XML GUIDE
FOR DUALSPHYSICS

OPEN BOUNDARY CONDITIONS
SPECIAL: INLET/OUTLET

July 2020

DualSPHysics team

http://dual.sphysics.org

XML file: Special-Inlet/Outlet Main idea of the methodology

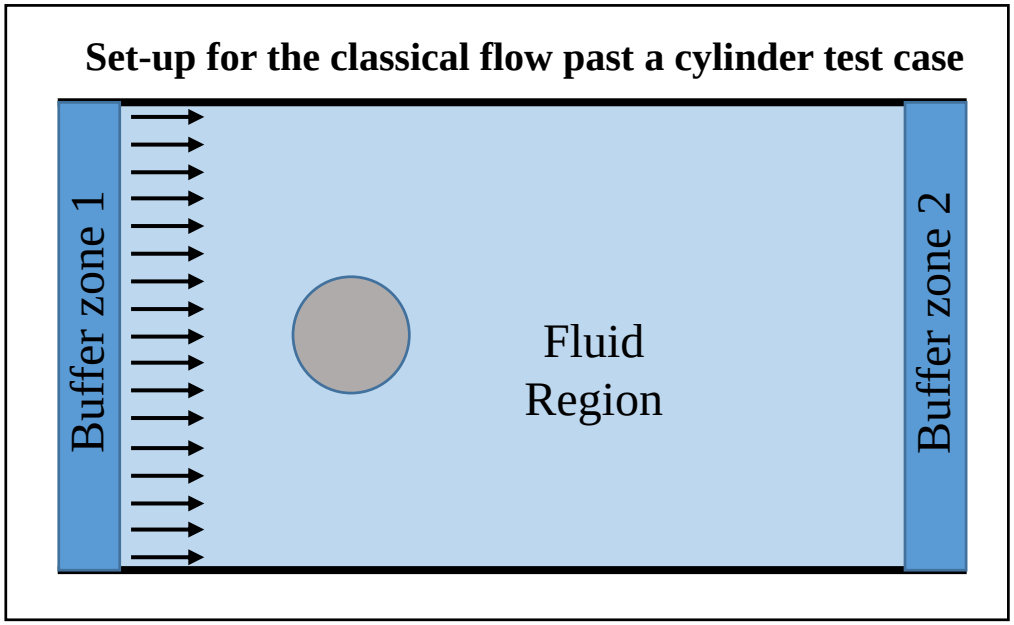
1 - Open boundaries are enforced creating buffer layers for inflow/outflow regions where buffer particles are created and deleted.

2 - Buffer regions prevent errors generated by the kernel truncation near the boundaries and particles inside the buffer areas are created and/or deleted to prevent the formation of voids

3 - Physical quantities (velocity and pressure) of the buffer particles can be either enforced or extrapolated from the fluid domain.

4 - A methodology to extrapolate (or impose) time-varying free-surface elevation in the buffer regions has been developed, allowing the generation of free-surface waves.

5 - A buffer layer can be both inlet and outlet depending on the type of problem to be solved, and thus on how the properties of these particles are calculated



XML file: Special-Inlet/Outlet

Inlet/outlet boundary conditions for imposing velocity, density-pressure, and free surface elevation at one (or more) open boundary of the SPH domain.

```
<special>
    <inout>
        <memoryresize size0="2" size="4" comment="Initial memory resize (size0) and the following memory
        resizes according to initial inlet/outlet particles (default=2 and 4)"/>
        <useboxlimit value="true" comment="(default=true)">
            <freecentre x="2" y="0" z="0" comment="(default=centre of simulation domain)"/>
        </useboxlimit>
        <determlimit value="1e+3" comment="(default=1e+3)" />
        <extrapolatemode value="1" comment="1:fast-single, 2:single, 3:double (default=1)" />
        <inoutzone>
            <refilling value="0" comment="0:Simple full, 1:Simple below zsurf, 2:Advanced (default=1)"/>
            <inputtreatment value="0" comment="0:No changes, 1:Convert fluid, 2:Remove fluid" />
            <layers value="6" comment="Number of inlet/outlet particle layers" />
            <zone2d comment="Input zone for 2-D simulations">
                <line>
                    <point x="-1.1" z="0.1" />
                    <point2 x="-1.1" z="1.0" />
                    <direction x="1" z="0" comment="Direction to input fluid" />
                </line>
            </zone2d>
            <imposevelocity mode="0" comment="0:fixed, 1:variable 2:Extrapolated (default=0)">
                <velocity v="2" comment="Uniform velocity" units_comment="m/s" />
            </imposevelocity>
            <imposerhop mode="2" comment="0:Imposed fixed, 1:Hydrostatic, 2:Extrap from g.n."/>
        </inoutzone>
        <inoutzone>
            <refilling value="1" comment="0:Simple full, 1:Simple below zsurf, 2:Advanced (default=1)"/>
            <inputtreatment value="1" comment="0:No changes, 1:Convert fluid, 2:Remove fluid" />
            <layers value="8" comment="Number of inlet/outlet particle layers" />
            <zone2d comment="Input zone for 2-D simulations">
                <particles mkfluid="2" direction="left" />
            </zone2d>
            <imposevelocity mode="0" comment="0:fixed, 1:variable 2:Extrapolated (default=0)">
                <velocity v="-2" comment="Uniform velocity" units_comment="m/s" />
            </imposevelocity>
            <imposerhop mode="2" comment="0:Imposed fixed, 1:Hydrostatic, 2:Extrap from g.n."/>
        </inoutzone>
    </inout>
</special>
```

