.5*dp)
ompthreads:1 Number of threads by host for parallel execution. 0:Multi-Core, 1:Single-Core
contactmethod:0 Contact method type. 0:NSC (Non Smooth Contacts), 1:SMC (SMooth Contacts)

```
<special>
    <chrono>
        <savedata value="0.01" />
        <schemescale value="1" />
        <collision activate="true">
            <distancedp value="0.5" />
            <ompthreads value="1" />
            <contactmethod value="0" />
        </collision>
        <bodyfloating id="cube" mkbound="51" modelfile="AutoActual" />
        <bodyfixed id="tank" mkbound="0" modelfile="AutoActual" modelnormal="invert" />
    </chrono>
</special>
```

Creates the objects in **Case_out/chrono_objs**:
cube_mkb0051.obj
tank_mkb0000.obj

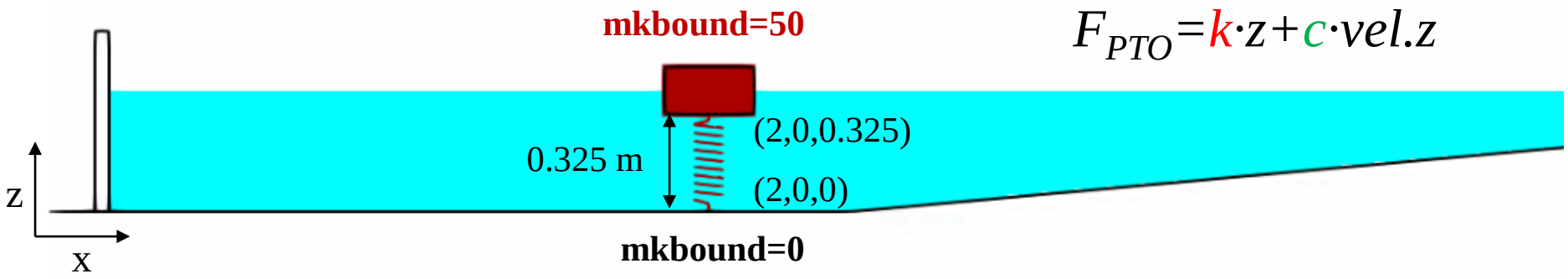


PROJECT CHRONO LIBRARY

LINEAR SPRING

$$F_{PTO}=k\cdot z+c\cdot vel.z$$

XML FILE




```xml
<chrono>
    <savedata value="0.05" />
    <schemescale value="1" />
    <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" />
        <point_fb2 x="2.0" y="0.0" z="0.325" />
        <stiffness value="100.0" />
        <damping value="500.0" />
        <rest_length value="0.325" />
        <savevtk>
            <nside value="16" />
            <radius value="5.0" />
            <length value="3.0" />
        </savevtk>
    </link_linearspring>
</chrono>
```

A spring will connect two bodies:
- Body 1: “Bottom” (mkbound=0)
- Body 2: “Floater” (mkbound=50)

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]

$$F_{PTO}=k\cdot z+c\cdot vel.z$$

VTK for visualisation: *Case_out/SpringVtk/Chrono_Springs_xxxx.vtk*
nside: number of sections for each revolution
radius: spring radius
length: length of each revolution

