Updates on boundary and initial conditions: How to check for normals

Aaron English
University of Parma

Outline

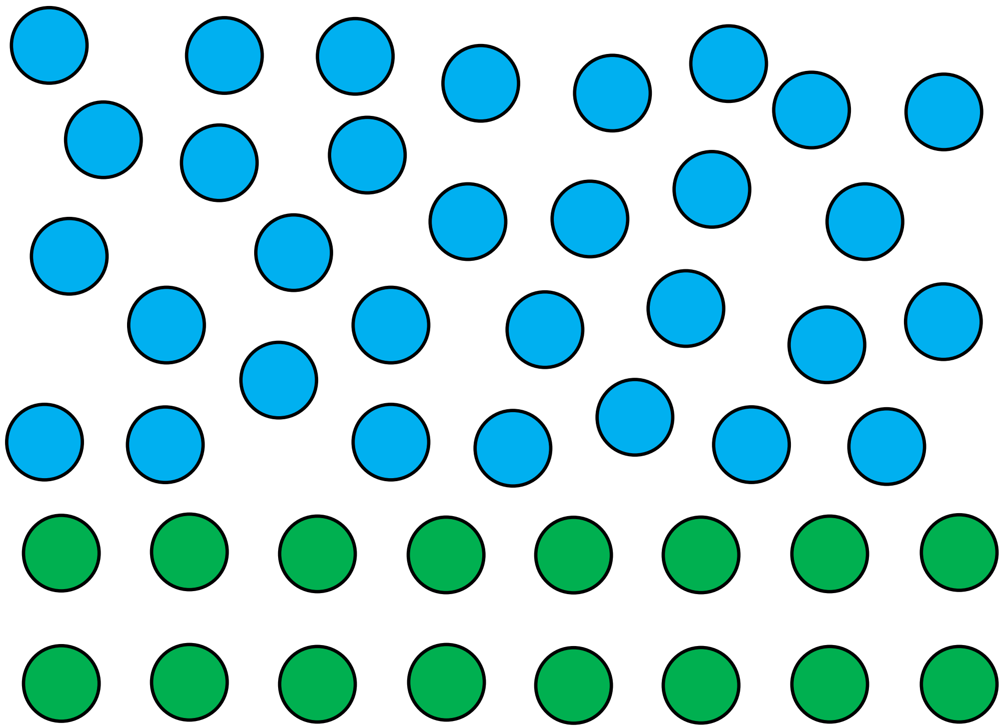
- Boundary Conditions in DualSPHysics – old and new
- Mixed boundary conditions by mkbound - new
- Defining initial conditions from previous simulations - new
- How to define and check your normals - advice

Boundary conditions in DSPH

DBC

$$\frac{d\rho_i}{dt} = \sum_j m_j \mathbf{u}_{ij} \cdot \nabla W_{ij}$$

$$\mathbf{u}_i = \mathbf{0}$$

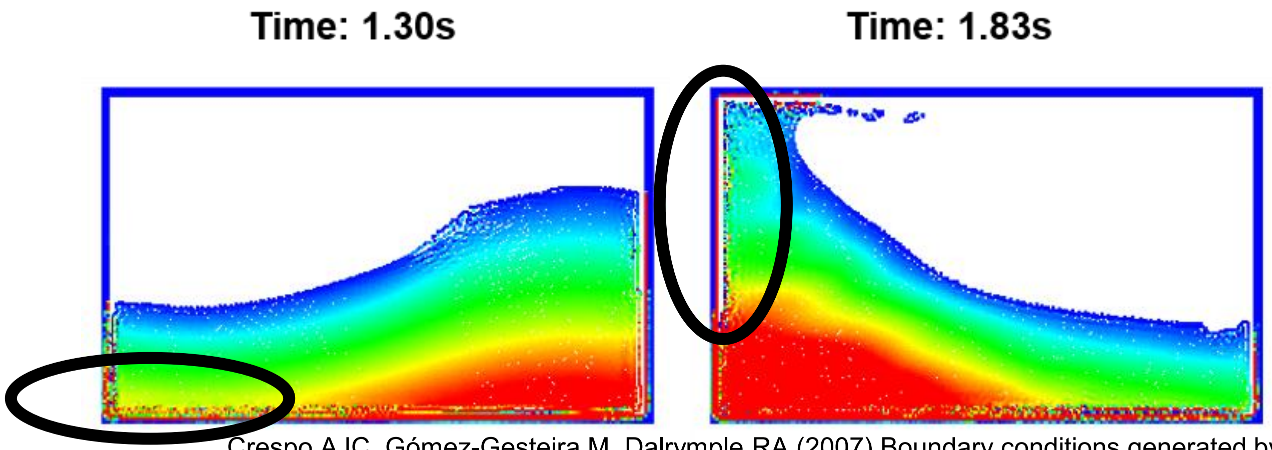


ADVANTAGES

- high computational efficiency
- ability to deal with very complex geometries
- can be used for real engineering problems

DISADVANTAGES

- unphysical density/pressure values of the boundary particles
- high repulsive force resulting in a separation distance (GAP)



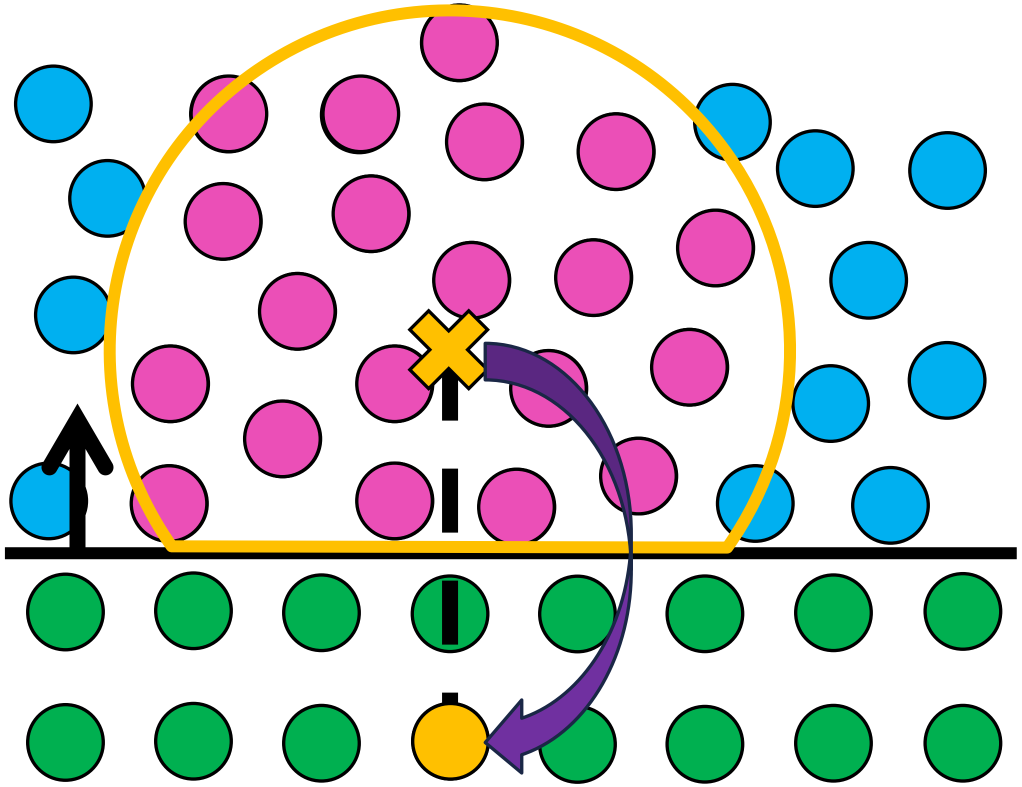
Crespo AJC, Gómez-Gesteira M, Dalrymple RA (2007) Boundary conditions generated by dynamic particles in SPH methods. Comput Mater Contin 5:173–184. https://doi.org/10.3970/cmc.2007.005.173

Boundary conditions in DSPH

mDBC slip mode 1 (old mDBC)

Vel = 0 boundary

Use boundary normal to create ghost node in fluid domain



$$\boldsymbol{u}_b = 0$$

$$\boldsymbol{A}_g \cdot \left[\rho_g \;\; \partial_x\rho_g \;\; \partial_y\rho_g \;\; \partial_z\rho_g\right]^{\mathrm{T}} = \mathbf{b}_g$$

$$\mathbf{b}_g = \left[\sum_j \rho_j V_j W_{gj} \;\; \sum_j \rho_j V_j \partial_x W_{gj} \;\; \sum_j \rho_j V_j \partial_y W_{gj} \;\; \sum_j \rho_j V_j \partial_z W_{gj}\right]^{\mathrm{T}}$$

$$\boldsymbol{A}_g = \begin{bmatrix} \sum_j V_j W_{gj} & \sum_j (x_j - x_g) V_j W_{gj} & \sum_j (y_j - y_g) V_j W_{gj} & \sum_j (z_j - z_g) V_j W_{gj} \\ \sum_j V_j \partial_x W_{gj} & \sum_j (x_j - x_g) V_j \partial_x W_{gj} & \sum_j (y_j - y_g) V_j \partial_x W_{gj} & \sum_j (z_j - z_g) V_j \partial_x W_{gj} \\ \sum_j V_j \partial_y W_{gj} & \sum_j (x_j - x_g) V_j \partial_y W_{gj} & \sum_j (y_j - y_g) V_j \partial_y W_{gj} & \sum_j (z_j - z_g) V_j \partial_y W_{gj} \\ \sum_j V_j \partial_z W_{gj} & \sum_j (x_j - x_g) V_j \partial_z W_{gj} & \sum_j (y_j - y_g) V_j \partial_z W_{gj} & \sum_j (z_j - z_g) V_j \partial_z W_{gj} \end{bmatrix}$$

$$\boxed{\rho_b = \rho_g + (\mathbf{r}_{\mathrm{b}} - \mathbf{r}_{\mathrm{g}}) \cdot \left[\partial_x\rho_g \;\; \partial_y\rho_g \;\; \partial_z\rho_g\right]^{\mathrm{T}}}$$

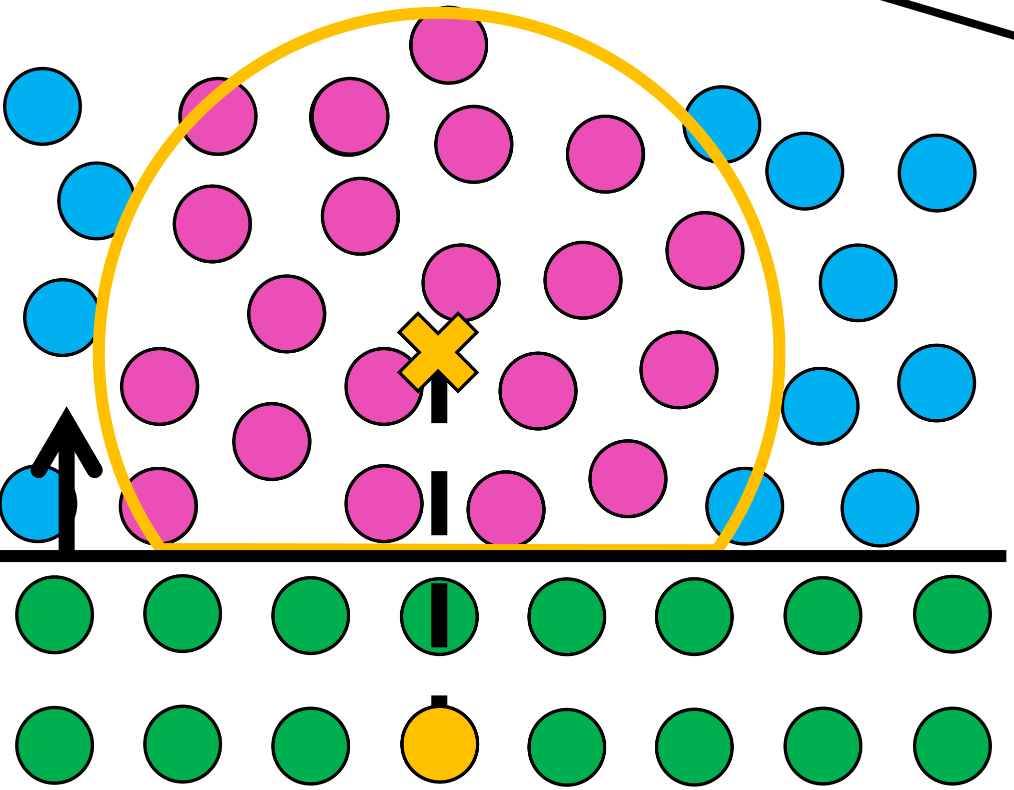
English, A., Domínguez, J.M., Vacondio, R. *et al.* Modified dynamic boundary conditions (mDBC) for general-purpose smoothed particle hydrodynamics (SPH): application to tank sloshing, dam break and fish pass problems. *Comp. Part. Mech.* **9**, 1–15 (2022). https://doi.org/10.1007/s40571-021-00403-3