XML file: Special-Inlet/Outlet MAIN PARAMETERS

memoryresize it is only related to computational efficiency. It controls how often the size of the particle arrays is increased (to avoid to many re-allocation). Do not change this if you don't really know what you are doing.

useboxlimit defines the portion of the domain where those options are valid:

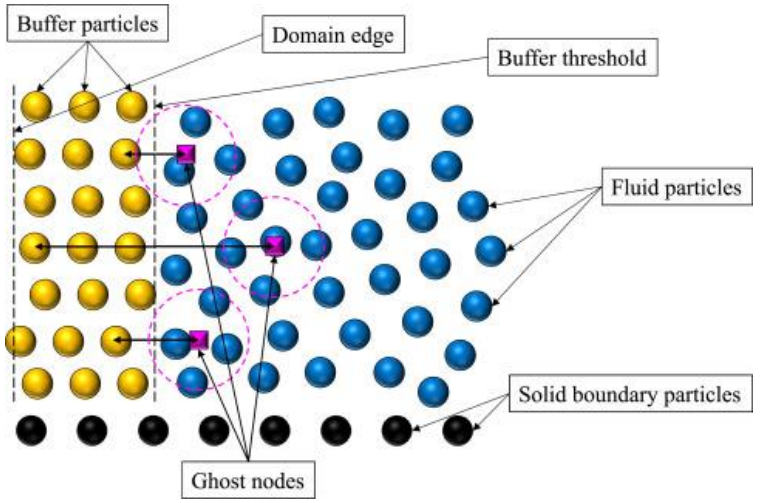
```
<useboxlimit value="true" comment="(default=true)">
    <freecentre x="2" y="0" z="0" comment="(default=centre of simulation domain)"/>
</useboxlimit>
```

If true, In/out process is only applied to InOut zones delimited by BoxLimit. This defines the portion of the domain where those options are valid. It is related to computational efficiency and default configuration is usually best option.

determlimit defines the type of extrapolation applied in the buffer zones:

```
<determlimit value="1e+3" comment="Use 1e-3 for 1st order or 1e+3 for 0th order (default=1e+3)"/>
```

If determlimit=1e-3, then a linear extrapolation is applied between the ghost node and the particle in the buffer, if determlimit=1e+3 then the values computed at the ghost node are replicated for the buffer particles. If you experience some instabilities near the buffer zone, you should try to use determlimit=1e+3 which is less accurate but more robust.



XML file: Special-Inlet/Outlet MAIN PARAMETERS

extrapolatemode defines the way the extrapolation of variables from fluid to buffer particles is operated.

```
<extrapolatemode value="1" comment="1:fast-single, 2:single, 3:double (default=1)" />
```

The method proposed by Liu & Liu (2026) is adopted to retrieve first order consistency. A linear system of 3 equations in 2D (and 4 equations in 3D) is solved to obtain the SPH approximation of the field function and its first order derivatives (see Tafuni et al. 2018 for further details on this):

$$f_k = \frac{\begin{vmatrix} \sum_l f_l W_{kl} \Delta V_l & \sum_l (\mathbf{x}_l - \mathbf{x}_k) W_{kl} \Delta V_l \\ \sum_l f_l W_{kl,\beta} \Delta V_l & \sum_l (\mathbf{x}_l - \mathbf{x}_k) W_{kl,\beta} \Delta V_l \end{vmatrix}}{\begin{vmatrix} \sum_l f(\mathbf{x}) W_{kl} \Delta V_l & \sum_l (\mathbf{x}_l - \mathbf{x}_k) W_{kl} \Delta V_l \\ \sum_l f(\mathbf{x}) W_{kl,\beta} \Delta V_l & \sum_l (\mathbf{x}_l - \mathbf{x}_k) W_{kl,\beta} \Delta V_l \end{vmatrix}}$$

$$f_{k,\beta} = \frac{\begin{vmatrix} \sum_l W_{kl} \Delta V_l & \sum_l f_l W_{kl} \Delta V_l \\ \sum_l W_{kl,\beta} \Delta V_l & \sum_l f_l W_{kl,\beta} \Delta V_l \end{vmatrix}}{\begin{vmatrix} \sum_l W_{kl} \Delta V_l & \sum_l (\mathbf{x}_l - \mathbf{x}_k) W_{kl} \Delta V_l \\ \sum_l W_{kl,\beta} \Delta V_l & \sum_l (\mathbf{x}_l - \mathbf{x}_k) W_{kl,\beta} \Delta V_l \end{vmatrix}}$$

if extrapolatemode is equal to:

- “1” -ffast-math option is activated,
- “2” regular single precision is adopted,
- “3” double precision is enabled.