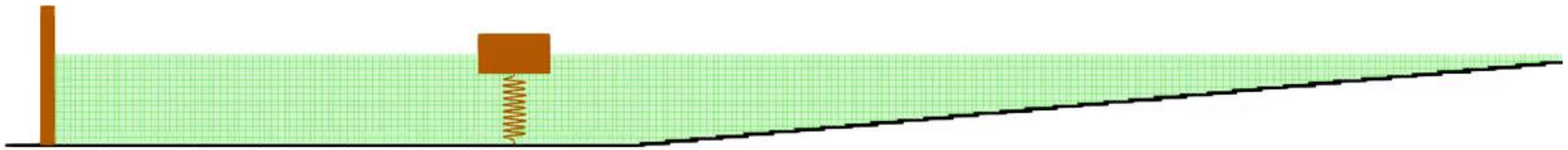
PROJECT CHRONO LIBRARY

LINEAR SPRING

XML FILE

Examples included in DualSPHysics_v5.0

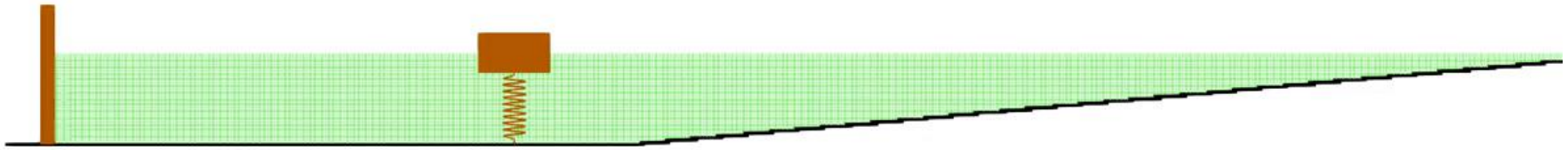
CasePointAbsorberSpring



$$F_{PTO}=k\cdot z+c\cdot vel.z$$

Runtime (GeForce RTX 2080): 497 s

CasePointAbsorberCoulomb



$$F_{PTO}=-sign(vel.z)\cdot F_b$$

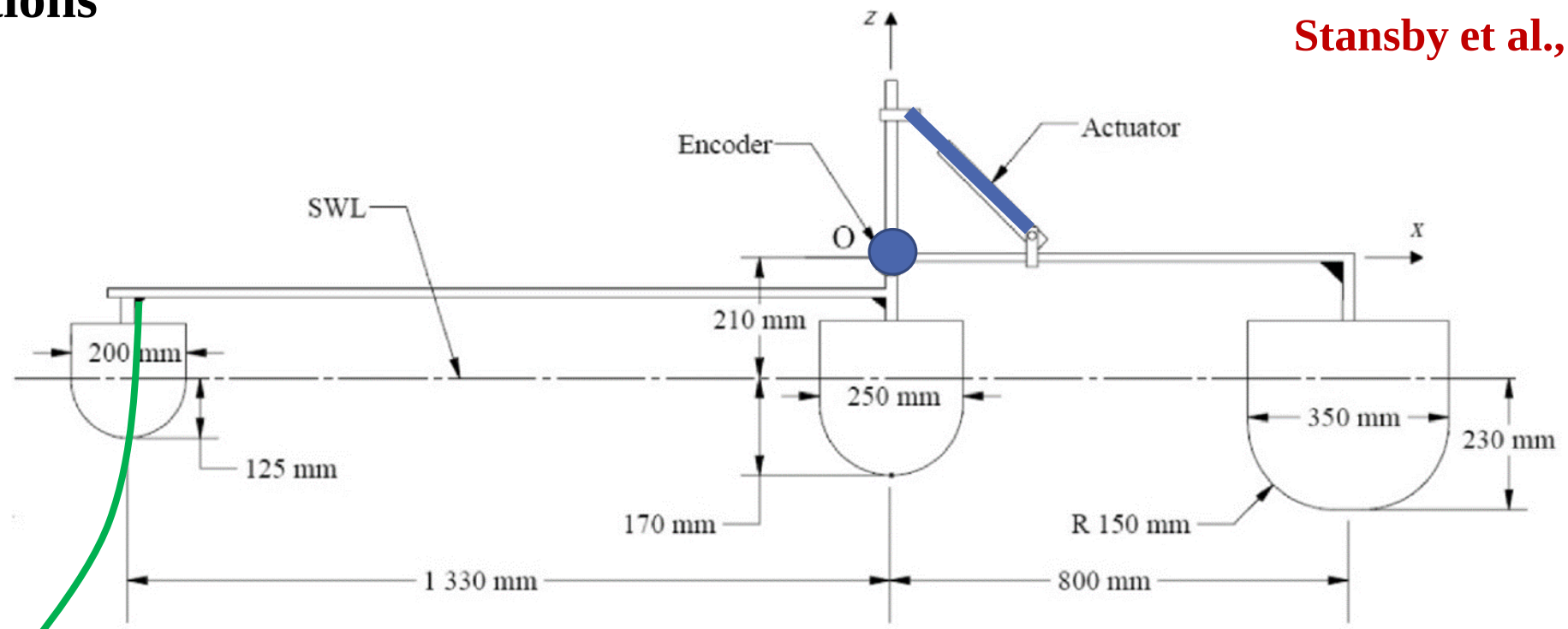
Particles: 19,897
Physical time: 10 s
Runtime (GeForce RTX 2080): 520 s