Boundary conditions in DSPH

mDBC slip mode 2 (new mDBC)

Matrix inversion
Invert the matrix $\boldsymbol{A}_g$ to find the density -> ρ_g

Shepherd
Use a shepherd sum to find the density -> ρ_g



ρ_g

$$P_g = c_0^2\left(\rho_g - \rho_0\right)$$

$$P_b = P_g + \rho_0[(\boldsymbol{g} - \boldsymbol{a}_b) \cdot \boldsymbol{n_b}][(\boldsymbol{x}_g - \boldsymbol{x}_b) \cdot \boldsymbol{n_b}]$$

$$\rho_b = \rho_0 + \frac{P_b}{c_0^2}$$

Antuono M., Pilloton C., Colagrossi A., Durante D. Clone particles: A simplified technique to enforce solid boundary conditions in SPH Comput. Methods Appl. Mech. Engrg., 409 (2023), Article 115973

Boundary conditions in DSPH

mDBC slip mode 2 (new mDBC)

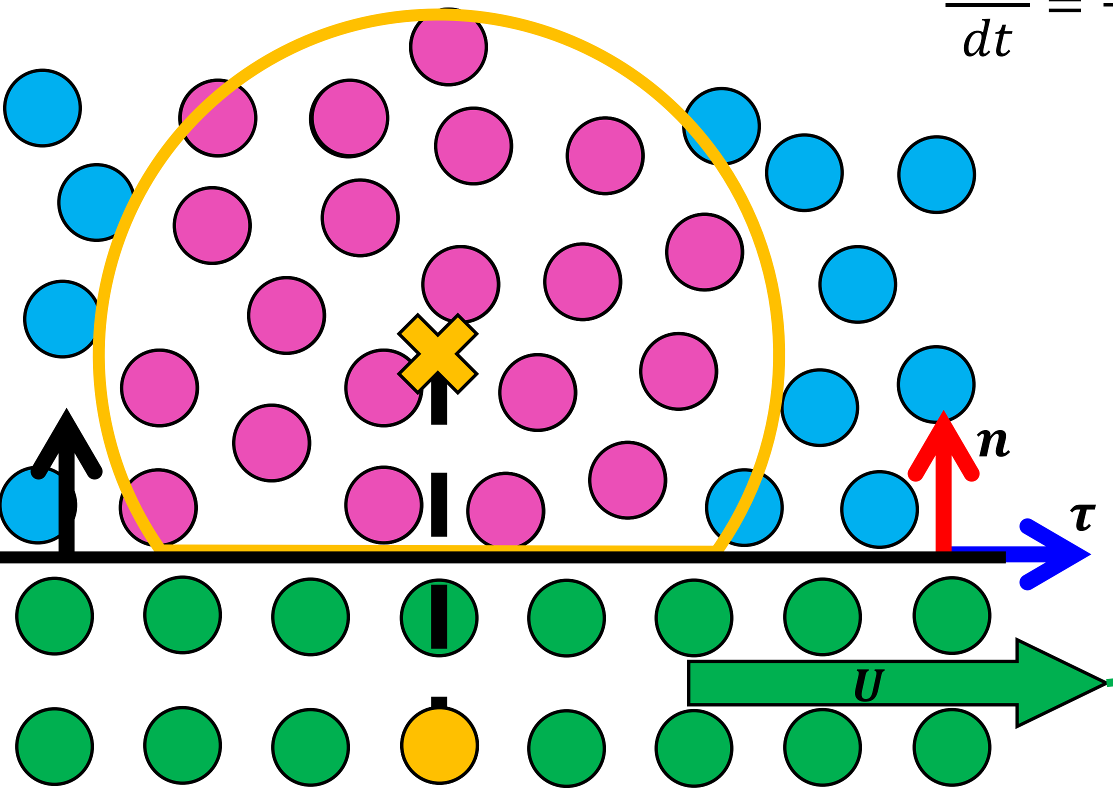
$$\boldsymbol{u}_g = \frac{\sum_j \boldsymbol{u}_j V_j W_{gj}}{\sum_j V_j W_{gj}} \qquad \boldsymbol{u}_b = 2\boldsymbol{U} - \boldsymbol{u}_g$$

No-slip boundary

$$\frac{d\boldsymbol{u}_i}{dt} = -\sum_j m_j \left(\frac{P_i + P_j}{\rho_i \, \rho_j}\right) \nabla W_{ij} + \boxed{\sum_j \frac{4\nu \, m_j \, \boldsymbol{r}_{ij} \cdot \nabla W_{ij}}{(\rho_i + \rho_j)\,(\boldsymbol{r}_{ij}^2 + 0.01h^2)} \boldsymbol{u}_{ij}} + \boldsymbol{g}_i$$

$$\boldsymbol{u}_b \cdot \boldsymbol{\tau} = (2\boldsymbol{U} - \boldsymbol{u}_g) - \left((2\boldsymbol{U} - \boldsymbol{u}_g) \cdot \boldsymbol{n}\right)$$

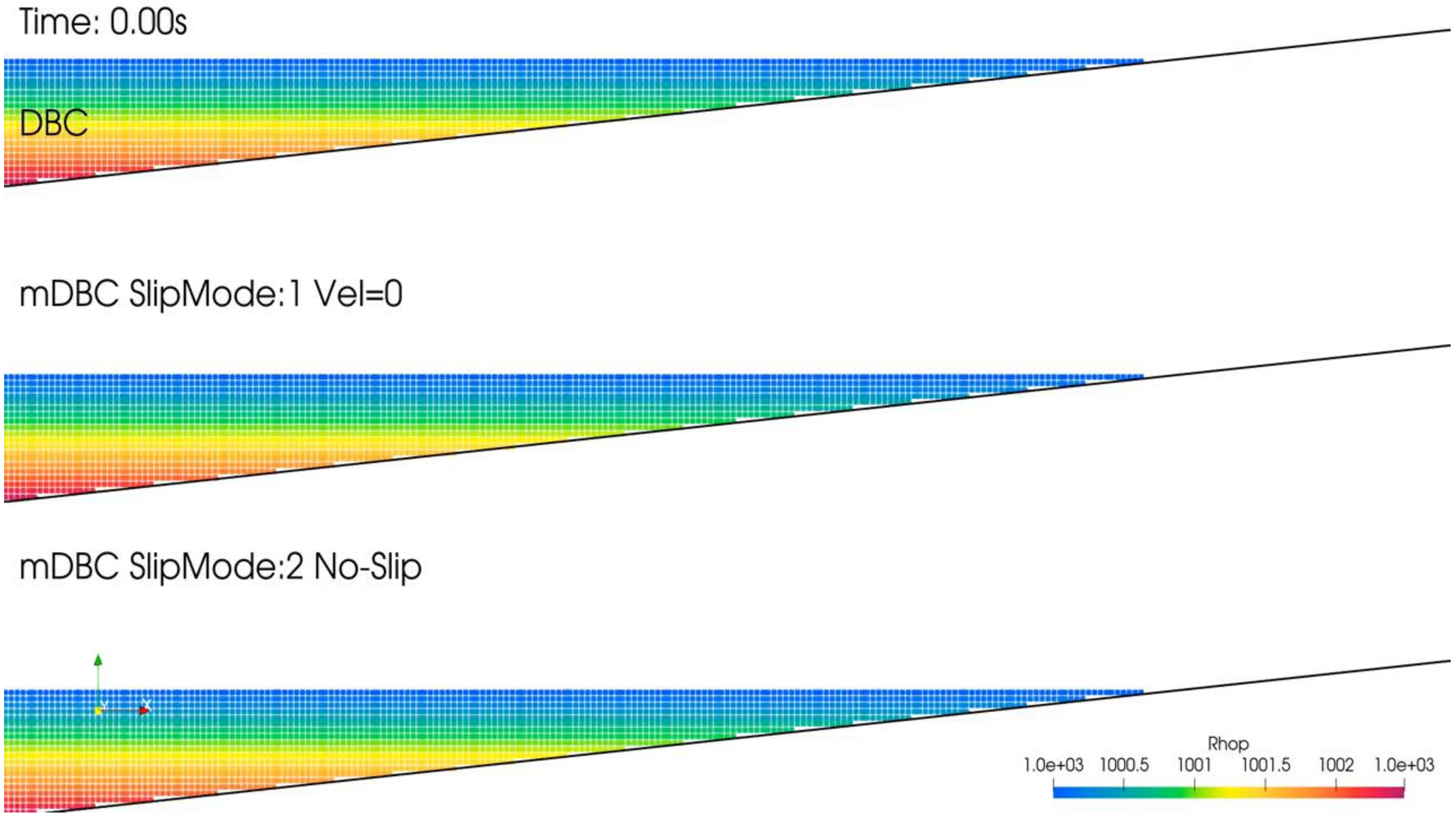
$$\cancel{\boldsymbol{u}_b \cdot \boldsymbol{n} = (2\boldsymbol{U} - \boldsymbol{u}_g) \cdot \boldsymbol{n}}$$



$$\frac{d\rho_i}{dt} = \rho_i \sum_j \frac{m_j}{\rho_j} \boxed{\boldsymbol{u}_{ij}} \cdot \nabla W_{ij} + DDT$$

Boundary conditions in DSPH

mDBC slip mode 2



Boundary conditions in DSPH

mDBC slip mode 3 (NEW)