Unless you experience instabilities in the buffer region, you should not change this

XML file: Special-Inlet/Outlet CREATE BUFFERS 2D

Buffer layers are created by defining simple geometries, such as a rectangle or a circle. For example, in a **2-D simulation** with a flat domain edge, the buffer area is invoked by the tag '**zone2d**', and can be defined in two ways.

1) The innermost buffer layer (closest to the fluid) is defined by a '**line**' passing between two points (**point, point2**). '**direction**' specifies the direction for the creation of the ghost nodes. '**rotate**' allows to rotate the layer about its center by a certain angle.

```
<zone2d comment="Input zone for 2-D simulations">
    <line>
        <point x="-1.1" z="0.1" />
        <point2 x="-1.1" z="1.0" />
        <direction x="1" z="0" comment="Direction to input fluid" />
    </line>
</zone2d>
```

2) An alternative is provided to create the innermost buffer layer by simply defining a direction to a fictitious set of SPH fluid particles. In the below, such particles have '**mkfluid=2**' and the buffer is created to the right of the fluid, hence ghost nodes are created to the left of the buffer, thus **direction="left"** is used.

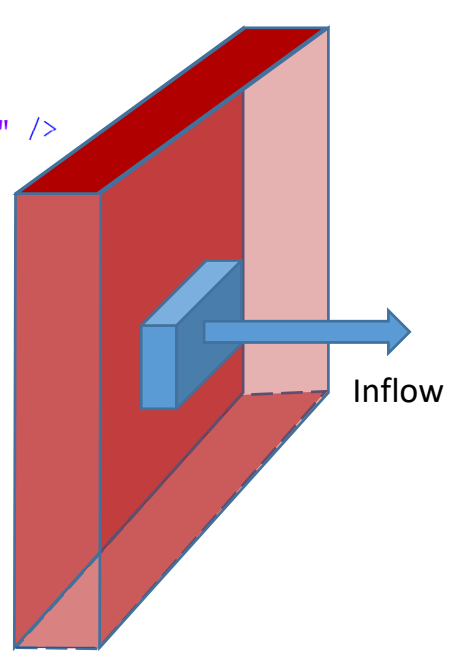
```
<zone2d comment="Input zone for 2-D simulations">
    <particles mkfluid="2" direction="left" />
</zone2d>
```

XML file: Special-Inlet/Outlet CREATE BUFFERS 3D

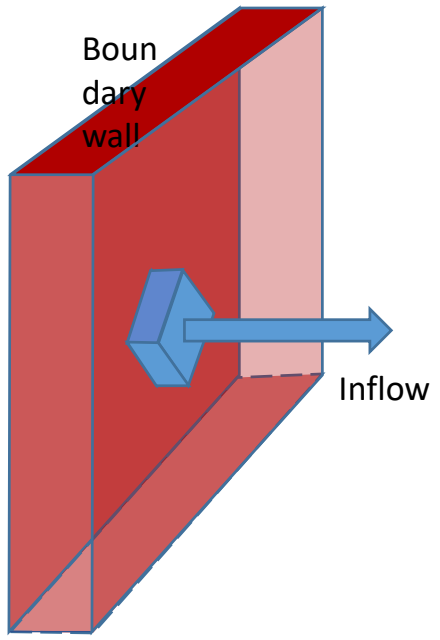
Buffer layers are created by defining simple geometries, such as a rectangle or a circle. For a **3-D simulation**, the tag becomes '**zone3d**' and is used as follows:

1) The command '**box**' is used for flat box-shaped buffers. The innermost surface (closest to the fluid) is created by extruding from a seed point '**point**' with sizes indicated in '**size x= y= z=** '. The buffer can be rotated similarly to the 2-D case.

```
<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" anglesunit="degrees" />
            <point1 x="1.4" y="0" z="0.7" />
            <point2 x="1.4" y="1" z="0.7" />
        </rotateaxis>
    </box>
</zone3d>
```