Time: 0.00 s

COUPLING DUALSPHYSICS+CHRONO

Applications

Stansby et al., 2017 APOR

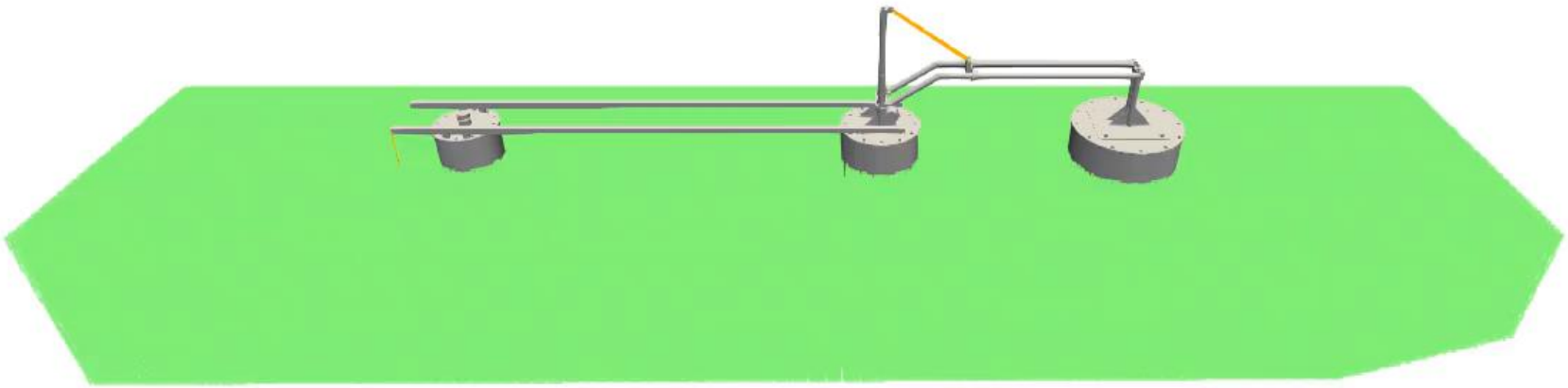


Interaction waves-structure	**SPH**
Mechanical constraint “hinge”	**CHRONO**
Pneumatic actuator or damper	**CHRONO**
Mooring line	**MOORDYN**