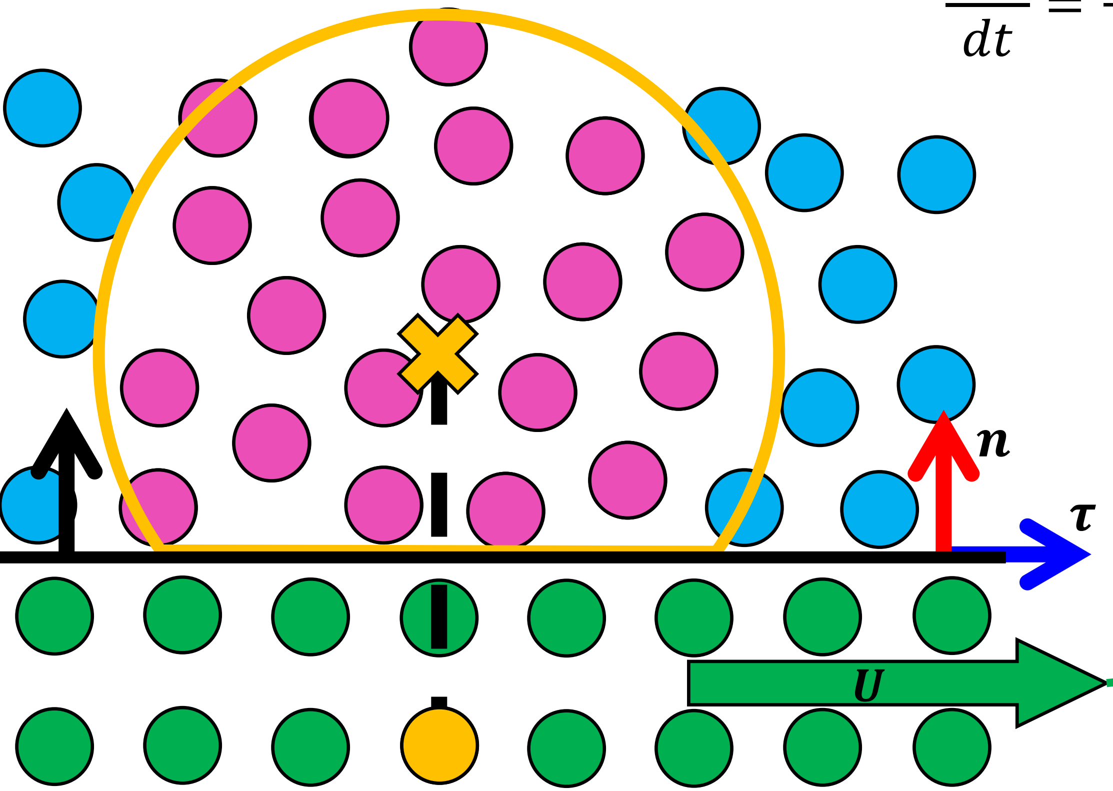
Free slip boundary

$$\boldsymbol{u}_g = \frac{\sum_j \boldsymbol{u}_j V_j W_{gj}}{\sum_j V_j W_{gj}} \qquad \boldsymbol{u}_b = \boldsymbol{u}_g$$

$$\frac{d\boldsymbol{u}_i}{dt} = -\sum_j m_j \left(\frac{P_i + P_j}{\rho_i\,\rho_j}\right)\nabla W_{ij} + \boxed{\sum_j \frac{4\nu\, m_j\, \boldsymbol{r}_{ij}\cdot\nabla W_{ij}}{(\rho_i+\rho_j)\,(\boldsymbol{r}_{ij}^2 + 0.01h^2)}\boldsymbol{u}_{ij}} + \boldsymbol{g}_i$$

$$\boldsymbol{u}_b\cdot\boldsymbol{\tau} = (\boldsymbol{u}_g) - ((\boldsymbol{u}_g)\cdot\boldsymbol{n})$$

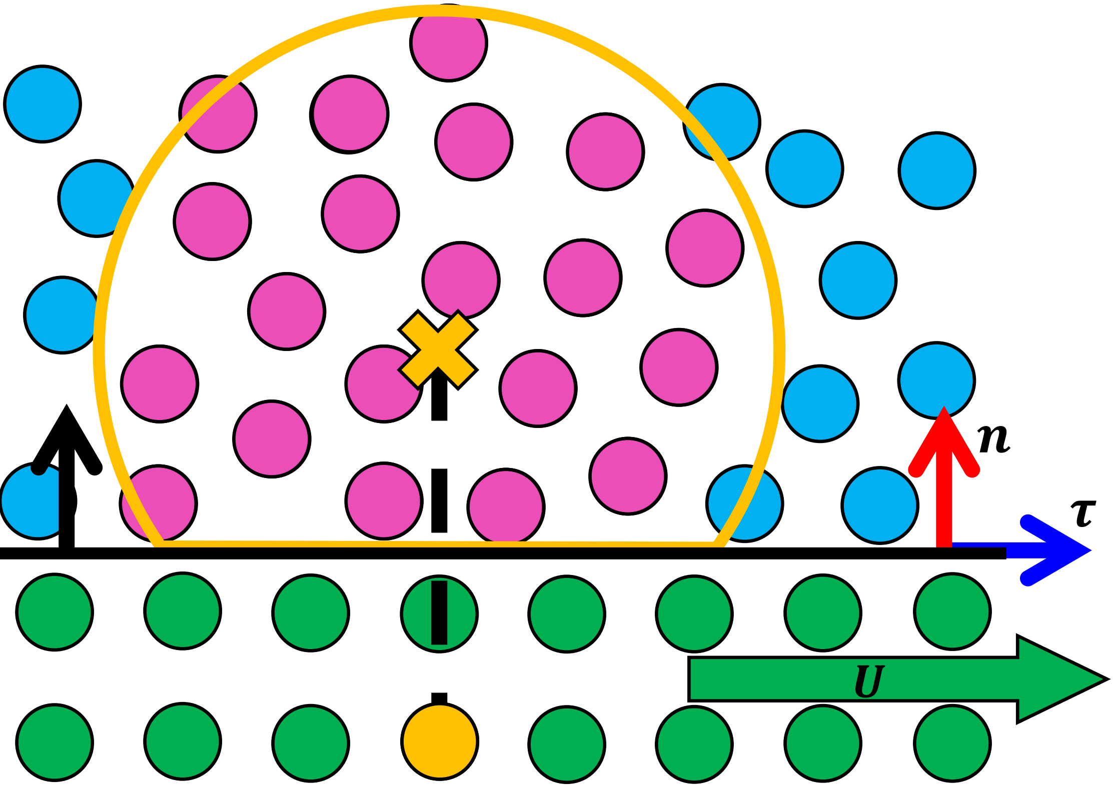
$$\boldsymbol{u}_b\cdot\boldsymbol{n} = (\boldsymbol{u}_g)\cdot\boldsymbol{n}$$



$$\frac{d\rho_i}{dt} = \rho_i \sum_j \frac{m_j}{\rho_j}\boxed{\boldsymbol{u_{ij}}}\cdot\nabla W_{ij} + DDT$$

Boundary conditions in DSPH

mDBC slip mode 11 (NEW)



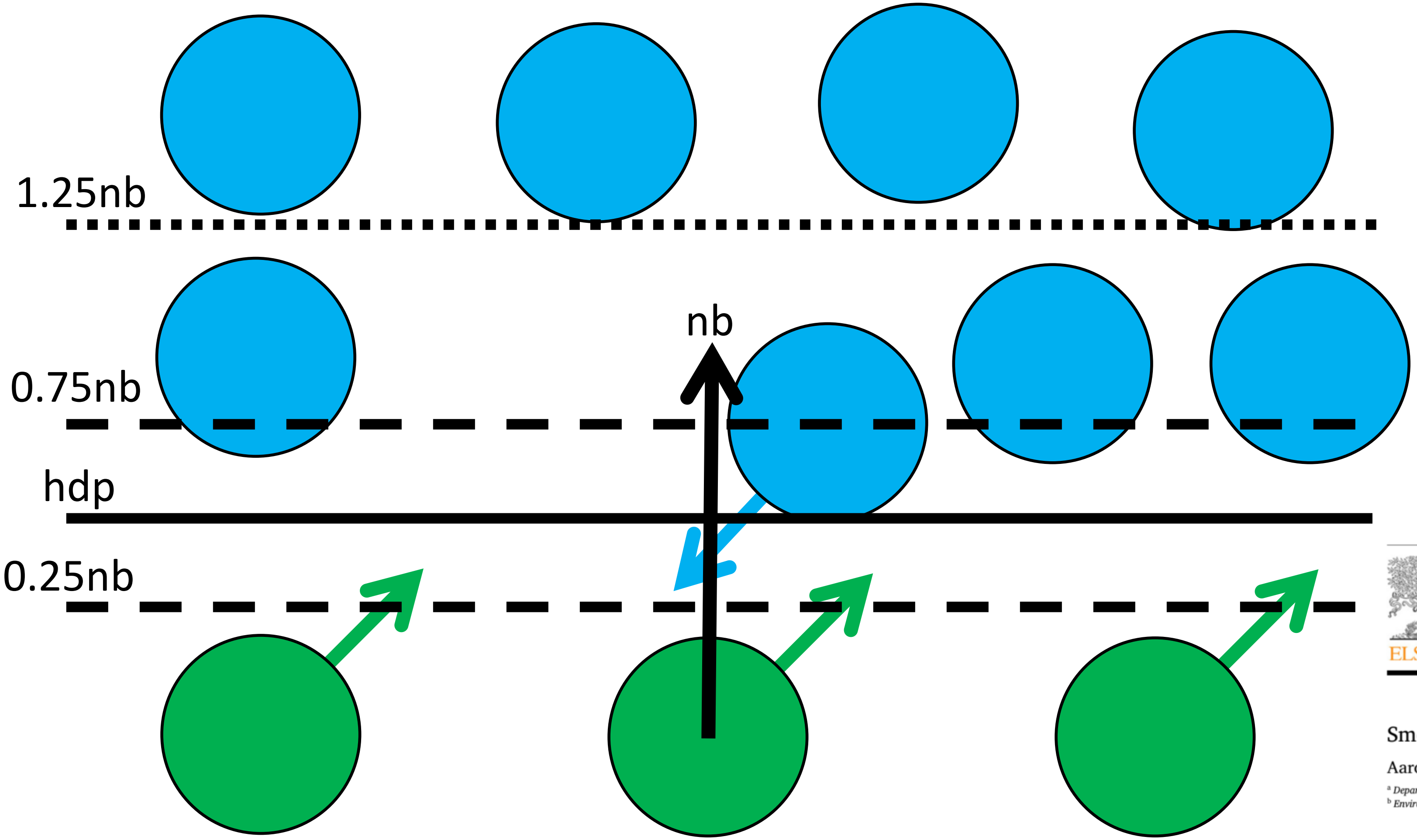
$$\cancel{\boldsymbol{u}_g = \frac{\sum_j \boldsymbol{u}_j V_j W_{gj}}{\sum_j V_j W_{gj}}} \qquad \boldsymbol{u}_b = 0$$

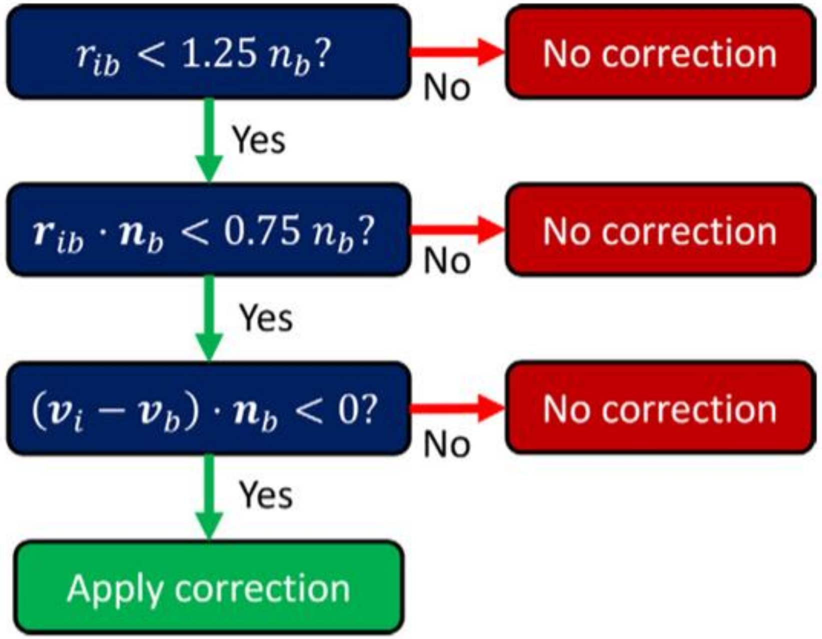
TanVel = 0 boundary

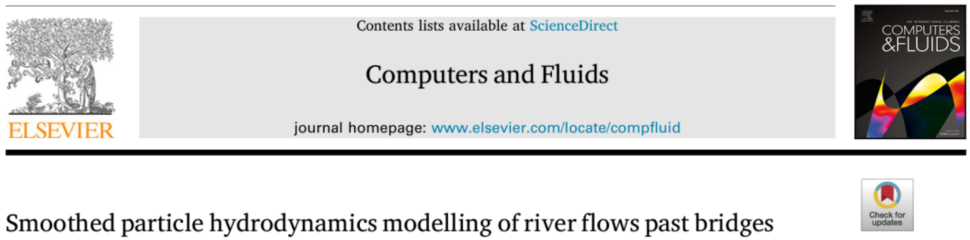
The same as slip mode 1 but with the new density correction