BufferZone 3D: Box



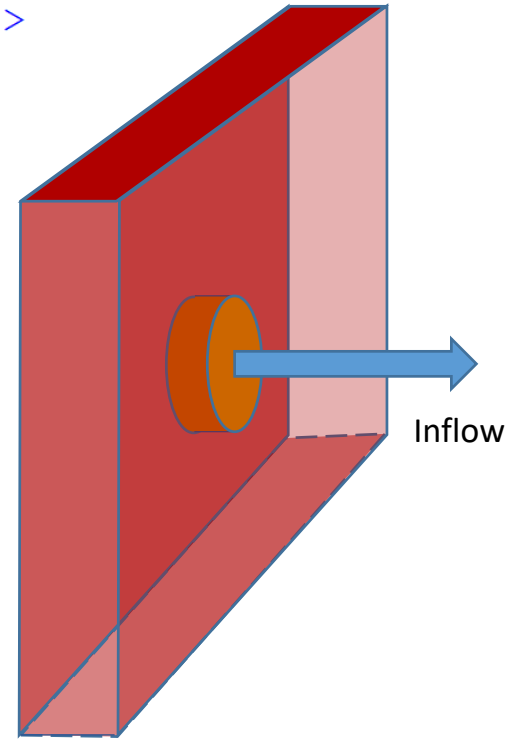
BufferZone 3D: Box Rotated 45°

XML file: Special-Inlet/Outlet CREATE BUFFERS 3D

Buffer layers are created by defining simple geometries, such as a rectangle or a circle. For a 3-D simulation, the tag becomes '**zone3d**' and is used as follows:

2) A circular counterpart is created with the command '**circle**' to produce a cylindrical inlet or outlet. The innermost surface (closest to the fluid) is created by indicating the circle center via '**point**', its '**radius**', and the '**direction**' of creation of the ghost nodes.

```
<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>
```



BufferZone 3D: Circle

XML file: Special-Inlet/Outlet CREATE BUFFERS 3D

Geometry of the buffers- buffer layers are created by defining simple geometries, such as a rectangle or a circle. For a 3-D simulation, the tag becomes '**zone3d**' and is used as follows:

3) As for the 2-D case, particles for 3-D buffer areas can also be created relative to special fluid particles, with given **mkfluid** and **direction** for ghost nodes. Only one fluid 'mk' can be used per buffer area. The example below shows the creation of four inlets from four fluid areas in the example '**Box4Inlet3D**'

```
<inoutzone>
    <refilling value="0" />
    <inputtreatment value="2" />
    <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" >
        <velocity v="2" comment="Uniform velocity" units_comment="m/s" />
    </imposevelocity>
</inoutzone>
<inoutzone>
    <refilling value="0" />
    <inputtreatment value="2"  />
    <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" >
       <velocity v="2" comment="Uniform velocity" units_comment="m/s" />
    </imposevelocity>
</inoutzone>
```

XML file: Special-Inlet/Outlet OPTIONS FOR EACH BUFFER ZONE

Each buffer zone has three different options that needs to be defined:

```
<inoutzone>
    <refilling value="0" />
    <inputtreatment value="2" />
    <layers value="6" comment="Number of inlet/outlet particle layers" />
```

The option '**refilling**' defines the way the buffer zone is kept filled with buffer particles when they enter in the fluid region (and so are converted in fluid ones). Options are the followings:

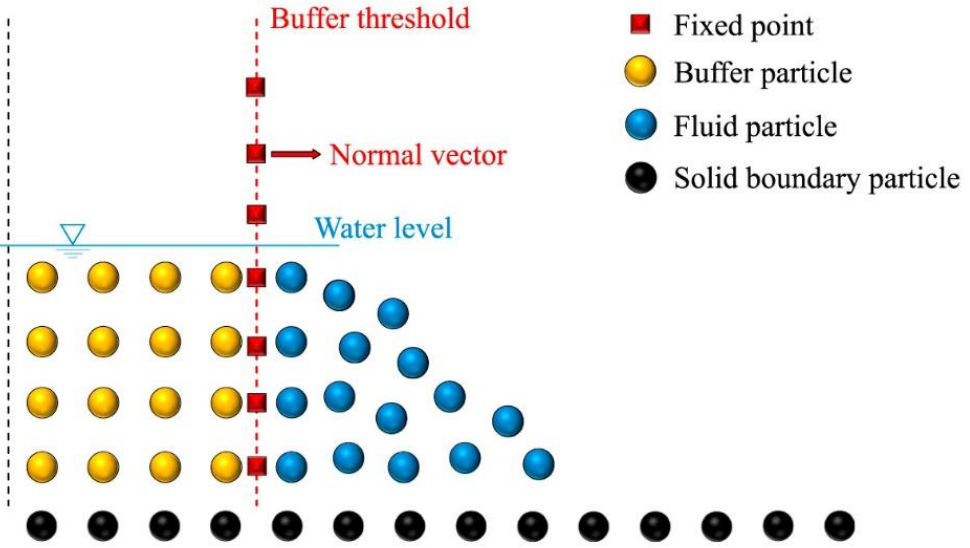
- **0 (Simple-Full):** The inlet / outlet area is filled with buffer particles up to the top (no free-surface limit (Zsurf) is considered). For each buffer particle that enters in the fluid region, a new buffer particle is created. This option can still be used with variable free-surface when velocity is imposed (inlet), because particles with Z > Zsurf have velocity equal to 0. It can give problems when velocity is extrapolated from the fluid region or for reverse flows because gaps may appear between particles in the buffer region.
- **1 (Simple-Zsurf):** The buffer region is filled with buffer particles up to Z <= Zsurf. For each inlet particle that enters in the fluid region, a new particle is created only if Z <= Zsurf. It does not work for free surface variable in time (increasing Zsurf does not create new rows of buffer particles). In reverse flows holes that may appear in the buffer region.
- **2 (Advanced):** This allows reverse flows but it is less computationally efficient than the previous ones. This method creates new buffer particles to fill possible holes in the buffer region. For quickly changing free-surface level simulations, it can be used in combination with local shifting to completely avoid holes in the buffer region.