COUPLING DUALSPHYSICS+CHRONO

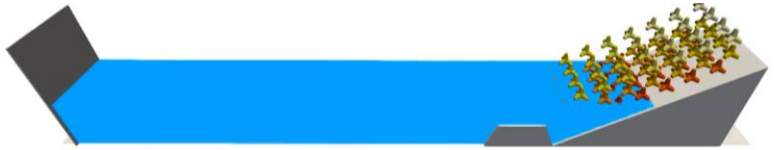
Applications

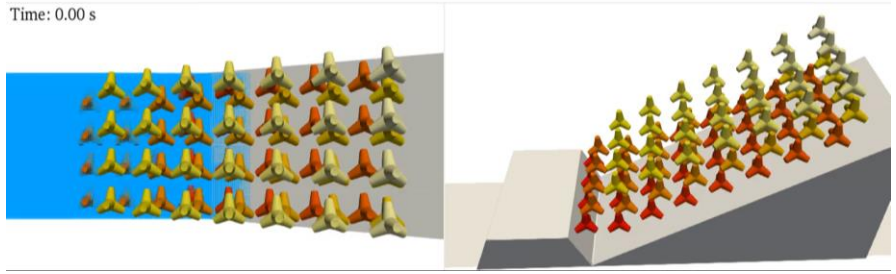
Stansby et al., 2017 APOR



COUPLING DUALSPHYSICS+CHRONO

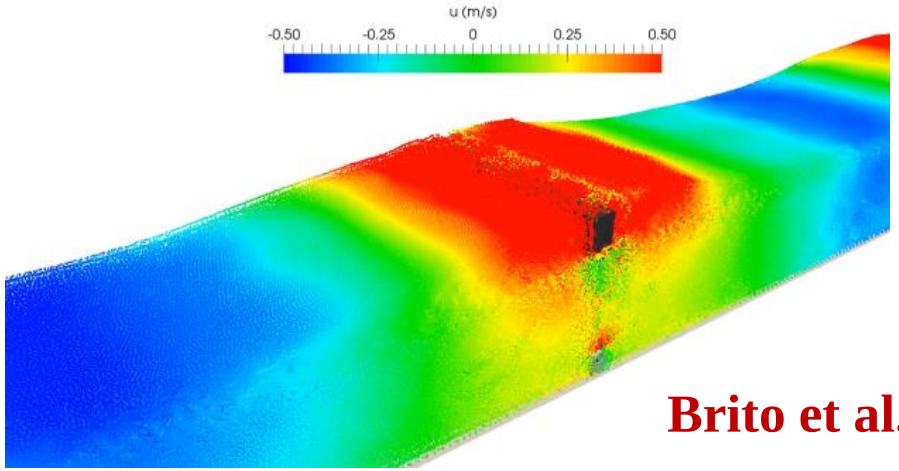
A) Design of coastal structures with moving units





Jun Mitsui (tetrapods)
Takashi Yamamoto (HC- and ST-blocks)

B) Design of Wave Energy Converters (WEC)



Brito et al., 2020 RENE

Bonaventura Tagliafierro (point absorber)
Daniel Clemente (E-motions)
Nicolas Hanousek (CarBine)
Francisco Bernardo (WEHC)
Nicolas Quartier (OWC)
Giacomo Viccione (Pelton turbine)

OUTLINE

Motivation

Coupling of DualSPHysics with MoorDyn

- MoorDyn library
- Formulation/Functionalities/Implementation
- Validation & examples

Coupling of DualSPHysics with Project Chrono

- Project Chrono library
- Formulation/Functionalities/Implementation
- Validation & examples

Main developers

Work in progress

MAIN DEVELOPERS

Instituto Superior Tecnico, Lisbon, Portugal

Dr Ricardo Canelas

- Coupling with DEM
- Coupling with Project Chrono

Dr Moisés Brito

- Simulation of OWSC

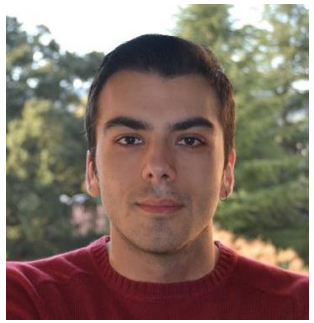
Universidade de Vigo, Spain

Dr José Manuel Domínguez Alonso

Dr Orlando García Feal

Iván Martínez Estévez

- Coupling with MoorDyn+
- New coupling with Project Chrono



OUTLINE

Motivation

Coupling of DualSPHysics with MoorDyn

- MoorDyn library
- Formulation/Functionalities/Implementation
- Validation & examples

