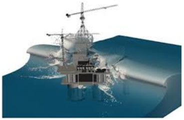
Open Boundary Conditions: New Developments, Capabilities and Practical Examples

Dr. Angelo Tafuni
New Jersey Institute of Technology

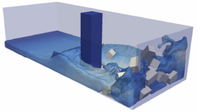
Outline

Available Boundary Conditions in DualSPHysics

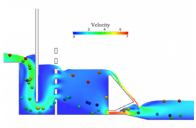
- Dynamic Boundary Conditions



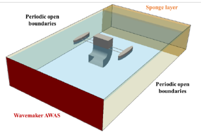
- DEM Coupling



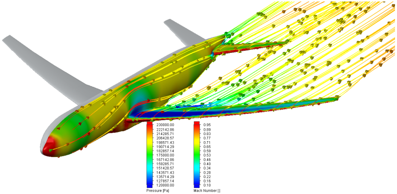
- Periodic Boundary Conditions



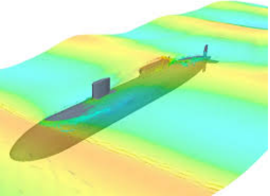
- Passive and Active Wave Absorption



Need for Open Boundary Conditions



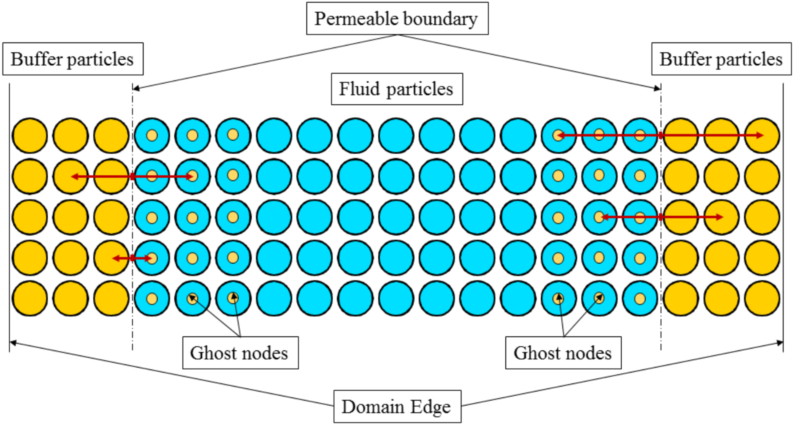
Flow past an airplane body



Free-surface flow past a submarine

- To simulate flow over partially and fully immersed bodies
- To reduce the extent of the computational domain in confined and external flow
- To adequately simulate the transport of fluid properties between the fluid domain and the boundaries

Approach: Buffer Areas for Inlets and Outlets



Sketch of the adopted I/O model (Tafuni et al. 2017).

Approach: Mathematical Formulation

$$\int f(\mathbf{x}) W_i(\mathbf{x})\,\mathrm{d}\mathbf{x} = f_i \int W_i(\mathbf{x})\,\mathrm{d}\mathbf{x} + f_{i,\beta} \int (\mathbf{x}-\mathbf{x}_\mathrm{i}) W_i(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

$$\int f(\mathbf{x}) W_{i,\beta}(\mathbf{x})\,\mathrm{d}\mathbf{x} = f_i \int W_{i,\beta}(\mathbf{x})\,\mathrm{d}\mathbf{x} + f_{i,\beta} \int (\mathbf{x}-\mathbf{x}_\mathrm{i}) W_{i,\beta}(\mathbf{x})\,\mathrm{d}\mathbf{x}$$

$$f_i = \frac{\begin{vmatrix} \sum_j f_j W_{ij} \Delta V_j & \sum_j (\mathbf{x}_\mathrm{j}-\mathbf{x}_\mathrm{i}) W_{ij} \Delta V_j \\ \sum_j f_j W_{ij,\beta} \Delta V_j & \sum_j (\mathbf{x}_\mathrm{j}-\mathbf{x}_\mathrm{i}) W_{ij,\beta} \Delta V_j \end{vmatrix}}{\begin{vmatrix} \sum_j f(\mathbf{x}) W_{ij} \Delta V_j & \sum_j (\mathbf{x}_\mathrm{j}-\mathbf{x}_\mathrm{i}) W_{ij} \Delta V_j \\ \sum_j f(\mathbf{x}) W_{ij,\beta} \Delta V_j & \sum_j (\mathbf{x}_\mathrm{j}-\mathbf{x}_\mathrm{i}) W_{ij,\beta} \Delta V_j \end{vmatrix}}$$

$$f_{i,\beta} = \frac{\begin{vmatrix} \sum_j W_{ij} \Delta V_j & \sum_j f_j W_{ij} \Delta V_j \\ \sum_j W_{ij,\beta} \Delta V_j & \sum_j f_j W_{ij,\beta} \Delta V_j \end{vmatrix}}{\begin{vmatrix} \sum_j W_{ij} \Delta V_j & \sum_j (\mathbf{x}_\mathrm{j}-\mathbf{x}_\mathrm{i}) W_{ij} \Delta V_j \\ \sum_j W_{ij,\beta} \Delta V_j & \sum_j (\mathbf{x}_\mathrm{j}-\mathbf{x}_\mathrm{i}) W_{ij,\beta} \Delta V_j \end{vmatrix}}$$

Yields a system of 4 equations in 4 unknowns (f_i and its gradient components) [Liu and Liu, 2006]

Approach: Mathematical Formulation

The 4x4 linear system $L_a\mathbf{f} = R_a$ is solved for all or some of the below

$$\mathbf{f} = [\rho_a, \rho_{a,\tilde{x}}, \rho_{a,\tilde{y}}, \rho_{a,\tilde{z}}]^T$$