Boundary conditions in DSPH

mDBC no-penetration (NEW)






Computers and Fluids 303 (2025) 106870

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

ELSEVIER

Smoothed particle hydrodynamics modelling of river flows past bridges

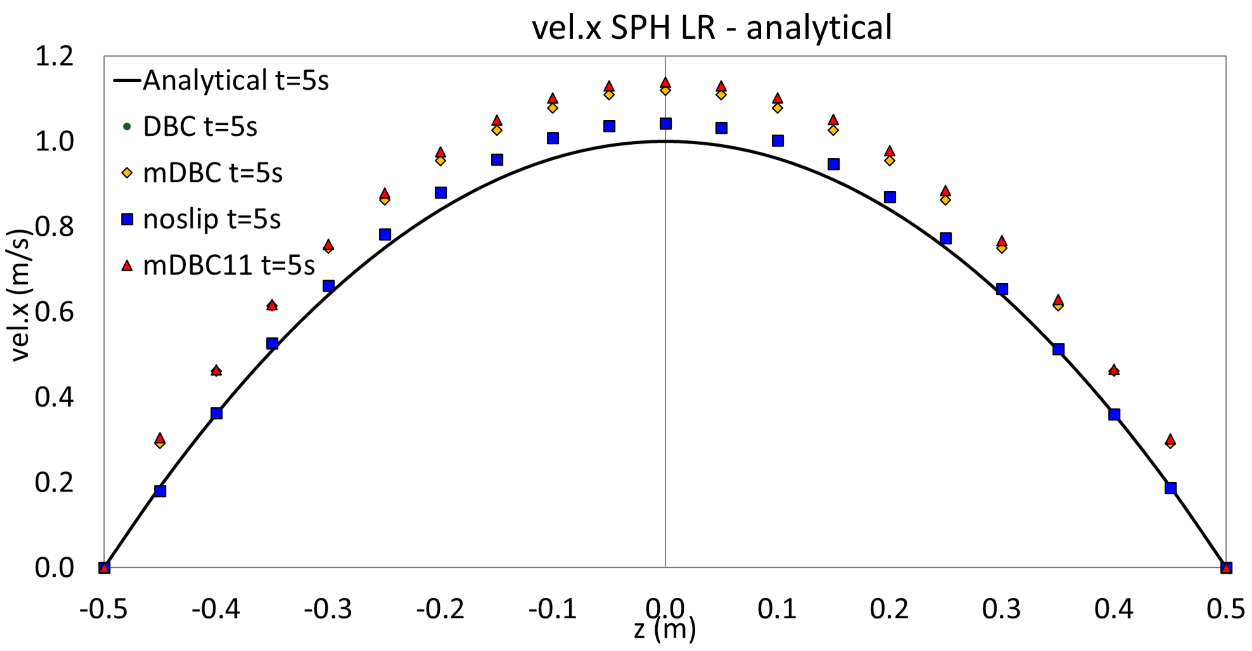
Aaron English [a,*], Renato Vacondio [a], Susanna Dazzi [a], José M. Domínguez [b]

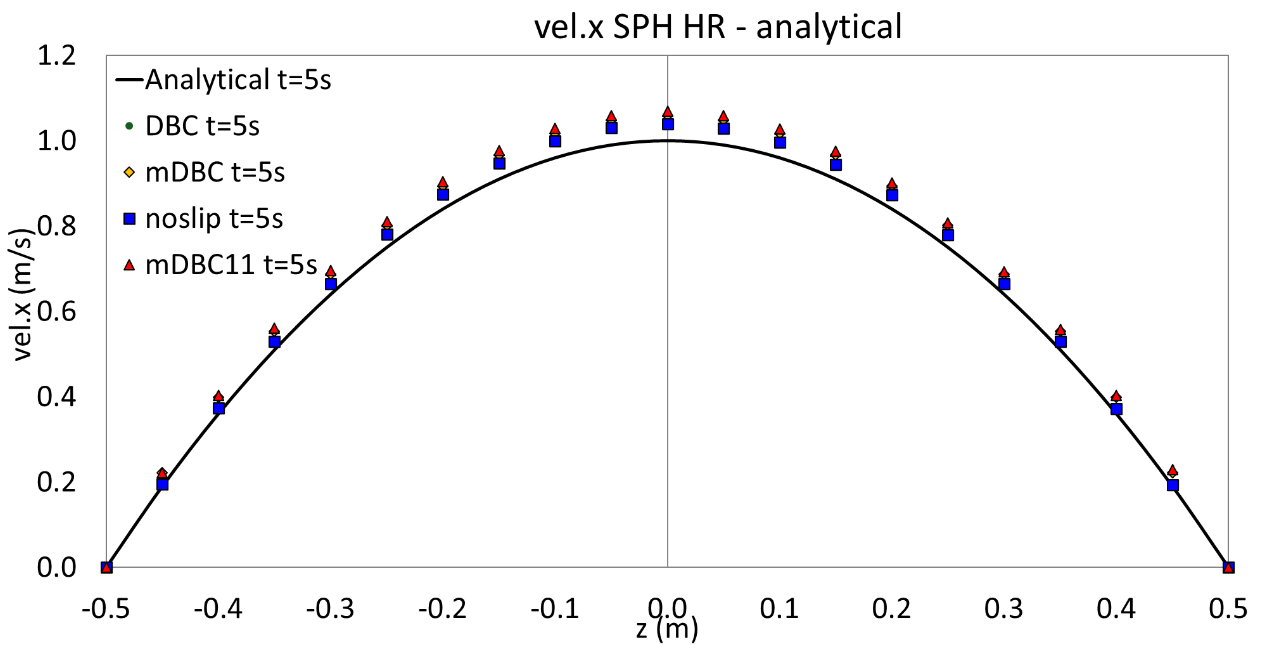
[a] Department of Engineering and Architecture, Università degli Studi di Parma, Parma, Italy
[b] Environmental Physics Laboratory, CIM-UVIGO, Universidade de Vigo, Vigo, Spain

Boundary conditions in DSPH

What is the difference between slip modes?

```
<parameter key="Boundary" value="3" comment="Boundary method 1:DBC, 2:mDBC, 3:MixBC (default=1)" />
<parameter key="SlipMode" value="2" comment="Slip mode for mDBC 1:Old-mDBC vel=0, 11:TanVel=0, 2:No-slip, 3:Free slip (default=1)" />
<parameter key="NoPenetration" value="1" comment="Activate No-Penetration option for TanVel=0, No-slip or Free-slip mDBC 0:Off, 1:On (default=0)" />
```

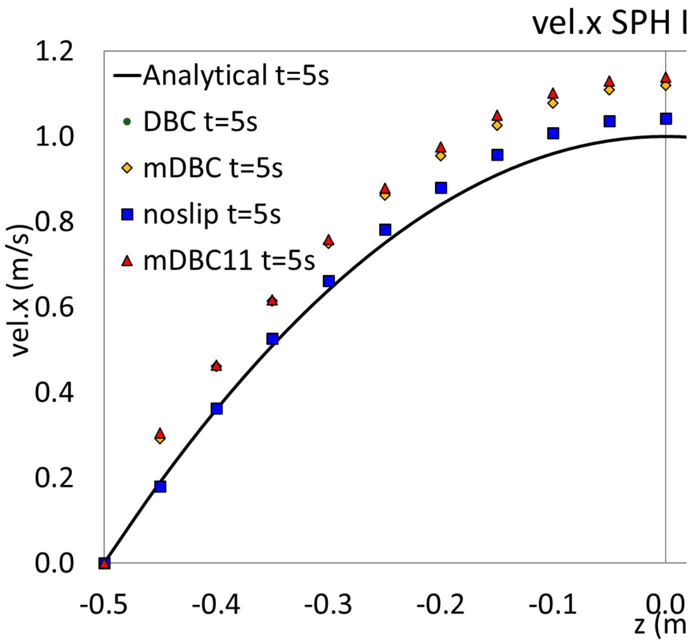


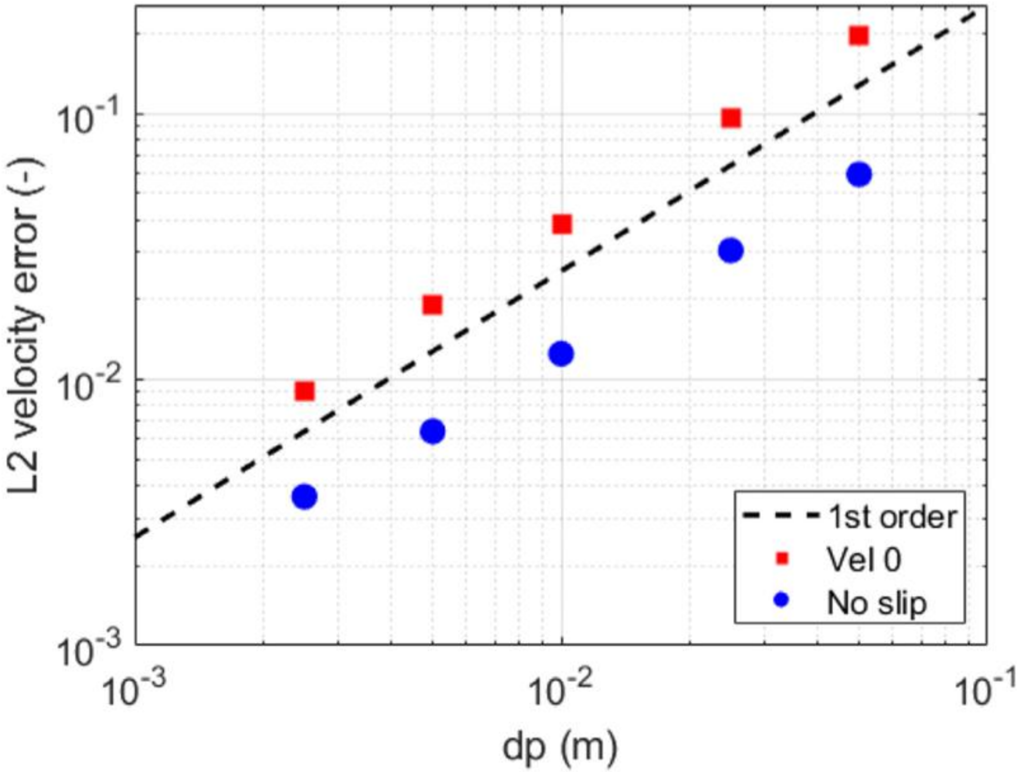


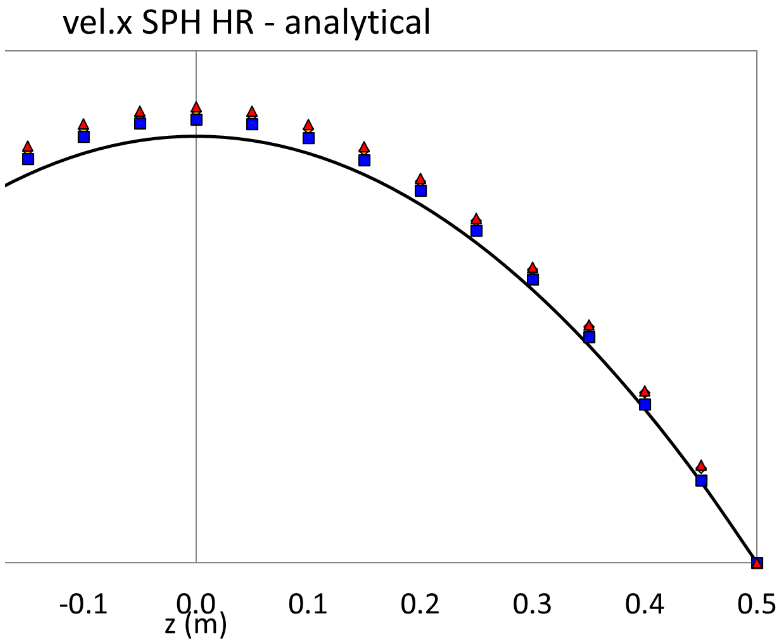
Boundary conditions in DSPH

What is the difference between slip modes?

```
<parameter key="Boundary" value="3" comment="Boundary method 1:DBC, 2:mDBC, 3:MixBC (default=1)" />
<parameter key="SlipMode" value="2" comment="Slip mode for mDBC 1:Old-mDBC vel=0, 11:TanVel=0, 2:No-slip, 3:Free slip (default=1)" />
<parameter key="NoPenetration" value="1" comment="Activate No-Penetration option for TanVel=0, No-slip or Free-slip mDBC 0:Off, 1:On (default=0)" />
```