XML file: Special-Inlet/Outlet OPTIONS FOR EACH BUFFER ZONE

The option "**inputtreatment**" defines the treatment of fluid particles entering the buffer zone:

- 0 (No changes): Fluid particles entering the buffer region are not converted in buffer ones. It can be used only for inlet (with velocity imposed), it is not suitable for outlet regions or reverse flows.
- 1 (Convert): Fluid particles entering the buffer zone are converted in buffer ones. This option is essential for outlet regions (and reverse flow).
- 2 (Remove): Fluid particles entering the inlet zone (regardless of its Z) they are excluded from the simulation. This option is not suitable for outlet regions or reverse flows. It is also the recommended option for inlet zones because if some fluid particles enter the buffer zone they are removed.

The option "**layers**" defines the number of layers of the buffer particles for the buffer zone. In the figure below there are 4 layers of particles. It is important that the closest fluid particles to the buffer have a full kernel support, therefore at least 5 layers are recommended, with higher values corresponding to higher h/dp ratios.



XML file: Special-Inlet/Outlet PHYSICAL VARIABLES

Physical variables (density, velocity and water depth) in the buffer zones can either be specified by the user or retrieved from the fluid region. The three variables over which a user has control are the *velocity*, for which the tag '**imposevelocity**' is used, the *density*, for which the tag '**imposerhop**' is used, and the *water level*, for which the tag '**imposezsurf**' is used.

While velocity and density are defined for each buffer particle, the water depth is defined in a unique way for the whole buffer region.

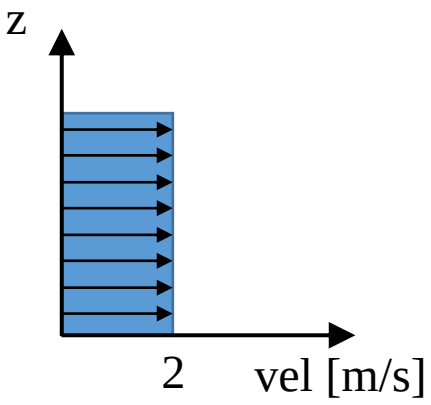
```
<imposevelocity mode="0" comment="Imposed velocity
    0:fixed value,
    1:variable value,
    2:Extrapolated value
    3:Interpolated velocity (default=0)">

<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)">
```

XML file: Special-Inlet/Outlet VELOCITY

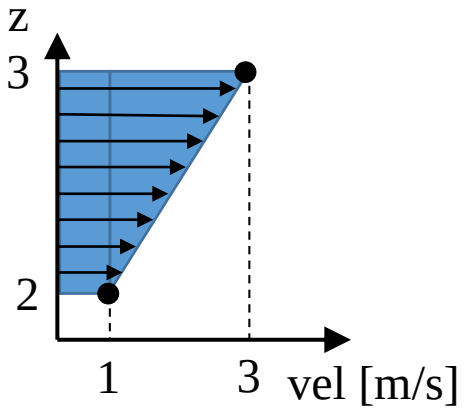
Buffer variables: velocity- when option "0" is selected for 'imposevelocity', each buffer particle has a velocity value that remains *constant in time*. This value can be the same for all particles, or change in space according to predefined velocity profiles.

Three different types of velocity profiles can be imposed:

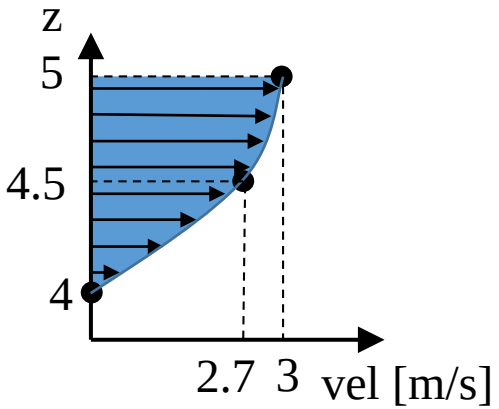
Fixed constant velocity



Fixed linear velocity



Fixed parabolic velocity



XML file: Special-Inlet/Outlet VELOCITY

'velocity' sets the velocity value according to a *uniform* profile defined by '**v**'

'velocity2' sets the velocity value according to a *linear* profile in z defined by 2 points