$$\mathbf{f} = [u_a, u_{a,\tilde{x}}, u_{a,\tilde{y}}, u_{a,\tilde{z}}]^T$$

$$\mathbf{f} = [v_a, v_{a,\tilde{x}}, v_{a,\tilde{y}}, v_{a,\tilde{z}}]^T$$

$$\mathbf{f} = [w_a, w_{a,\tilde{x}}, w_{a,\tilde{y}}, w_{a,\tilde{z}}]^T$$

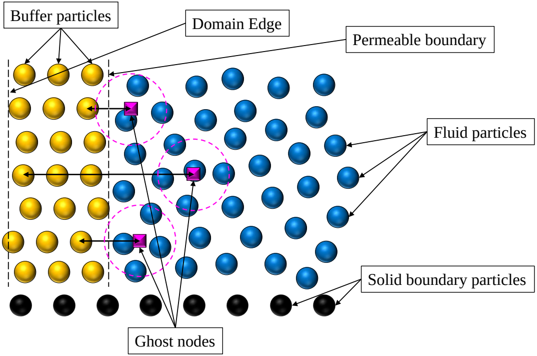
$$L_a = \sum_b V_b \begin{bmatrix} W_{ab} \\ \nabla_{ax} W_{ab} \\ \nabla_{ay} W_{ab} \\ \nabla_{az} W_{ab} \end{bmatrix} \begin{bmatrix} 1, & x_{ab}, & y_{ab}, & z_{ab} \end{bmatrix} \qquad \mathcal{R}_a = \sum_b m_b \begin{bmatrix} W_{ab} \\ \nabla_{ax} W_{ab} \\ \nabla_{ay} W_{ab} \\ \nabla_{az} W_{ab} \end{bmatrix}$$

Cost of the boundary treatment:

- two additional loops over buffer particles, calculations strictly in double precision for matrix inversion.
- Bad matrix? Automatically updates buffer particles with zero-th order accuracy.

Outline

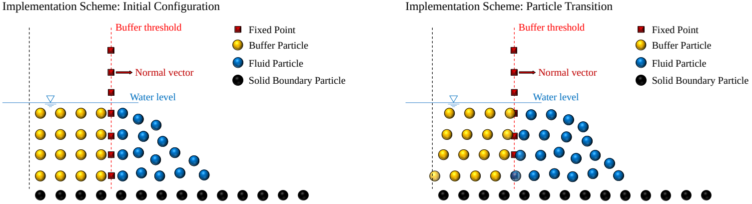
Open Boundary Algorithm: Features



Sketch of the implemented I/O model (Tafuni et al. 2018).

Same C++ class of particles for buffer areas $->$ Dual inlet/outlet behavior possible

Open Boundary Algorithm: Features

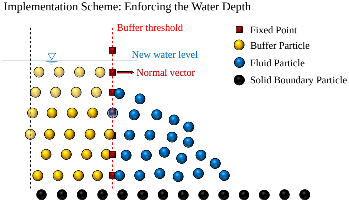


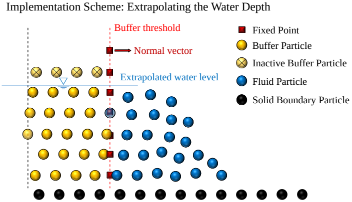
Sketch of the proposed boundary algorithm in DualSPHysics.

Buffer particles are created by extruding from a threshold curve or surface made of fixed points, along the normal direction to this curve/surface. Based on their movement, different scenarios can occurr:

1. When a buffer particle crosses the fixed threshold, it is converted to fluid and a new particle is initialized in the buffer at a distance equal to the buffer width
2. When a fluid particle crosses the threshold, it is converted to a buffer particle
3. When a buffer particle crosses the domain edge, it is automatically eliminated and its array values are purged

Open Boundary Algorithm: Features



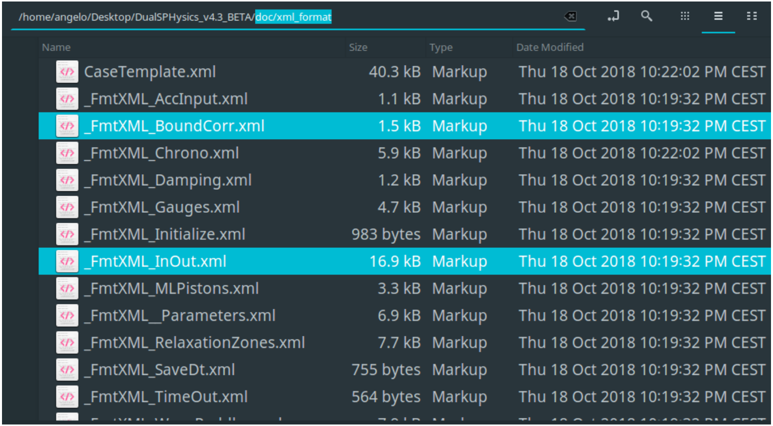


Sketch of the proposed boundary algorithm in DualSPHysics.

When the velocity, density, and/or free-surface BCs are assigned by the user:

- Fluid properties can be frozen to constant values at the boundary, or...
- ...space– and time–dependent profiles can be easily assigned at the boundary, or...
- ...buffer variables can be assigned from external file/software

Templates



The folder has a collection of XML templates for GenCase, each describing a specific feature of the code. Highlighted are the two of interest for OBC.

Templates: Inlet/Outlet Creation

_FmtXML_InOut.xml

```
<special>
    <!-- <inout reuseids="false" resizetime="0.5">
        <useboxlimit value="true" comment="In/out process is only applied to InOut zones delimited
by BoxLimit (default=true)">
            <freecentre x="2" y="0" z="0" comment="Centre of zone where InOut is not applied
(default=centre of simulation domain)" units_comment="m" />
        </useboxlimit>
        <userefilling value="true" comment="Use advanced refilling algorithm but slower. It is
necessary when outflow becomes inflow (default=false)" />
        <determlimit value="1e+3" comment="Use 1e-3 for first_order or 1e+3 for zeroth_order
(default=1e+3)" />
        <extrapolatemode value="1" comment="Calculation mode for rhop and velocity extrapolation
from ghost nodes 1:fast-single, 2:single, 3:double (default=1)" /> -->
</special>
```

All properties of buffer areas are defined in the tag <*special*>

Templates: Inlet/Outlet Creation

_FmtXML_InOut.xml

```
<special>
    <inout reuseids="false" resizetime="0.5">
        <useboxlimit value="true" comment="In/out process is only applied to InOut zones delimited by BoxLimit (default=true)">
            <freecentre x="2" y="0" z="0" comment="Centre of zone where InOut is not applied (default=centre of simulation domain)" units_comment="m" />
        </useboxlimit>
        <userefilling value="true" comment="Use advanced refilling algorithm but slower. It is necessary when outflow becomes inflow (default=false)" />
        <determlimit value="1e+3" comment="Use 1e-3 for first_order or 1e+3 for zeroth_order (default=1e+3)" />
        <extrapolatemode value="1" comment="Calculation mode for rhop and velocity extrapolation from ghost nodes 1:fast-single, 2:single, 3:double (default=1)" />
<!--  [MORE HERE ...]  -->
    </inout>
</special>
```