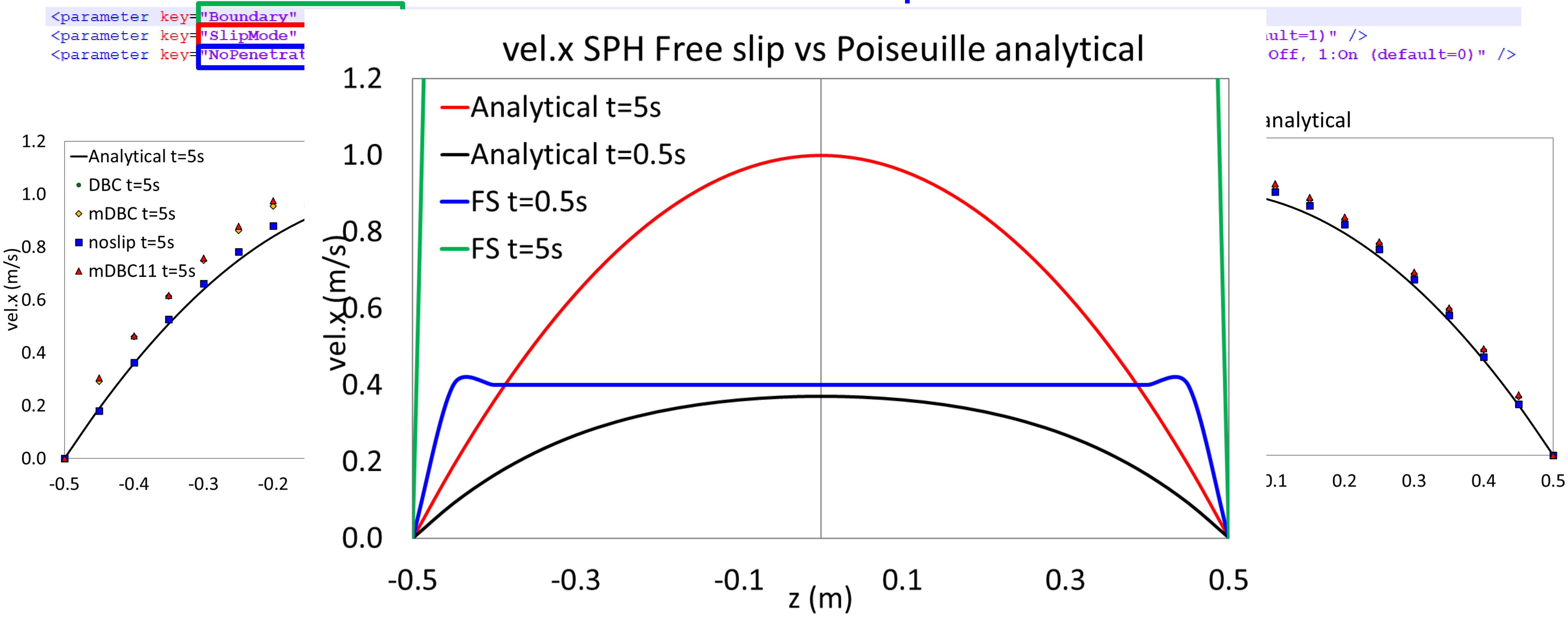






Boundary conditions in DSPH

What is the difference between slip modes?



Mixed boundary conditions (NEW)

What about boundary option 3?

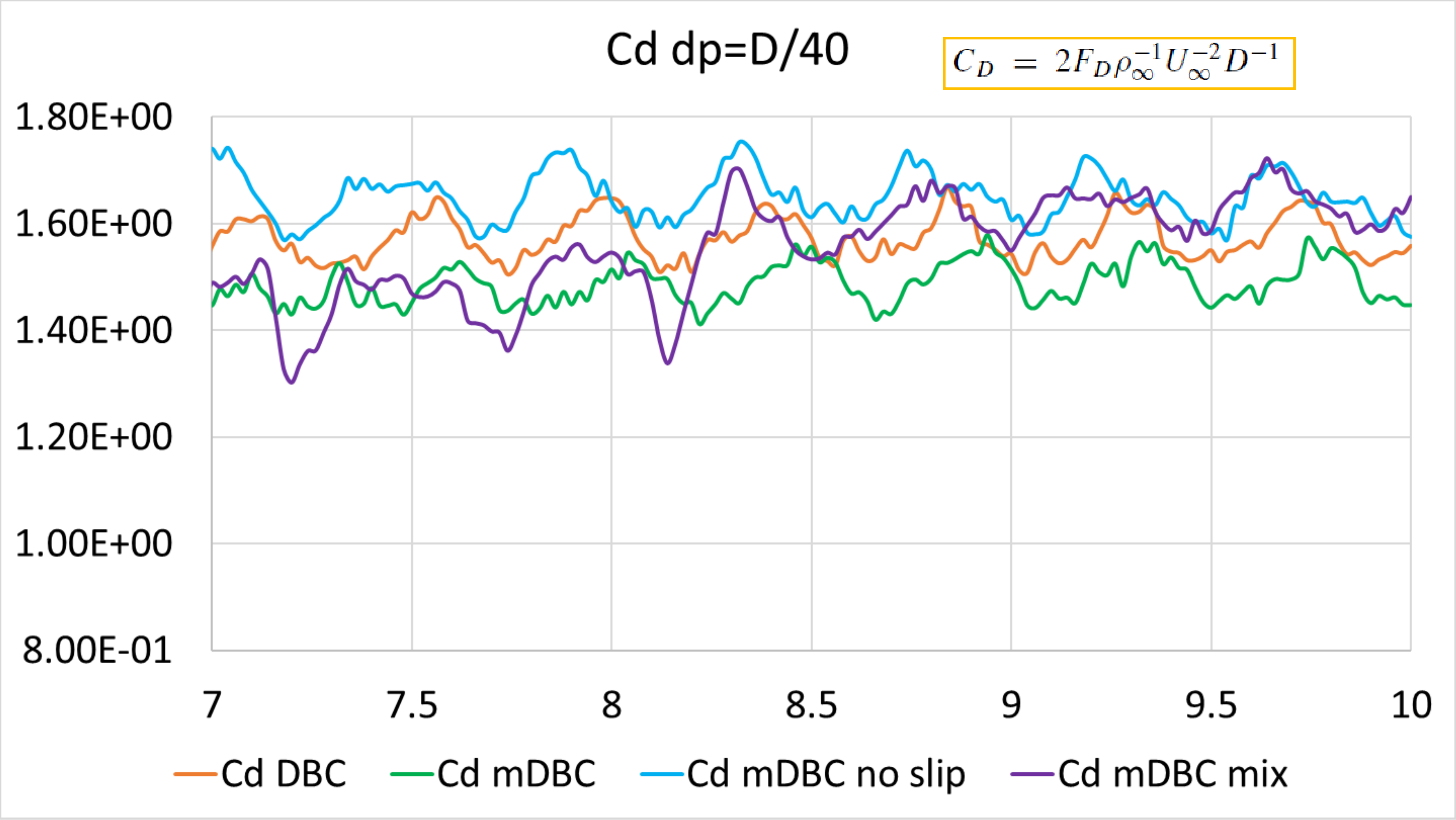
```
<parameter key="Boundary" value="3" comment="Boundary method 1:DBC, 2:mDBC, 3:MixBC (default=1)" />
<parameter key="SlipMode" value="2" comment="Slip mode for mDBC 1:Old-mDBC vel=0, 11:TanVel=0, 2:No-slip, 3:Free slip (default=1)" />
<parameter key="NoPenetration" value="1" comment="Activate No-Penetration option for TanVel=0, No-slip or Free-slip mDBC 0:Off, 1:On (default=0)" />

<execution>
    <special>
        <mixboundary>
            <!-- boundtype: DBC, MDBC_V0 (Old mDBC), MDBC_T0 (mDBC2 tanvel=0), MDBC_NS (mDBC2 No-slip), MDBC_FR (mDBC2 Free slip) -->
            <!-- nopenetration: 1/0 Activate No-Penetration option for TanVel=0, No-slip or Free-slip mDBC -->
            <!-- mkbound: mkbound value or list of values (2,5,7-8) -->
            <default boundtype="MDBC_NS" nopenetration="0" />
            <bcmk mkbound="0" boundtype="MDBC_NS" nopenetration="1" />
            <bcmk mkbound="1,2" boundtype="MDBC_FR" nopenetration="1" />
            <bcmk mkbound="3-5" boundtype="MDBC_T0" nopenetration="1" />
            <bcmk mkbound="6,7,9" boundtype="DBC" nopenetration="0" />
            <bcmk mkbound="8" boundtype="MDBC_V0" nopenetration="0" />
            <bcmk mkbound="10" boundtype="MDBC_NS" nopenetration="0" />
            <bcmk mkbound="11-12" boundtype="MDBC_FR" nopenetration="0" />
        </mixboundary>
    </special>
```