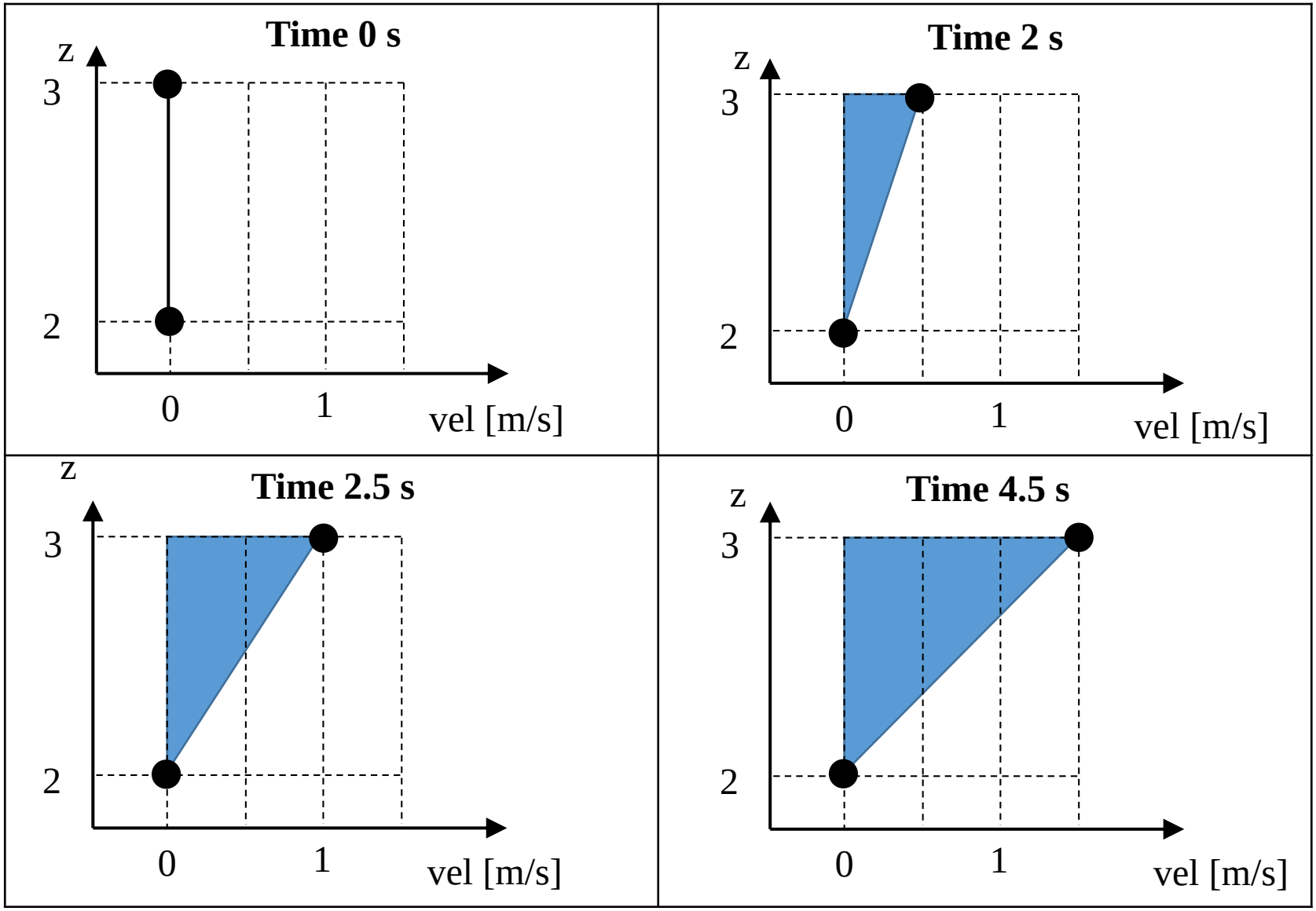
'velocity3' sets the velocity value according to a *parabolic* profile in z defined by 3 points

And they are defined in the xml files as follows:

```
<!-- Fixed Constant velocity -->
<imposevelocity mode="0" comment="Imposed velocity 0:fixed value in time">
    <velocity v="2" comment="Uniform velocity" units_comment="m/s" />
</imposevelocity>
<!-- Fixed Linear velocity -->
<imposevelocity mode="0" comment="Imposed velocity 0:fixed value in time">
    <velocity2 v="1" v2="3" z="2" z2="3" comment="Linear velocity" units_comment="m/s" />
</imposevelocity>
<!-- Fixed Parabolic velocity -->
<imposevelocity mode="0" comment="Imposed velocity 0:fixed value in time">
    <velocity3 v="0" v2="0.608" v3="3" z="4" z2="4.5" z3="5" comment="Parabolic velocity" units_comment="m/s" />
</imposevelocity>
```

XML file: Special-Inlet/Outlet VELOCITY

To impose a velocity profile that is changing in time such as this:



XML file: Special-Inlet/Outlet VELOCITY

Buffer variables: velocity- option "1" has to be selected for 'imposevelocity', each buffer particle has a velocity value that can *change in time*. Similarly to the previous case, this value can be the same for all particles, or change in space according to predefined velocity profiles.