Options available before creating the buffer geometry:

- *reuseid*: Obsolete, will be removed.
- *resizetime*: Frequency of memory allocation. Performance parameter.
- *useboxlimit*: Flags the areas where I/O algorithm is activated
- *userefilling*: Allows dual inlet/outlet behavior in the buffer area
- *determlimit*: Determines the accuracy of the transport to/from ghost nodes
- *extrapolatemode*: Double precision is enforced in different portions of the code

Templates: Inlet/Outlet Creation

_FmtXML_InOut.xml

```
<special>
    <!-- [MORE HERE] -->
        <inoutzone>
            <convertfluid value="true" comment="Converts fluid in inlet/outlet area
(default=true)" />
            <layers value="8" comment="Number of inlet/outlet particle layers" />
    <!-- [GEOMETRY OF THE BUFFER IS CREATED HERE] -->
        <imposevelocity mode="0-or-1-or-2-or-3" comment="Imposed velocity 0:fixed value, 1:variable
value, 2:Extrapolated velocity, 3:Interpolated velocity (default=0)">
    <!-- [MORE HERE] -->
            </imposevelocity>
            <imposerhop mode="0-or-1-or-2" comment="Outlet rhop 0:Imposed fixed value,
1:Hydrostatic, 2:Extrapolated from ghost nodes (default=0)">
    <!-- [MORE HERE] -->
            </imposerhop>
    <imposezsurf mode="0-or-1-or-2" comment="Inlet Z-surface 0:Imposed fixed value, 1:Imposed
variable value, 2:Calculated from fluid domain (default=0)">
    <!-- [MORE HERE] -->
    </imposezsurf>
        </inoutzone>
    </inout>
</special>
```

A new buffer is created via the tag <*inoutzone*>, then the user has two options before choosing the geometry:

- *layers*: Number of particle layers forming the buffer from the threshold
- *convertfluid*: Fluid going into the buffer area is converted into buffer particles

Templates: Inlet/Outlet Creation

_FmtXML_InOut.xml

```
<special>
    <!-- [MORE HERE] -->
        <inoutzone>
    <!-- [GEOMETRY OF THE BUFFER IS CREATED HERE] -->
        <imposevelocity mode="0-or-1-or-2-or-3" comment="Imposed velocity 0:fixed value, 1:variable
value, 2:Extrapolated velocity, 3:Interpolated velocity (default=0)">
    <!-- [MORE HERE] -->
            </imposevelocity>
            <imposerhop mode="0-or-1-or-2" comment="Outlet rhop 0:Imposed fixed value,
1:Hydrostatic, 2:Extrapolated from ghost nodes (default=0)">
    <!-- [MORE HERE] -->
            </imposerhop>
    <imposezsurf mode="0-or-1-or-2" comment="Inlet Z-surface 0:Imposed fixed value, 1xml:Imposed
variable value, 2:Calculated from fluid domain (default=0)">
    <!-- [MORE HERE] -->
    </imposezsurf>
        </inoutzone>
    </inout>
</special>
```

Geometry is created (we will see in the examples) and then, for every buffer area, the user has to handle three flow properties:

- *imposerhop*: Density of buffer particles (three options)
- *imposevelocity*: Velocity of buffer particles (four options)
- *imposezsurf*: Water depth within the buffer region (three options)

Templates: BoundCorr for Dynamic Boundary Particles

_FmtXML_BoundCorr.xml

```
<special>
    <boundcorr>
        <determlimit value="1e-3" comment="Use 1e-3 for first_order or 1e+3 for zeroth_order (default=1e-3)" />
        <extrapolatemode value="1" comment="Calculation mode for rhop extrapolation from ghost nodes 1:fast-single, 2:single, 3:double (default=1)" />
        <!-- limitpoint and direction are explicitly defined. -->
        <mkzone mkbound="1">
            <limitpoint x="-1.1" y="0" z="0.1" />
            <direction x="1" y="0" z="0" comment="Direction to fluid" />
        </mkzone>
        <!-- limitpoint and direction are calculated automatically (for boxes). -->
        <mkzone mkbound="2">
            <autoconfig direction="right" comment="Direction (top/bottom/left/right/front/back) to fluid using automatic configuration (only for parallel boxes)" />
        </mkzone>
        <!-- direction is defined and limitpoint is calculated automatically (for sloping surfaces). -->
        <mkzone mkbound="3">
            <autolimitpoint dpfactor="0.5" comment="Point is calculated starting from bound particles at distance dp*dpfactor" />
            <direction x="1" y="0" z="0" comment="Direction to fluid" />
        </mkzone>
    </boundcorr>
</special>
```

Three options for density extrapolation of DBP:

- *limitpoint*: Manually define fluid/boundary threshold
- *autoconfig*: Automatically define fluid/boundary threshold via direction
- *autolimitpoint*: Automatically define fluid/boundary threshold via dp factor

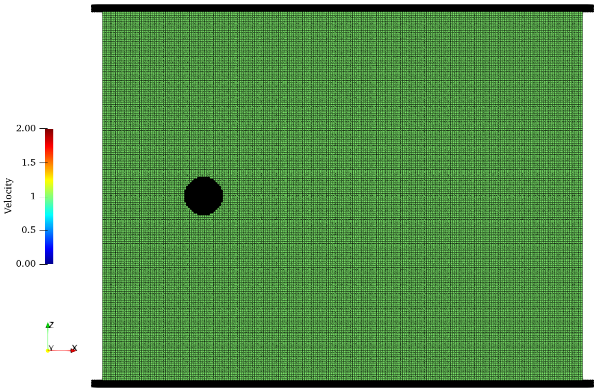
Outline

2-D Flow Past a Cylinder

Name	Size	Type	Date Modified
1_FlowCylinder	13 items	Folder	Fri 19 Oct 2018 12:31:07 PM CEST
2_OpenChannel	7 items	Folder	Fri 19 Oct 2018 12:42:25 PM CEST
3_ReverseFlow	7 items	Folder	Fri 19 Oct 2018 12:44:21 PM CEST
4_Waves2D	8 items	Folder	Fri 19 Oct 2018 12:48:50 PM CEST
5_ShapesInlet3D	7 items	Folder	Fri 19 Oct 2018 01:04:08 PM CEST
6_Box4Inlet3D	7 items	Folder	Fri 19 Oct 2018 01:14:42 PM CEST
7_CurrentHull	8 items	Folder	Fri 19 Oct 2018 01:22:09 PM CEST

Folders of test cases in "examples/inletoutlet".