Still need to define normals for all boundaries even DBC bounds

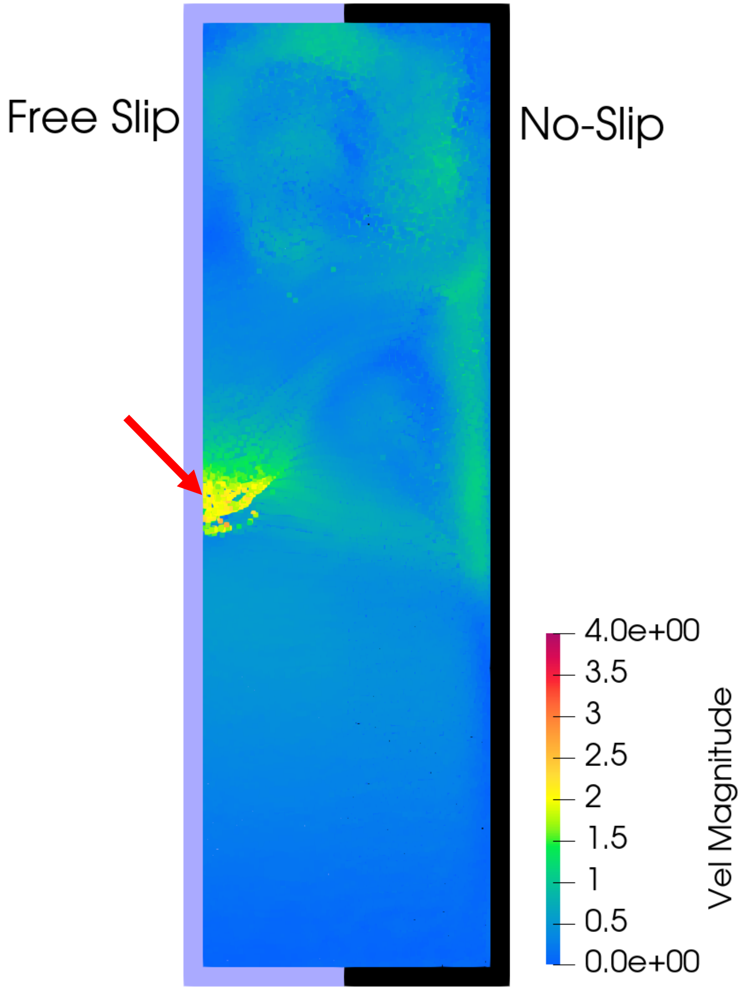
Mixed boundary conditions

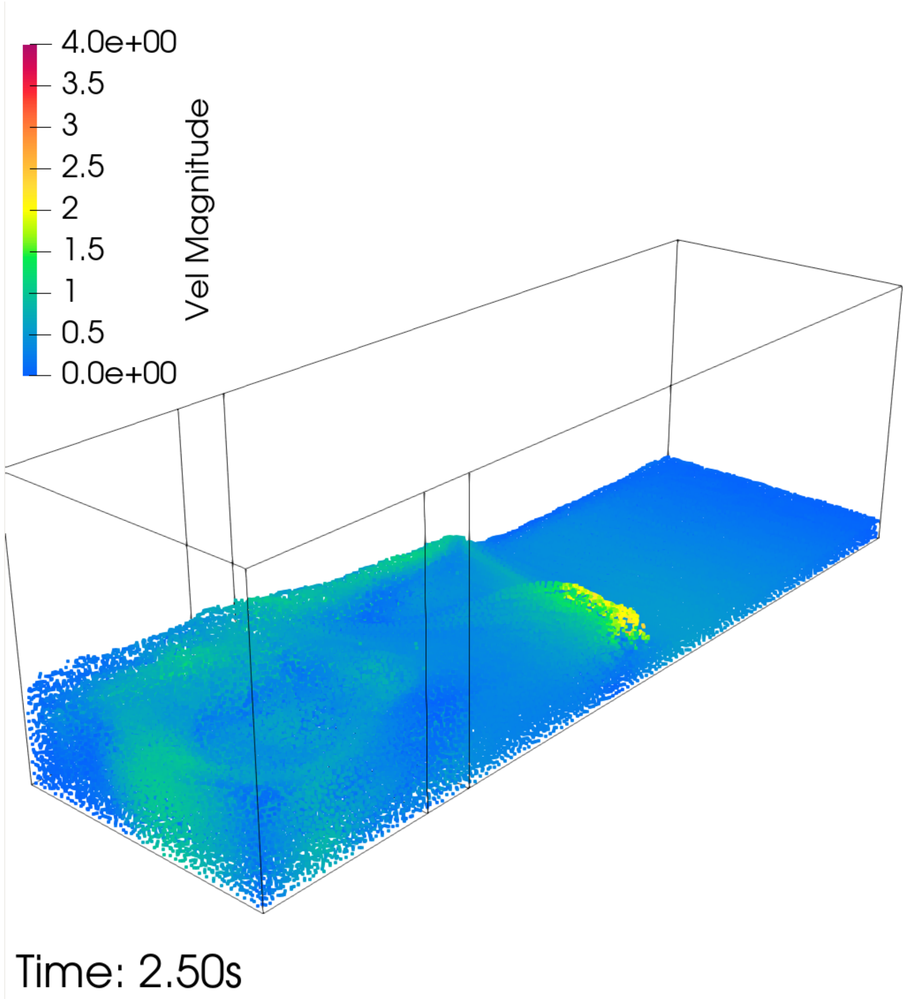
Examples – Flow past a cylinder



Mixed boundary conditions

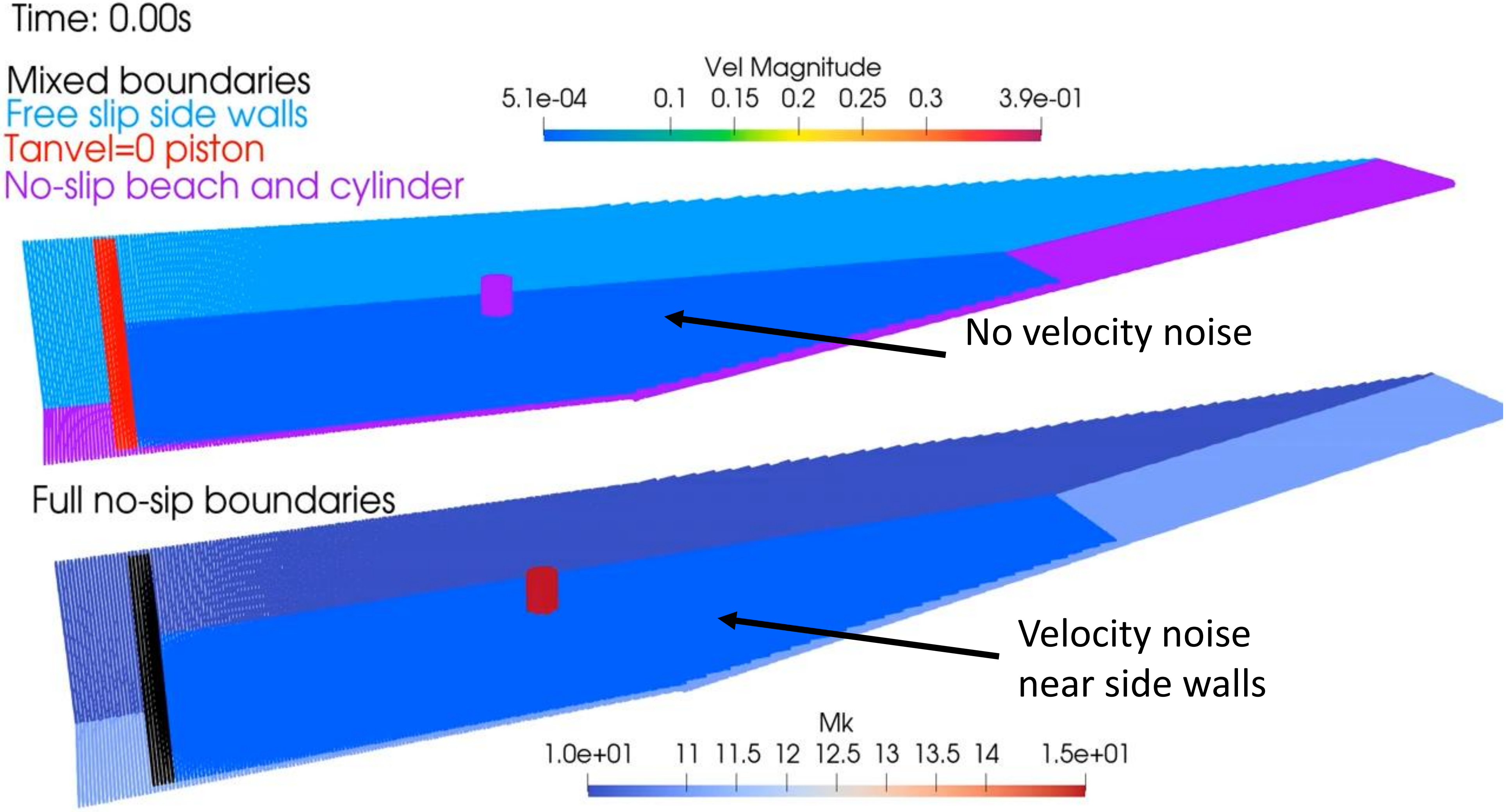
Examples – 3D dambreak





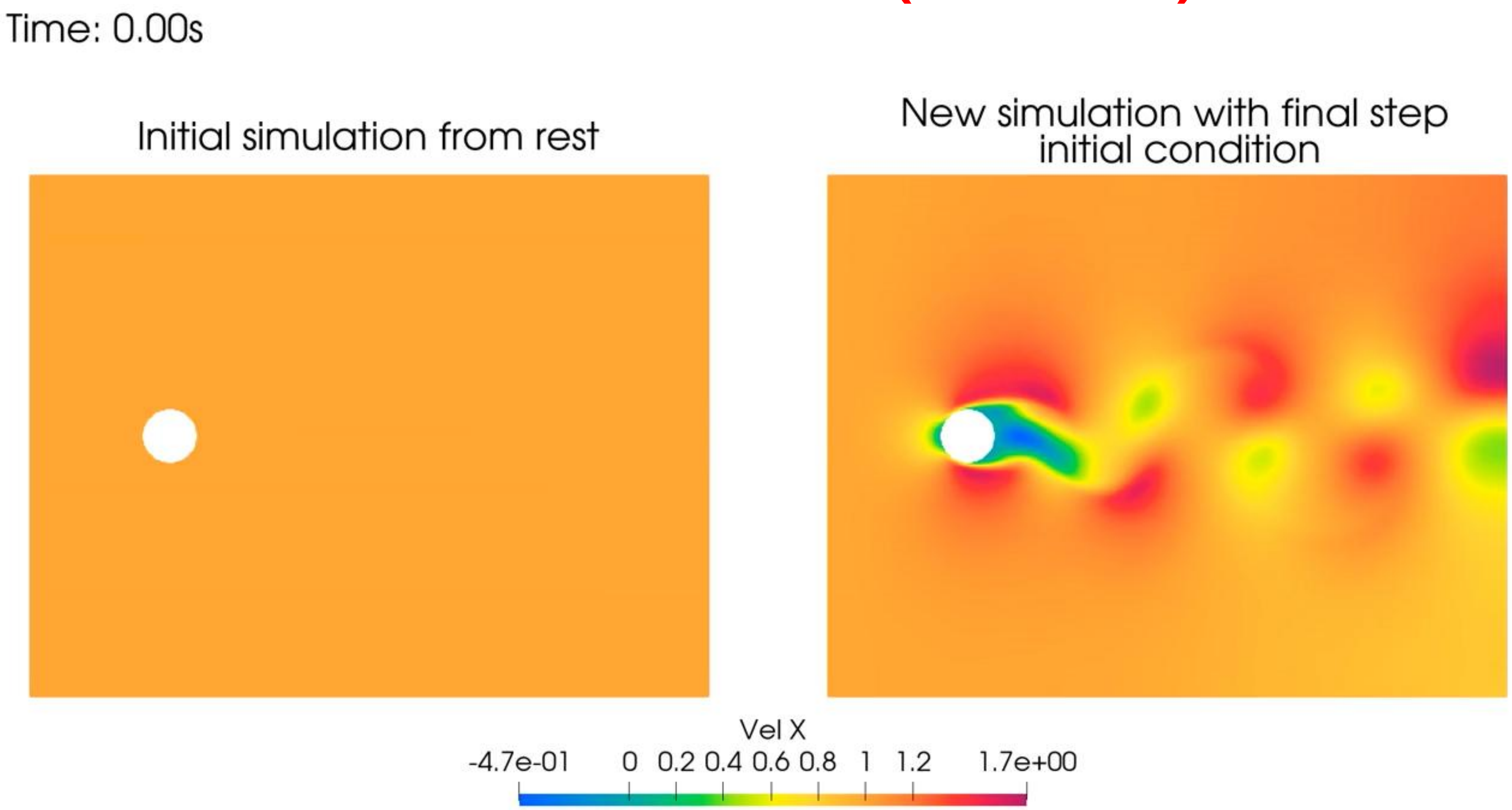
Mixed boundary conditions

Examples – waves on a beach with an obstacle

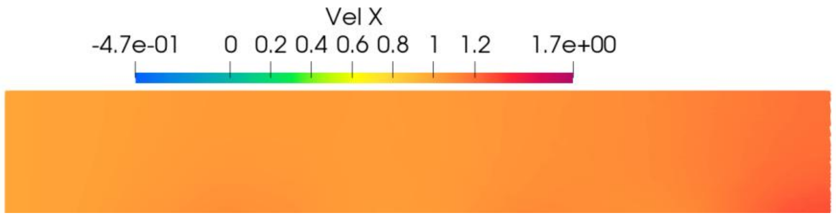


Defining initial conditions from previous simulations (NEW)