Coupling of DualSPHysics with Project Chrono

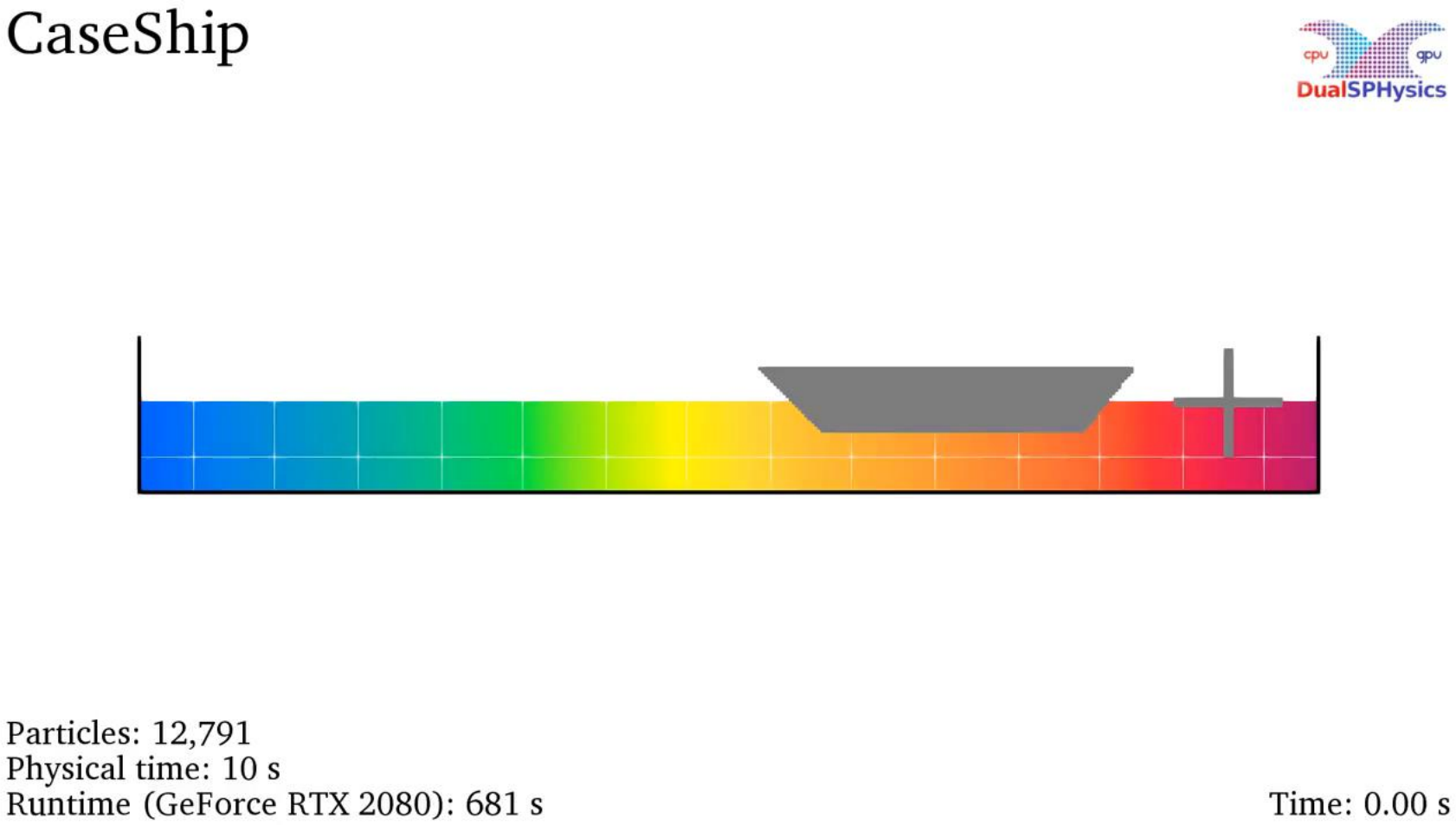
- Project Chrono library
- Formulation/Functionalities/Implementation
- Validation & examples