2-D Flow Past a Cylinder



$$\mathrm{Re} = \frac{U_\infty D}{\nu}$$

$$U_\infty = 1;$$
$$D = 0.1;$$

$$\nu_1 = 0.005$$
$$\downarrow$$
$$\mathrm{Re}_1 = 20$$

$$\nu_2 = 0.001$$
$$\downarrow$$
$$\mathrm{Re}_2 = 100$$

Flow past a cylinder for two Reynolds numbers, i.e. 20 and 100.

2-D Flow Past a Cylinder

```
<!-- INITIALIZATION OF FLUID VELOCITY -->
<initials>
    <velocity mkfluid="0">
        <direction x="1" y="0" z="0" />
        <velocity v="1.0" comment="Uniform profile velocity" units_comment="m/s" />
    </velocity>
</initials>
<!-- MOTION OF UPPER AND LOWER BOUNDARY -->
<motion>
    <objreal ref="1">
        <begin mov="1" start="0" finish="100" />
        <mvrect id="1" duration="100">
            <velini x="1" y="0" z="0" units_comment="m/s" />
            <vel x="1" y="0" z="0" units_comment="m/s" />
        </mvrect>
    </objreal>
    <objreal ref="2">
        <begin mov="1" start="0" finish="100" />
        <mvrect id="1" duration="100">
            <velini x="1" y="0" z="0" units_comment="m/s" />
            <vel x="1" y="0" z="0" units_comment="m/s" />
        </mvrect>
    </objreal>
</motion>
```

Periodicity in X must be specified

```
<parameter key="XPeriodicIncZ" value="0" comment="Increase of Z with periodic BC" />
```

2-D Flow Past a Cylinder

We then introduce the inlet...

```
<special> <!-- HERE WE DEFINE THE INLET AND OUTLET -->
    <inout reuseids="false" resizetime="0.5">
        <userefilling value="true" comment="Use advanced refilling algorithm but slower. It is necessary when
outflow becomes inflow (default=false)" />
        <determlimit value="1e+3" comment="Use 1e-3 for first_order or 1e+3 for zeroth_order (default=1e+3)" />
        <!-- INLET -->
        <inoutzone>
            <layers value="4" comment="Number of inlet/outlet particle layers" />
            <zone2d comment="Input zone for 2D simulations">s
                <particles mkfluid="1" direction="right" />
            </zone2d>
            <imposevelocity mode="0" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated
value (default=0)">
                <velocity v="1.0" comment="Uniform velocity" units_comment="m/s" />
            </imposevelocity>
            <imposerhop mode="2" comment="Outlet rhop 0:Imposed fixed value, 1:Hydrostatic, 2:Extrapolated
from ghost nodes (default=0)" />
            <rhop value="1000" comment="Characteristic inlet rhop (default=1000)" units_comment="kg/m^2" />
            <imposezsurf mode="2" comment="Inlet Z-surface 0:Imposed fixed value, 1:Imposed variable value,
2:Calculated from fluid domain (default=0)">&gt;
                <zbottom value="-1.5" comment="Bottom level of water (only for Hydrostatic option)"
units_comment="m" />
                <zsurf value="1.5" comment="Characteristic inlet Z-surface (only for Hydrostatic option)"
units_comment="m" />
            </imposezsurf>
        </inoutzone>
```

2-D Flow Past a Cylinder

... and the outlet

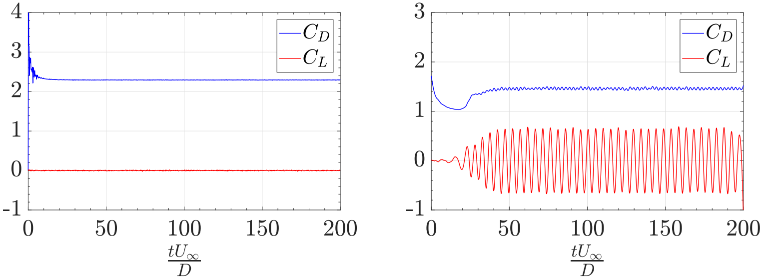
```
<!-- OUTLET -->
<inoutzone>
    <layers value="4" comment="Number of inlet/outlet particle layers" />
    <zone2d comment="Input zone for 2-D simulations">
        <particles mkfluid="2" direction="left" />
    </zone2d>
    <imposevelocity mode="2" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated
value (default=0)"/>
    <imposerhop mode="2" comment="Outlet rhop 0:Imposed fixed value, 1:Hydrostatic, 2:Extrapolated
from ghost nodes (default=0)"/>
    <imposezsurf mode="2" comment="Inlet Z-surface 0:Imposed fixed value, 1:Imposed variable value,
2:Calculated from fluid domain (default=0)">&gt;
        <zbottom value="-1.5" comment="Bottom level of water (only for Hydrostatic option)"
units_comment="m" />
        <zsurf value="1.5" comment="Characteristic inlet Z-surface (only for Hydrostatic option)"
units_comment="m" />
    </imposezsurf>
</inoutzone>
</inout>
```

2-D Flow Past a Cylinder: Use of BoundCorr

The OBC algorithm is used to extrapolate the density of lower and upper boundaries:

```
            <!-- THE GHOST NODES ARE USED FOR THE DENSITY EXTRAPOLATION OF DBP FORMING THE UPPER AND LOWER BOUNDARIES
-->
            <boundcorr>
                <determlimit value="1e+3" comment="Use 1e-3 for first_order or 1e+3 for zeroth_order" />
                <mkzone mkbound="1">
                    <limitpoint x="0" y="0" z="-0.995" /> <!-- THIS IDENTIFIES THE SEPARATING LINE BETWEEN THE
BOUNDARY AND FLUID, AT dp/2 DISTANCE FROM BOTH -->
                    <direction x="0" y="0" z="1" comment="Direction to fluid" /> <!-- THIS IDENTIFIES THE DIRECTION
ALONG WHICH GHOST NODES WILL BE PLACED -->
                </mkzone>
            </boundcorr>
            <boundcorr>
                <determlimit value="1e+3" comment="Use 1e-3 for first_order or 1e+3 for zeroth_order" />
                <mkzone mkbound="2">
                    <limitpoint x="0" y="0" z="0.995" /> <!-- THIS IDENTIFIES THE SEPARATING LINE BETWEEN THE BOUNDARY
AND FLUID, AT dp/2 DISTANCE FROM BOTH -->
                    <direction x="0" y="0" z="-1" comment="Direction to fluid" /> <!-- THIS IDENTIFIES THE DIRECTION
ALONG WHICH GHOST NODES WILL BE PLACED -->
                </mkzone>
            </boundcorr>
        </special>
```