An unsteady linear velocity profile case is shown as an example:

```
<!-- Variable Linear velocity using values in XML file -->
<imposevelocity mode="1" comment="Imposed velocity 1:variable value ">
    <velocitytimes2 comment="Linear inlet velocity in time">
        <timevalue time="0.0" v="0" v2="0.0" z="2" z2="3" />
        <timevalue time="2.0" v="0" v2="0.5" z="2" z2="3" />
        <timevalue time="2.5" v="0" v2="1.0" z="2" z2="3" />
        <timevalue time="4.5" v="1" v2="1.5" z="2" z2="3" />
    </velocitytimes2>
</imposevelocity>
```

'velocitytimes2' is used instead of 'velocity2' to enforce an unsteady profile. The value of the linear profile changes according to the sets of values indicated in '**timevalue**', which include the specific time at which the linear profile changes according to the values of '**v**', '**v2**', '**z**', and '**z2**'.

Velocity profiles can also be imported from external files. For example, for a uniform value of the velocity that changes over time, the tag '**velocityfile**' is used as follows:

```
<velocityfile file="Waves.csv" comment="Uniform inlet velocity data (time;v)"/>
```

XML file: Special-Inlet/Outlet VELOCITY

Buffer variables: velocity- when option "2" is selected for 'imposevelocity', the velocity of the buffer particles is extrapolated from the corresponding *ghost nodes*, and no other parameter need be specified.

Option "3" allows the interpolation of the velocity profile from an *external mesh*, which can be provided using the following syntax

```
<imposevelocity mode="3" comment="Imposed velocity 0:fixed value, 1:variable
value, 2:Extrapolated velocity, 3:Interpolated velocity (default=0)">
    <gridveldata file="wave.csv" comment="CSV file with velocity data in
    different times and positions" />
    <gridresetvelz value="true" comment="Reset Z velocity after interaction
    (default=false)" />
    <gridposzero x="0" z="0" comment="Defines position of grid (default=0)"/>
</imposevelocity>
```

'gridveldata' provides the path to the file where the grid is contained.

'**gridresetvelz**' avoid Z velocity of buffer particles.

'**gridposzero**' defines the origin of grid data.

XML file: Special-Inlet/Outlet DENSITY

Buffer variables: density- the value of the density of buffer particles can be chosen following three options:

```
<imposerhop mode="0" comment="Inlet/outlet rhop
    0:Imposed fixed value,
    1:Hydrostatic,
    2:Extrapolated from ghost nodes (default=0)"/>
```

'mode=0' sets the density of all particles equal to reference density of the fluid.

'mode=1' sets the density of buffer particles to yield a hydrostatic pressure distribution in *z*; for this, the initial water depth '**zbottom**' and free surface level '**zsurf**' should be defined in <imposezsurf> section.

'mode=2' sets the density of all particles according to extrapolated values from the ghost *nodes*

XML file: Special-Inlet/Outlet SURFACE ELEVATION

Buffer variables: water depth- the value of the water depth of buffer particles can be chosen based on three options, following the same rationale seen earlier for the velocity:

```
<imposezsurf mode="0" comment="Inlet Z-surface
    0:Imposed fixed value,
    1:Imposed variable value,
    2:Calculated from fluid domain (default=0)">
```

'mode=0' sets the buffer depth to a constant value of **zsurf** over time

'mode=1' changes the buffer depth over time according to set values of time and depth

'mode=2' sets the buffer depth according to values extrapolated from the ghost nodes

```
<!-- Imposed fixed zsurf -->
    <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)"/>
        <zsurf value="1" comment="Characteristic inlet Z-surface (for Hydrostatic density)"/>
    </imposezsurf>
```

'zbottom' defines the bottom level of water.

'**zsurf**' defines the Z of water level.

Both values are calculated from buffer dimensions when they are missing.

XML file: Special-Inlet/Outlet SURFACE ELEVATION

Buffer variables: water depth- an example of water depth changing over time is shown:

```
<!-- Imposed variable zsurf -->
    <imposezsurf mode="1" comment="Inlet Z-surface 0:Imposed fixed value,
    1:Imposed variable value, 2:Calculated from fluid domain (default=0)">
        <savevtk value="true" comment="Creates VTK files with Zsurf for each
        PART (default=false)" />
        <remove value="true" comment="Removes particles above the Zsurf
        limit (default=false)" />
        <zbottom value="0" comment="Bottom level of water (used for
        Hydrostatic option)" units_comment="m" />
        <zsurftimes comment="Characteristic inlet Z-Surf changes in time"
        units_comment="m/s">
            <timevalue time="0" zsurf="1" />
            <timevalue time="1" zsurf="1" />
            <timevalue time="2" zsurf="0.4" />
            <timevalue time="3" zsurf="0.4" />
            <timevalue time="5" zsurf="2" />
            <timevalue time="7" zsurf="0.4" />
        </zsurftimes>
    </imposezsurf>
```

As can be seen, similarly to 'velocitytimes' for the velocity, here the tag '**zsurftimes**' allows to specify the time instants of water depth change and new water depth value via '**timevalue**', after which a pair of values '**time**' and '**zsurf**' must be provided.