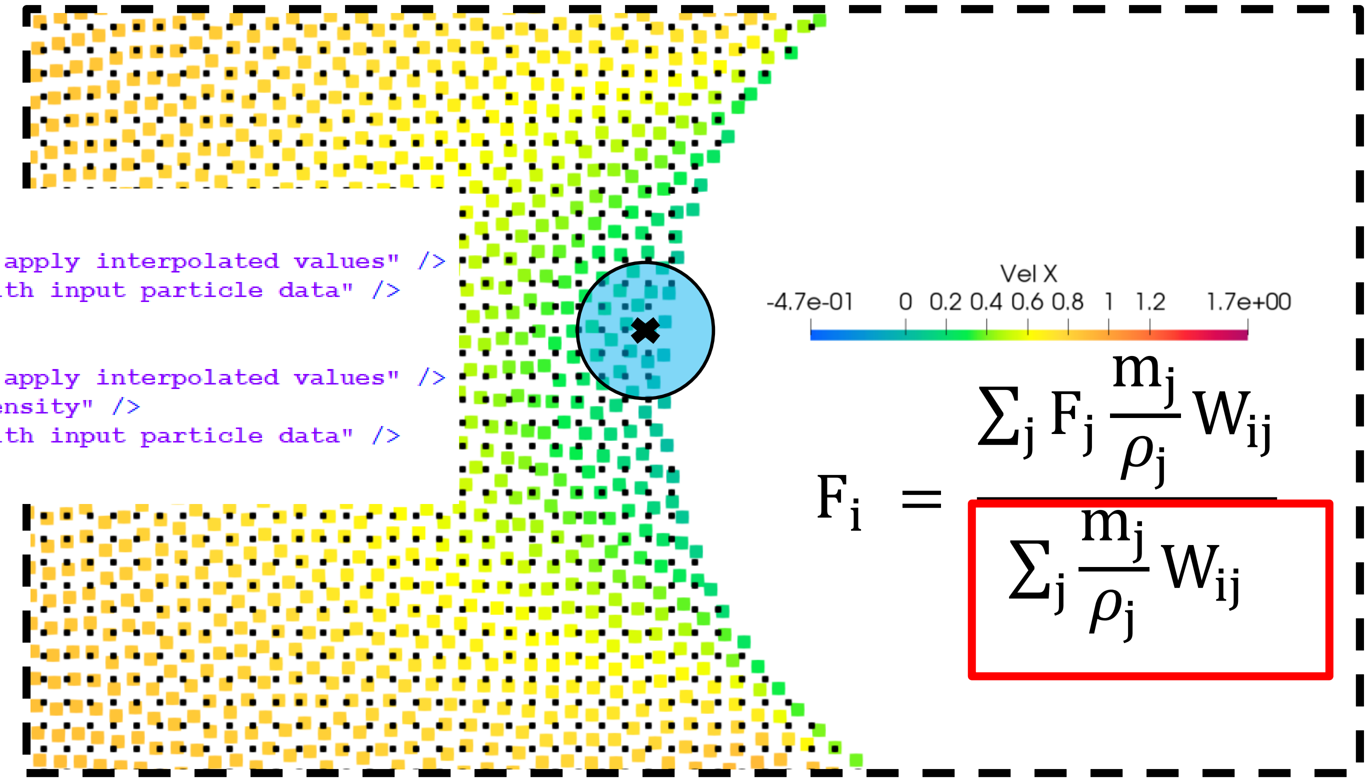
Takes a long time to reach steady state and even longer for higher resolution



Defining initial conditions from previous simulations – set properties




```xml
<initials>
    <velpart mkfluid="0">
        <minratio v="0.5" comment="Minimum fluid ratio (volume) to apply interpolated values" />
        <partfile v="Initial2D/Part_0300.bi4" comment="PART file with input particle data" />
    </velpart>
    <densitypart mkfluid="0">
        <minratio v="0.5" comment="Minimum fluid ratio (volume) to apply interpolated values" />
        <fitpress v="true" comment="Fits the pressure instead of density" />
        <partfile v="Initial2D/Part_0300.bi4" comment="PART file with input particle data" />
    </densitypart>
</initials>
```



$$F_i = \frac{\sum_j F_j \frac{m_j}{\rho_j} W_{ij}}{\boxed{\sum_j \frac{m_j}{\rho_j} W_{ij}}}$$

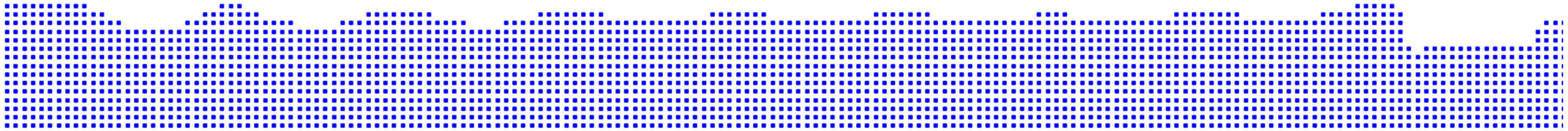
Can initialise fluid density, pressure, velocity and surface elevation

Defining initial conditions from previous simulations – set free surface

```xml
<!-- Water  up stream -->
<setmkfluid mk="0" />
<setboxlimitmode mode="full" />
<fillbox x="-0.1" y="0.25" z="0.005">
    <modefill>void</modefill>
    <point  x="-1.475" y="0.0025"z="0.0" />
    <size  x="3.4" y="0.495" z="0.15" />
</fillbox>
```

Creates fillbox to fill domain with fluid particles

New initial condition



Previous final free surface profile

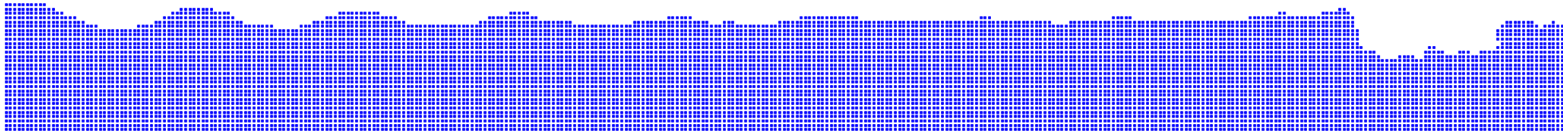
Defining initial conditions from previous simulations – change dp

```
    <!-- Water  up stream -->
    <setmkfluid mk="0" />
    <setboxlimitmode mode="full" />
    <fillbox x="-0.1" y="0.25" z="0.005">
        <modefill>void</modefill>
        <point  x="-1.475" y="0.0025"z="0.0" />
        <size  x="3.4" y="0.495" z="0.15" />
  </fillbox>
  <!-- Water from bi4 -->
    <setmkvoid mk="0" />
    <redraw mkfluid="0">
        <filterfluid maxratio="0.5" file="Initial2D/Part_0300.bi4" />
    </redraw>
    <_shapeout file="" />
</mainlist>
```

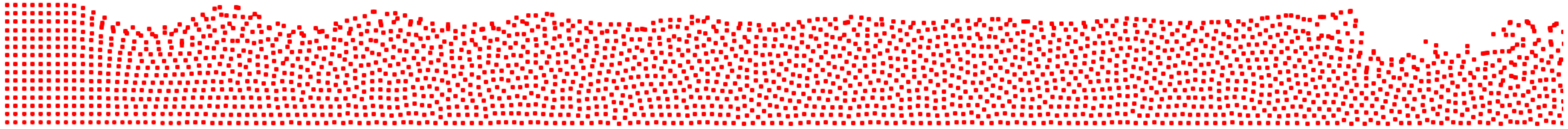
Creates fillbox to fill domain with fluid particles

Redraws freesurface and removes fluid particles of chosen mkfluid according to previous .bi4 data

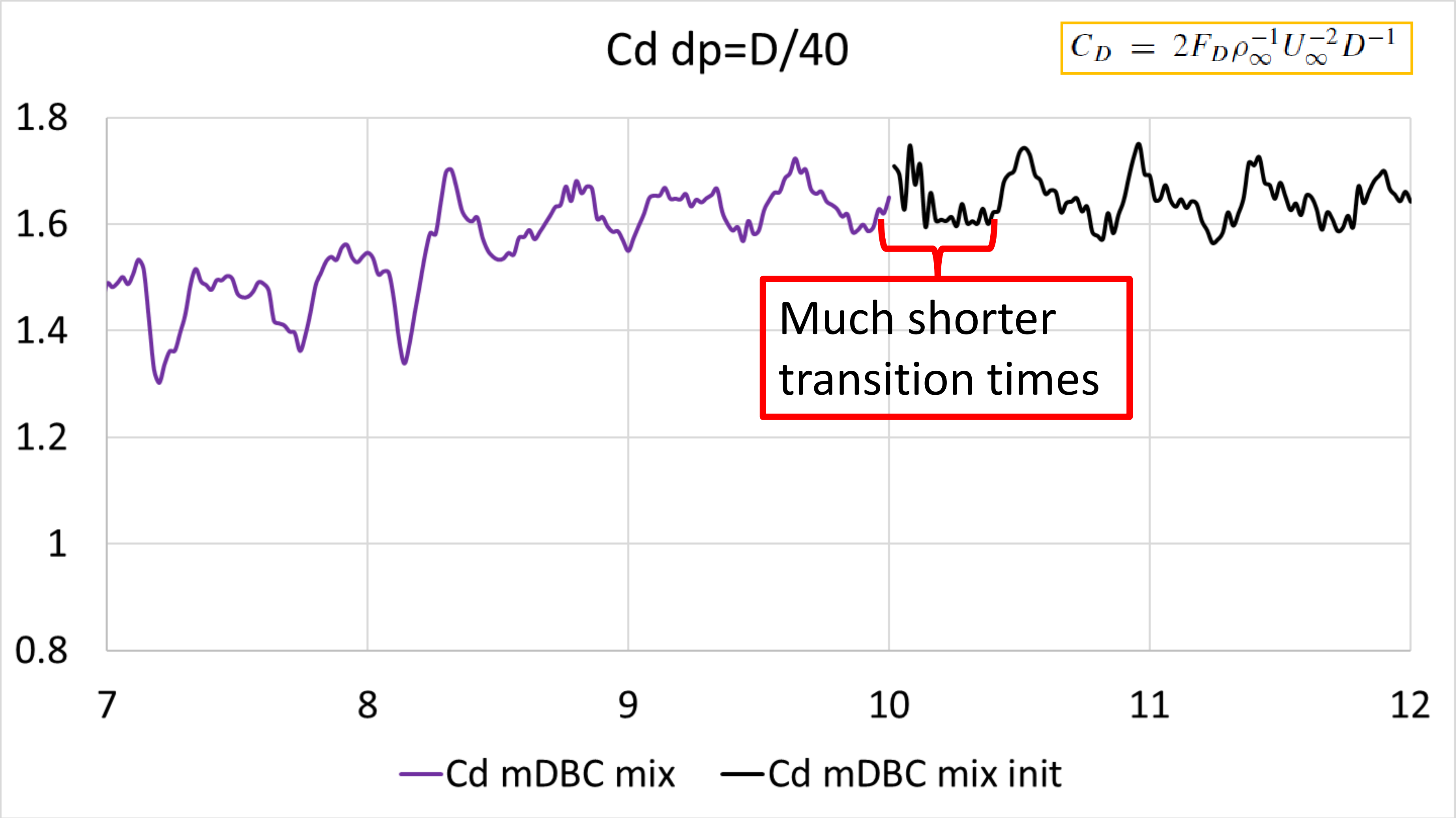
New finer resolution



Previous course resolution



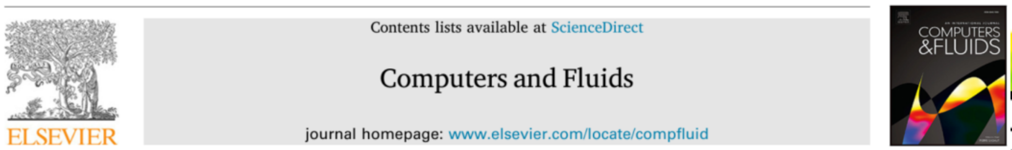
Defining initial conditions from previous simulations - examples