2-D Flow Past a Cylinder



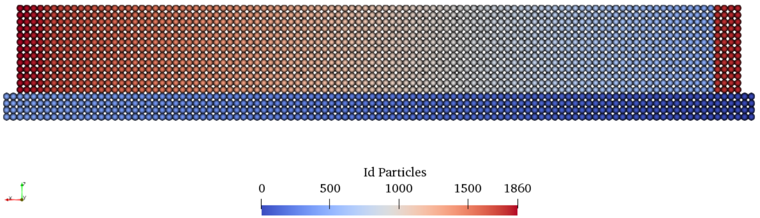
You can use ComputeForces to extrapolate the fluid force on the cylinder, and retrieve drag and lift coefficients, respectively C_D and C_L. Here both are shown for Re = 20 (left) and Re = 200 (right) at decent resolution (Tafuni et al. 2018).

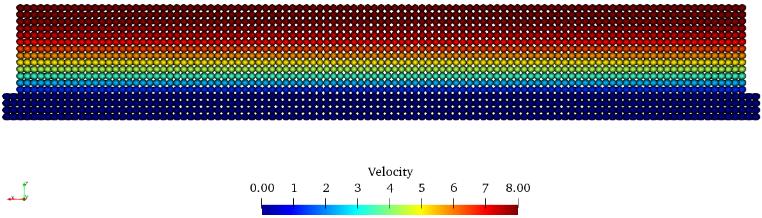
2-D Open-Channel Flow

Name	Size	Type	Date Modified
1_FlowCylinder	13 items	Folder	Fri 19 Oct 2018 12:31:07 PM CEST
2_OpenChannel	7 items	Folder	Fri 19 Oct 2018 12:42:25 PM CEST
3_ReverseFlow	7 items	Folder	Fri 19 Oct 2018 12:44:21 PM CEST
4_Waves2D	8 items	Folder	Fri 19 Oct 2018 12:48:50 PM CEST
5_ShapesInlet3D	7 items	Folder	Fri 19 Oct 2018 01:04:08 PM CEST
6_Box4Inlet3D	7 items	Folder	Fri 19 Oct 2018 01:14:42 PM CEST
7_CurrentHull	8 items	Folder	Fri 19 Oct 2018 01:22:09 PM CEST

Folders of test cases in "examples/inletoutlet".

2-D Open-Channel Flow





2-D Open-Channel Flow

Here we initialize the fluid with a parabolic velocity ...

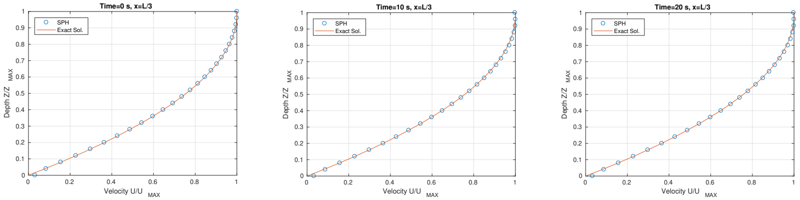
```
<special>
    <initialize>
        <fluidvelocity mkfluid="0">
            <direction x="1" y="0" z="0" />
            <velocity3 v="0" v2="6" v3="8" z="0" z2="0.5" z3="1" comment="Parabolic inlet velocity" units_comment="m/s" />
        </fluidvelocity>
    </initialize>
```

2-D Open-Channel Flow

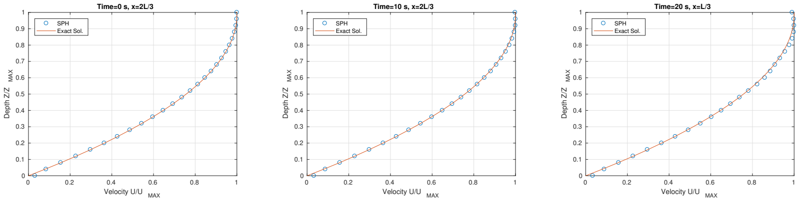
... and we assign the same velocity profile at the inlet

```
<inoutzone>
    <layers value="4" comment="Number of inlet/outlet particle layers" />
    <zone2d comment="Input zone for 2-D simulations">
        <particles mkfluid="1" direction="right" />
    </zone2d>
    <imposevelocity mode="0" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated value (default=0)">
        <velocity3 v="0" v2="6" v3="8" z="0" z2="0.5" z3="1" comment="Parabolic inlet velocity" units_comment="m/s" />
    </imposevelocity>
```

2-D Open-Channel Flow



Velocity profile at $L/3$.



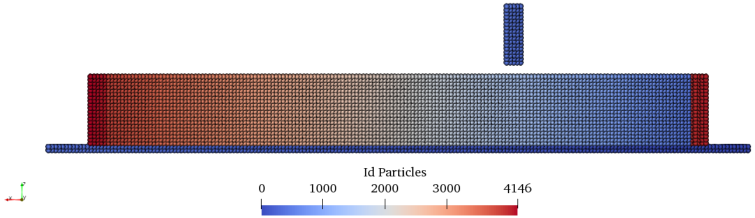
Velocity profile at $2L/3$.