'remove' at every time step, this option removes buffer particles that are above the free surface level measured at close-by ghost nodes, leaving the buffer depth at the same level of the fluid nearby.

XML file: Special-Inlet/Outlet SURFACE ELEVATION

Buffer variables: water depth- an example of water depth from an external source:

```
<!-- Imposed variable zsurf using values in a external file-->
    <imposezsurf mode="1" comment="Inlet Z-surface 0:Imposed fixed value,
    1:Imposed variable value, 2:Calculated from fluid domain (default=0)">
        <savevtk value="true" comment="Creates VTK files with Zsurf for each
        PART (default=false)" />
        <remove value="false" comment="Removes particles above the Zsurf
        limit (default=false)" />
        <zbottom value="0" comment="Bottom level of water (used for
        Hydrostatic option)" units comment="m" />
        <zsurffile file="FileZsurf.csv" comment="Zsurf data (time;zsurf)"/>
    </imposevelocity>
```

zsurffile this allows to read the buffer depth from the file “FileZsurf.csv”, where the data must be arranged following the format (time;zsurf)

When working with variable buffer depth, it is also possible to generate VTK files of the variable *zsurf* by enabling ‘**savevtk**’ with the option *true*

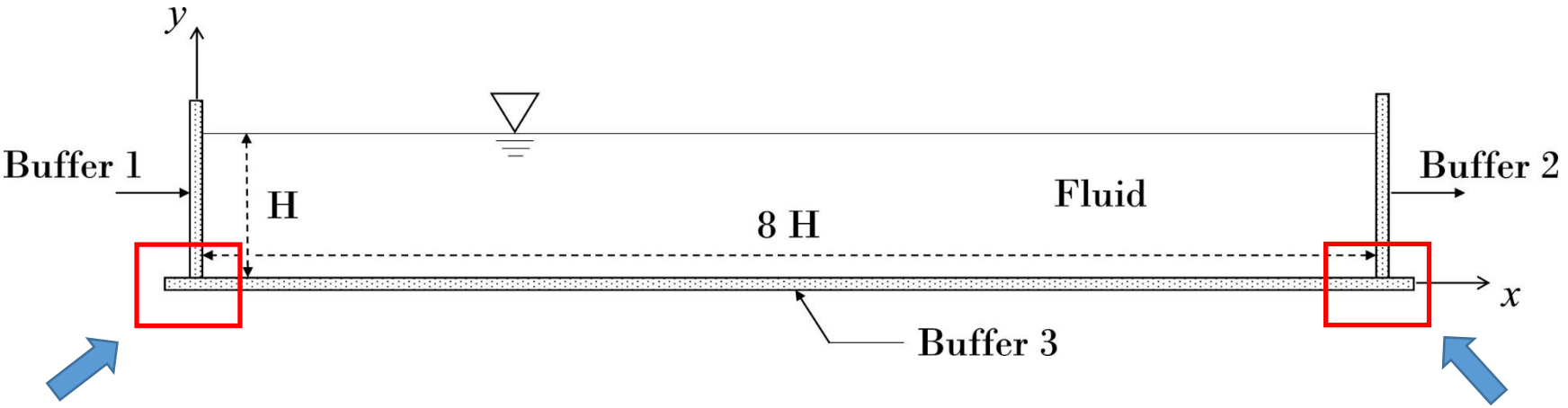
XML file: Special-Inlet/Outlet SURFACE ELEVATION

Buffer variables: water depth- the value of the water depth of buffer particles can be chosen based on three options, following the same rationale seen earlier for the velocity:

```
<!-- Zsurf calculated from fluid -->
    <imposezsurf mode="2" comment="Inlet Z-surface 0:Imposed fixed value,
    1:Imposed variable value, 2:Calculated from fluid domain (default=0)">
        <savevtk value="true" comment="Creates VTK files with Zsurf for each
        PART (default=false)" />
        <remove value="false" comment="Removes particles above the Zsurf
        limit (default=false)" />
        <zbottom value="0" comment="Bottom level of water (used for
        Hydrostatic option)" units_comment="m" />
        <zsurf value="1" comment="Initial Z-surface" units_comment="m" />
    </imposezsurf>
```

the water surface level is automatically calculated in front of the buffer during the simulation.

It is important to know that, if regular DBC are used for walls (closed boundaries) instabilities might appear in the region near the buffer zones (red squares in the figure below):



To avoid this there are two options:

1. use <boundcorr> feature for specific portion of the boundaries (see FlowCylinder example in **inletoutlet** folder)
2. use M-DBC for the whole closed boundaries in the simulation (see examples in **mdbc** folder)