Main developers

Work in progress

WORK IN PROGRESS

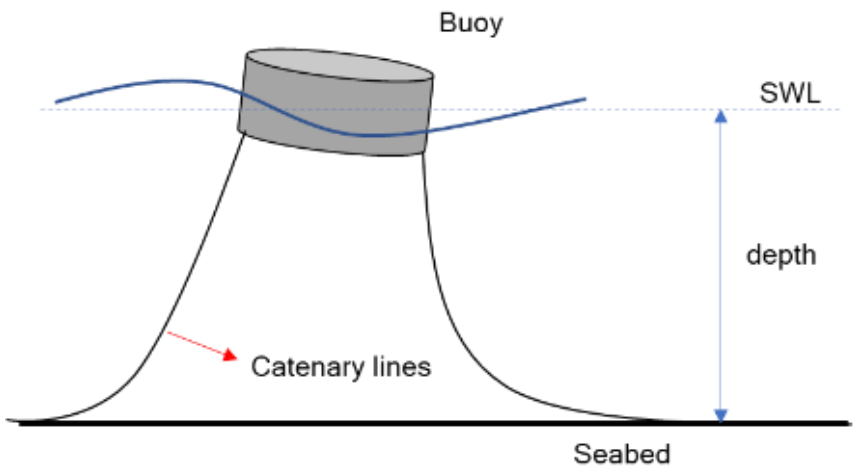
External motion can be imposed to floaters in combination with CHRONO

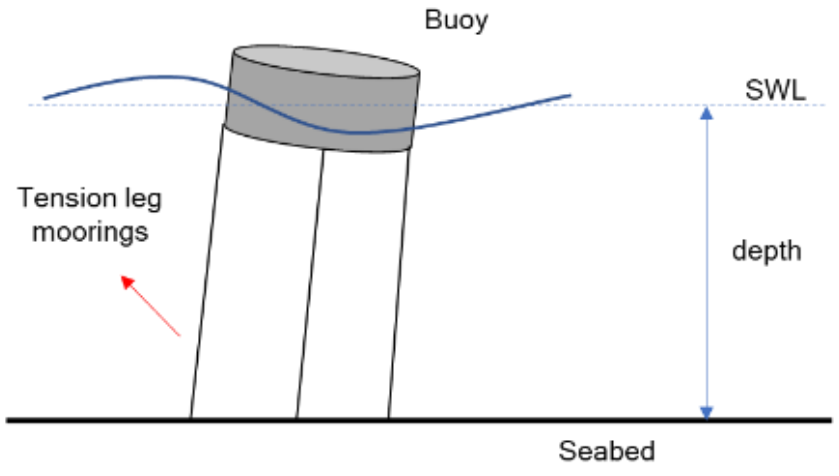


WORK IN PROGRESS

External motion can be imposed to floaters in combination with CHRONO

Different type of moorings lines: catenary and tension moorings





WORK IN PROGRESS

External motion can be imposed to floaters in combination with CHRONO

Different type of moorings lines: catenary and tension moorings

New version of Project Chrono v6 (current version is coupled with v4)

WORK IN PROGRESS

External motion can be imposed to floaters in combination with CHRONO

Different type of moorings lines: catenary and tension moorings

New version of Project Chrono v6 (current version is coupled with v4)

Coupling with FEA module to simulate flexible structures

Joe El Rahi (FEA coupling)

MORE INFORMATION

DualSPHysics_v5.0\doc\guides\XML_GUIDE_MOORDYN.pdf

DualSPHysics_v5.0\examples\moordyn

01_MooredBox
02_WavesMooring2D
03_WavesMooring3D
04_ConnectedBoxes
05_SteppedTank

DualSPHysics_v5.0\doc\guides\XML_GUIDE_CHRONO.pdf

DualSPHysics_v5.0\examples\chrono

01_Pendulum
02_Spring
03_FlexibleGate
04_Pelamis
05_OWSC
06_ZipLine
07_DamBreakCubes
08_WaterMill
09_Turbine
10_PointAbsorber
11_CurrentWheelPulley
12_ExternalVelocity

Coupling with MoorDyn library and with Project Chrono

ALEJANDRO J. C. CRESPO
IVÁN MARTÍNEZ-ESTÉVEZ
UNIVERSIDADE DE VIGO