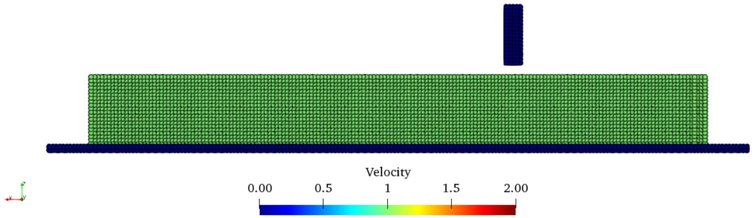
2-D Open-Channel Flow with Reversion

Name	Size	Type	Date Modified
1_FlowCylinder	13 items	Folder	Fri 19 Oct 2018 12:31:07 PM CEST
2_OpenChannel	7 items	Folder	Fri 19 Oct 2018 12:42:25 PM CEST
3_ReverseFlow	7 items	Folder	Fri 19 Oct 2018 12:44:21 PM CEST
4_Waves2D	8 items	Folder	Fri 19 Oct 2018 12:48:50 PM CEST
5_ShapesInlet3D	7 items	Folder	Fri 19 Oct 2018 01:04:08 PM CEST
6_Box4Inlet3D	7 items	Folder	Fri 19 Oct 2018 01:14:42 PM CEST
7_CurrentHull	8 items	Folder	Fri 19 Oct 2018 01:22:09 PM CEST

Folders of test cases in "examples/inletoutlet".

2-D Open-Channel Flow with Reversion





2-D Open-Channel Flow with Reversion

Here the left buffer is chosen to drive the changes in velocity sign

```
<inoutzone>
    <layers value="5" comment="Number of inlet/outlet particle layers" />
    <zone2d comment="Input zone for 2-D simulations">
        <particles mkfluid="1" direction="right" />
    </zone2d>
    <imposevelocity mode="1" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated value (default=0)">
        <velocitytimes comment="Uniform inlet velocity in time">
            <timevalue time="0" v="1" />
            <timevalue time="8" v="0" />
            <timevalue time="9" v="-1" />
            <timevalue time="15" v="0" />
            <timevalue time="15.5" v="1" />
            <timevalue time="20.5" v="1" />
        </velocitytimes>
    </imposevelocity>
```

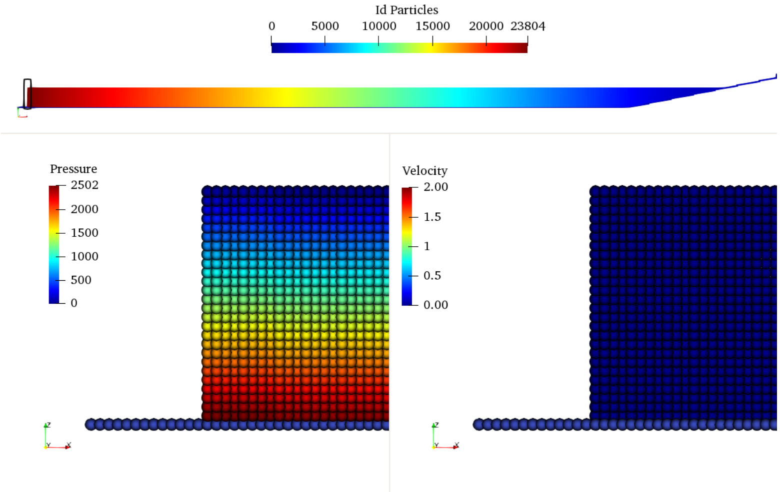
while everything else at the two boundaries is extrapolated from the fluid via the ghost nodes!

2-D Wave Generation

Name	Size	Type	Date Modified
1_FlowCylinder	13 items	Folder	Fri 19 Oct 2018 12:31:07 PM CEST
2_OpenChannel	7 items	Folder	Fri 19 Oct 2018 12:42:25 PM CEST
3_ReverseFlow	7 items	Folder	Fri 19 Oct 2018 12:44:21 PM CEST
4_Waves2D	8 items	Folder	Fri 19 Oct 2018 12:48:50 PM CEST
5_ShapesInlet3D	7 items	Folder	Fri 19 Oct 2018 01:04:08 PM CEST
6_Box4Inlet3D	7 items	Folder	Fri 19 Oct 2018 01:14:42 PM CEST
7_CurrentHull	8 items	Folder	Fri 19 Oct 2018 01:22:09 PM CEST

Folders of test cases in "examples/inletoutlet".

2-D Wave Generation



2-D Wave Generation

To achieve proper wave generation, the velocity of the particles in the left (and only) buffer is assigned via an external file with values computed from Stokes' theory:

```
<inoutzone>
    <convertfluid value="true" comment="Converts fluid in inlet/outlet area (default=true)" />
    <layers value="8" comment="Number of inlet/outlet particle layers" />
    <zone2d comment="Inlet/Outlet zone for 2-D simulations">
        <particles mkfluid="1" direction="right" />
    </zone2d>
    <imposevelocity mode="1" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated value (default=0)">
        <velocityfile file="waves.csv" comment="Uniform inlet velocity data (time;v)" />
    </imposevelocity>
```

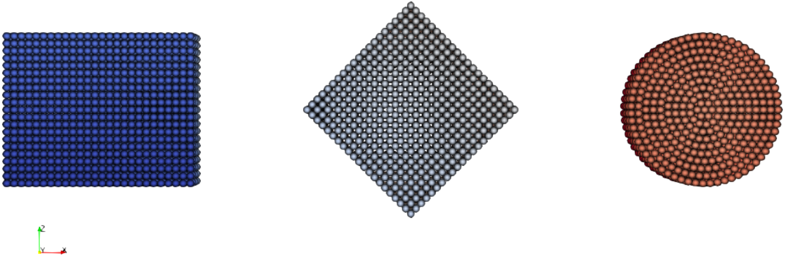
while everything else at the two boundaries is extrapolated from the fluid via the ghost nodes!

3-D Buffer Shapes

Name	Size	Type	Date Modified
1_FlowCylinder	13 items	Folder	Fri 19 Oct 2018 12:31:07 PM CEST
2_OpenChannel	7 items	Folder	Fri 19 Oct 2018 12:42:25 PM CEST
3_ReverseFlow	7 items	Folder	Fri 19 Oct 2018 12:44:21 PM CEST
4_Waves2D	8 items	Folder	Fri 19 Oct 2018 12:48:50 PM CEST
5_ShapesInlet3D	7 items	Folder	Fri 19 Oct 2018 01:04:08 PM CEST
6_Box4Inlet3D	7 items	Folder	Fri 19 Oct 2018 01:14:42 PM CEST
7_CurrentHull	8 items	Folder	Fri 19 Oct 2018 01:22:09 PM CEST

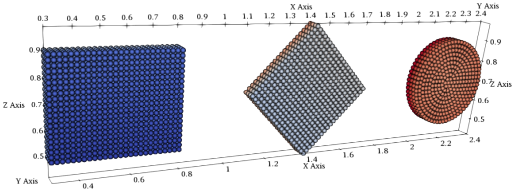
Folders of test cases in "examples/inletoutlet".

3-D Buffer Shapes



Different shape options for buffer regions.

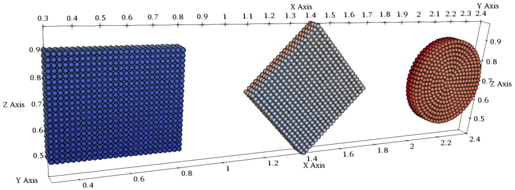
3-D Buffer Shapes: Square



Square

```
<inoutzone>
    <convertfluid value="false" comment="Converts fluid in inlet/outlet area (default=true)" />
    <layers value="4" comment="Number of inlet/outlet particle layers" />
    <zone3d comment="Input zone for 3-D simulations">
        <box>
            <point x="0.3" y="2.8" z="0.5" />
            <size x="0.5" y="0" z="0.4" />
            <direction x="0" y="-1" z="0" />
        </box>
    </zone3d>
    <imposevelocity mode="0" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated value (default=0)">
        <velocity v="2" comment="Uniform velocity" units_comment="m/s" />
    </imposevelocity>
</inoutzone>
```

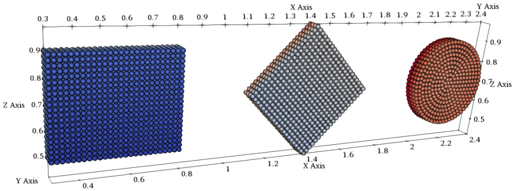
3-D Buffer Shapes: Diamond



Diamond

```
<inoutzone>
    <convertfluid value="false" comment="Converts fluid in inlet/outlet area (default=true)" />
    <layers value="4" comment="Number of inlet/outlet particle layers" />
    <zone3d comment="Input zone for 3-D simulations">
        <box>
            <point x="1.2" y="2.8" z="0.5" />
            <size x="0.4" y="0" z="0.4" />
            <direction x="0" y="-1" z="0" />
            <rotateaxis angle="-45" anglesunits="degrees">
                <point1 x="1.4" y="0" z="0.7" />
                <point2 x="1.4" y="1" z="0.7" />
            </rotateaxis>
        </box>
    </zone3d>
    <imposevelocity mode="0" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated value (default=0)">
        <velocity v="2" comment="Uniform velocity" units_comment="m/s" />
    </imposevelocity>
</inoutzone>
```

3-D Buffer Shapes: Circle



Circle

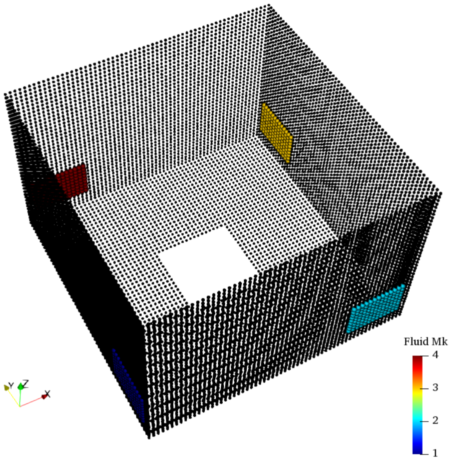
```
<inoutzone>
    <convertfluid value="false" comment="Converts fluid in inlet/outlet area (default=true)" />
    <layers value="4" comment="Number of inlet/outlet particle layers" />
    <zone3d comment="Input zone for 3-D simulations">
        <circle>
            <point x="2.2" y="2.8" z="0.7" />
            <radius v="0.2" />
            <direction x="0" y="-1" z="0" />
        </circle>
    </zone3d>
    <imposevelocity mode="0" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated value (default=0)">
        <velocity v="2" comment="Uniform velocity" units_comment="m/s" />
    </imposevelocity>
</inoutzone>
```

3-D Multi-Inlet Single-Outlet Flow

Name	Size	Type	Date Modified
1_FlowCylinder	13 items	Folder	Fri 19 Oct 2018 12:31:07 PM CEST
2_OpenChannel	7 items	Folder	Fri 19 Oct 2018 12:42:25 PM CEST
3_ReverseFlow	7 items	Folder	Fri 19 Oct 2018 12:44:21 PM CEST
4_Waves2D	8 items	Folder	Fri 19 Oct 2018 12:48:50 PM CEST
5_ShapesInlet3D	7 items	Folder	Fri 19 Oct 2018 01:04:08 PM CEST
6_Box4Inlet3D	7 items	Folder	Fri 19 Oct 2018 01:14:42 PM CEST
7_CurrentHull	8 items	Folder	Fri 19 Oct 2018 01:22:09 PM CEST

Folders of test cases in "examples/inletoutlet".

3-D Multi-Inlet Single-Outlet Flow

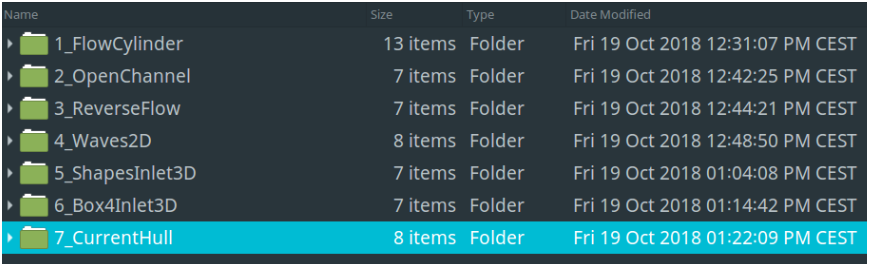


Multi-inlet single-outlet water flow in a cubical tank (semi-permeable inlet boundary)

3-D Multi-Inlet Single-Outlet Flow

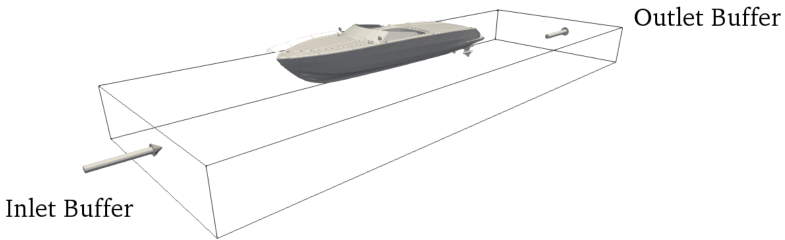
```xml
<inoutzone>
    <convertfluid value="false" comment="Converts fluid in inlet/outlet area (default=true)" />
    <layers value="6" comment="Number of inlet/outlet particle layers" />
    <zone3d comment="Input zone for 3-D simulations">
        <particles mkfluid="0" direction="right" />
    </zone3d>
    <imposevelocity mode="0" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated value (default=0)">
        <velocity v="2" comment="Uniform velocity" units_comment="m/s" />
    </imposevelocity>
</inoutzone>
<inoutzone>
    <convertfluid value="false" comment="Converts fluid in inlet/outlet area (default=true)" />
    <layers value="6" comment="Number of inlet/outlet particle layers" />
    <zone3d comment="Input zone for 3-D simulations">
        <particles mkfluid="1" direction="back" />
    </zone3d>
    <imposevelocity mode="0" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated value (default=0)">
        <velocity v="2" comment="Uniform velocity" units_comment="m/s" />
    </imposevelocity>
</inoutzone>
<inoutzone>
    <convertfluid value="false" comment="Converts fluid in inlet/outlet area (default=true)" />
    <layers value="6" comment="Number of inlet/outlet particle layers" />
    <zone3d comment="Input zone for 3-D simulations">
        <particles mkfluid="2" direction="left" />
    </zone3d>
    <imposevelocity mode="0" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated value (default=0)">
        <velocity v="2" comment="Uniform velocity" units_comment="m/s" />
    </imposevelocity>
</inoutzone>
<inoutzone>
    <convertfluid value="false" comment="Converts fluid in inlet/outlet area (default=true)" />
    <layers value="6" comment="Number of inlet/outlet particle layers" />
    <zone3d comment="Input zone for 3-D simulations">
        <particles mkfluid="3" direction="front" />
    </zone3d>
    <imposevelocity mode="0" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated value (default=0)">
        <velocity v="2" comment="Uniform velocity" units_comment="m/s" />
    </imposevelocity>
</inoutzone>
```

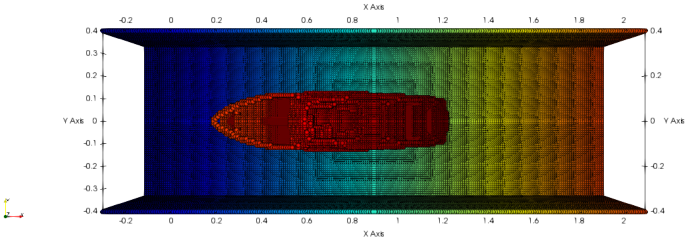
3-D Flow Past a Boat Hull



Folders of test cases in "examples/inletoutlet".

3-D Flow Past a Boat Hull





3-D Flow Past a Boat Hull

Inlet

```
<inoutzone>
    <convertfluid value="false" comment="Converts fluid in inlet/outlet area (default=true)" />
    <layers value="5" comment="Number of inlet/outlet particle layers" />
    <zone3d comment="Input zone for 2-D simulations">
        <particles mkfluid="1" direction="right" />
        <!-- *** Direction values: left, right, front, back, top, bottom *** -->
    </zone3d>
    <imposevelocity mode="0" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated velocity,
3:Interpolated velocity (default=0)">
        <velocity v="2" comment="Uniform velocity" units_comment="m/s" />
    </imposevelocity>
    <imposerhop mode="2" comment="Outlet rhop 0:Imposed fixed value, 1:Hydrostatic, 2:Extrapolated from ghost nodes
(default=0)" />

    <imposezsurf mode="0" comment="Inlet Z-surface 0:Imposed fixed value, 1:Imposed variable value, 2:Calculated from
fluid domain (default=0)">
        <zbottom value="0" comment="Bottom level of water (used for Hydrostatic option)" units_comment="m" />
        <zsurf value="0.2" comment="Characteristic inlet Z-surface (used for Hydrostatic option)" units_comment="m" />
    </imposezsurf>
</inoutzone>
```

Outlet

```
<inoutzone>
    <convertfluid value="true" comment="Converts fluid in inlet/outlet area (default=true)" />
    <layers value="8" comment="Number of inlet/outlet particle layers" />
    <zone3d comment="Input zone for 2-D simulations">
        <particles mkfluid="2" direction="left" />
    </zone3d>
    <imposevelocity mode="0" comment="Imposed velocity 0:fixed value, 1:variable value, 2:Extrapolated velocity,
3:Interpolated velocity (default=0)">
        <velocity v="-2" comment="Uniform velocity" units_comment="m/s" />
    </imposevelocity>
    <imposerhop mode="2" comment="Outlet rhop 0:Imposed fixed value, 1:Hydrostatic, 2:Extrapolated from ghost nodes
(default=0)" />
    <imposezsurf mode="0" comment="Inlet Z-surface 0:Imposed fixed value, 1:Imposed variable value, 2:Calculated from
fluid domain (default=0)">
        <zbottom value="0" comment="Bottom level of water (used for Hydrostatic option)" units_comment="m" />
        <zsurf value="0.2" comment="Characteristic inlet Z-surface (used for Hydrostatic option)" units_comment="m" />
    </imposezsurf>
</inoutzone>
```

Outline

Conclusion and Remarks

Strenghts

- Versatile SPH open boundary conditions for large-scale simulations (DualSPHysics)
- Use on CPU and GPU
- Many options for imposing flow parameters at the open boundaries
- Good results both in 2-D and 3-D

Limitations

- Method is first order at most
- Method is currently restricted to flat inlet/outlet
- Method lacks resume option (work is already in progress)

Directions for Future Work

Future Work

- Explore a more robust algorithm for generating buffer geometries based solely on surface normals
- Explore the use of higher order approximations for problems where the first order accuracy is not adequate
- Coupling with other methods (already being investigated)

References and Acknowledgements

Main reference paper to this work:

A. Tafuni, J. Domínguez, R. Vacondio, A. Crespo **A versatile algorithm for the treatment of open boundary conditions in smoothed particle hydrodynamics gpu models** *Comput Methods Appl Mech Eng* Volume 342, 1 December 2018, Pages 604-624, doi: 10.1016/j.cma.2018.08.004

THANK YOU FOR YOUR ATTENTION