Defining initial c
previous simula

Time: 0.00s

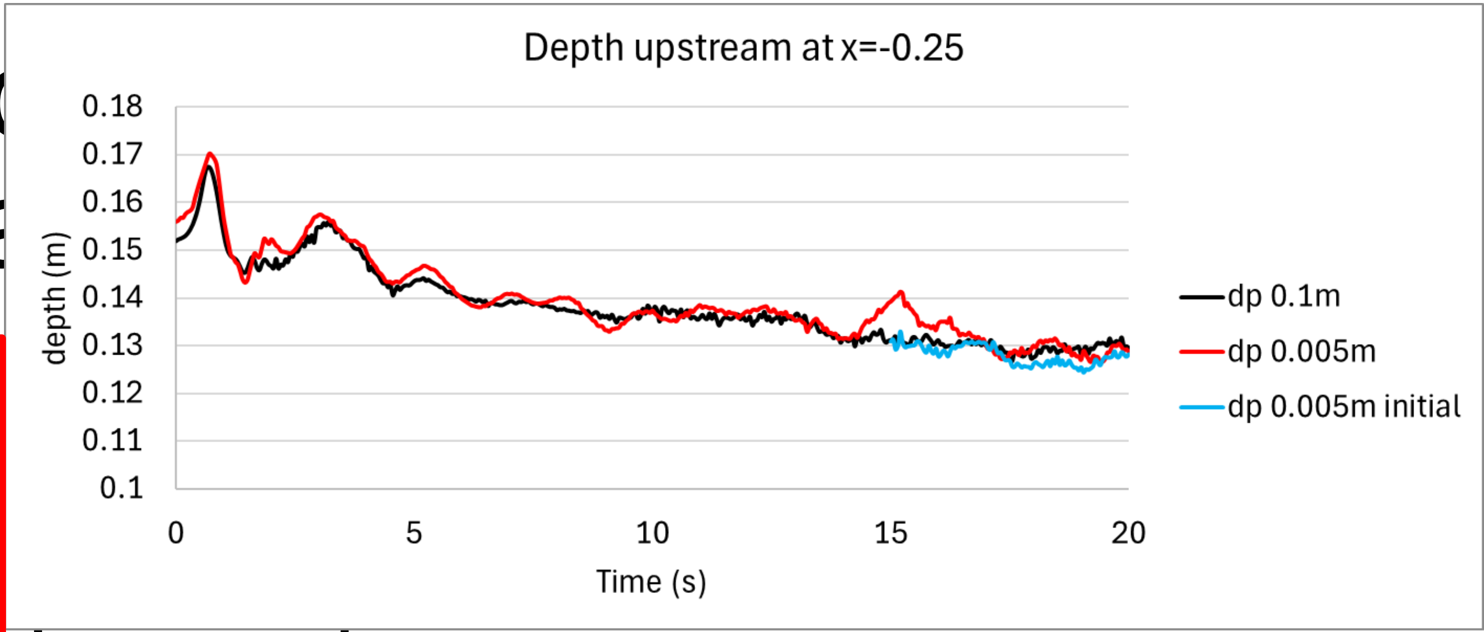
Run times on RTX 4090:
Fine resolution 1535 secs (24,357 part)
Coarse resolution 586 secs (6,678 parts)
Initialised sim 600 secs (20,823 parts)
Coarse + init combo 1186 secs (77%)

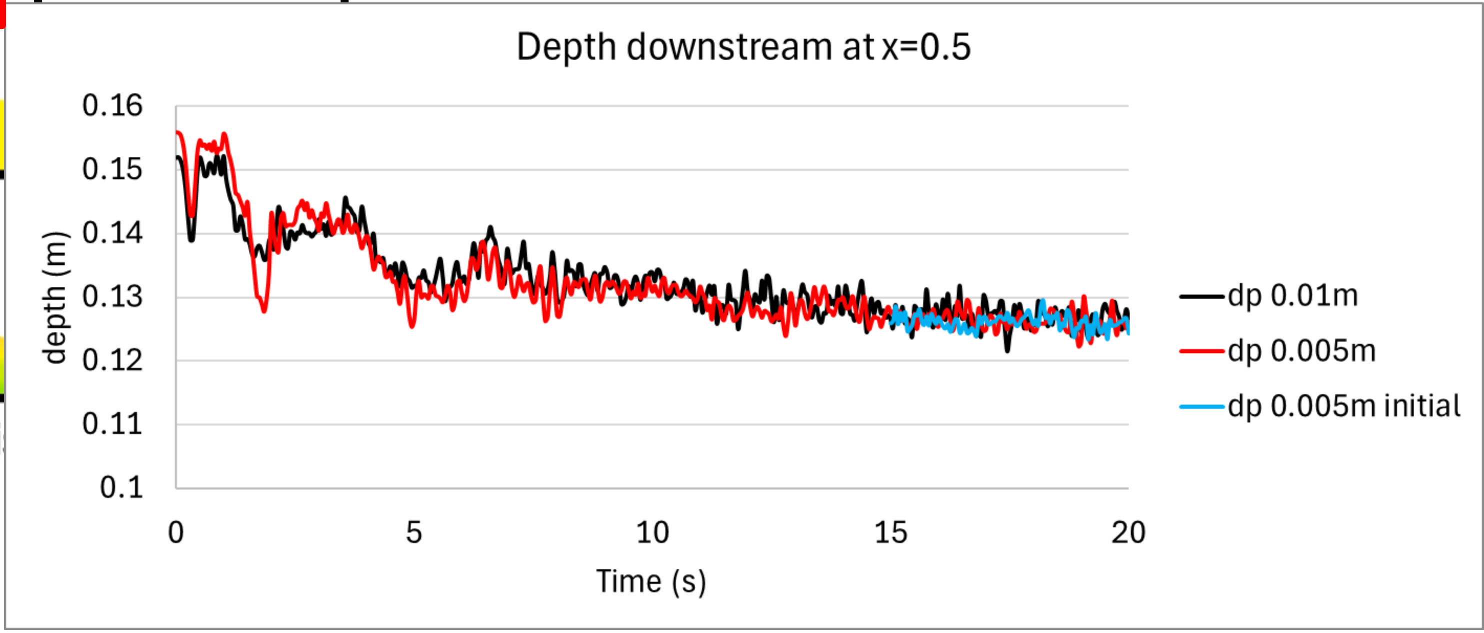


Computers and Fluids 303 (2025) 106870
Contents lists available at ScienceDirect
Computers and Fluids
journal homepage: www.elsevier.com/locate/compfluid

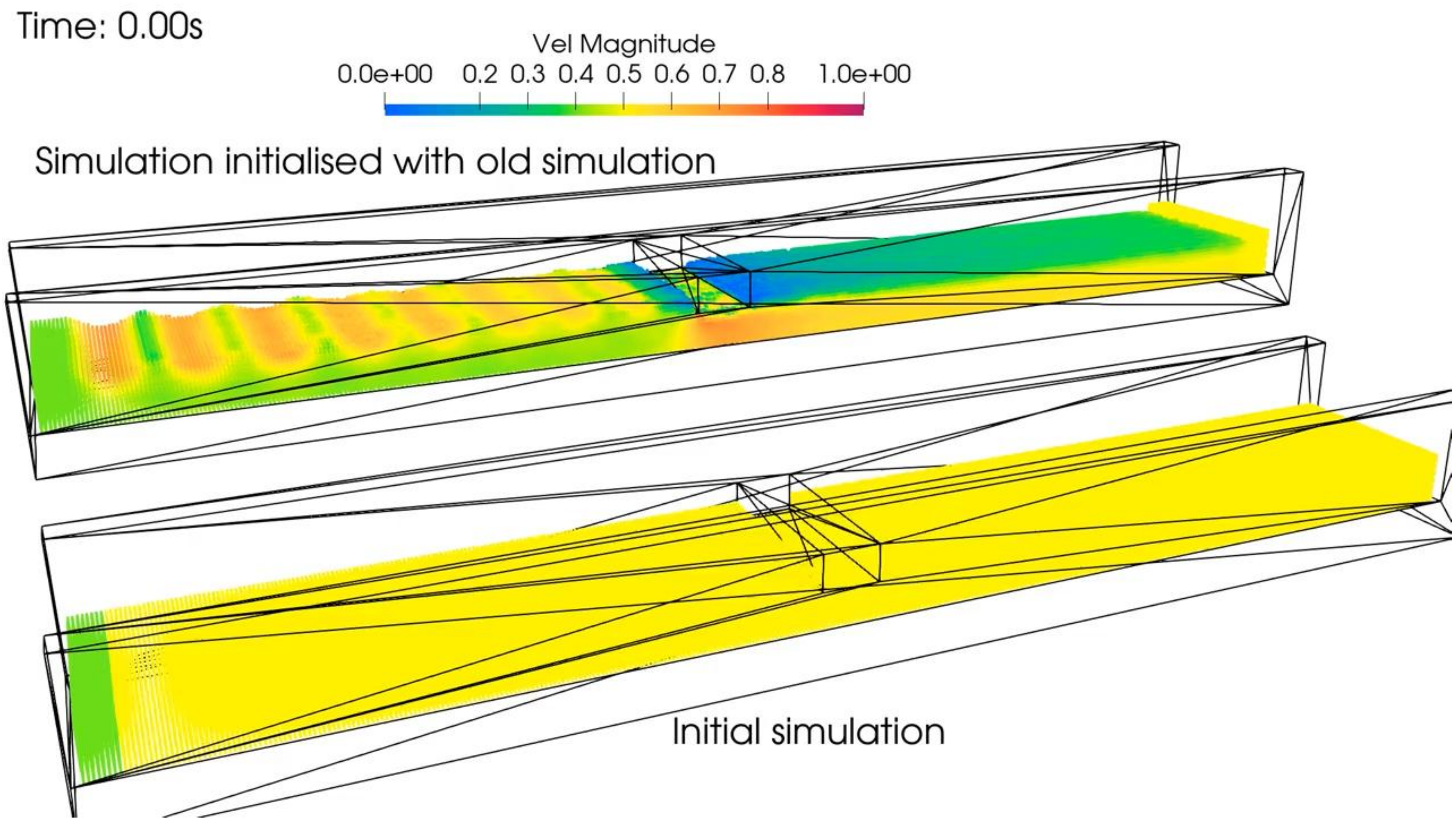
Smoothed particle hydrodynamics modelling of river flows past bridges

Aaron English [a,*], Renato Vacondio [a], Susanna Dazzi [a], José M. Domínguez [b]
[a] Department of Engineering and Architecture, Università degli Studi di Parma, Parma, Italy
[b] Environmental Physics Laboratory, CIM-UVIGO, Universidade de Vigo, Vigo, Spain

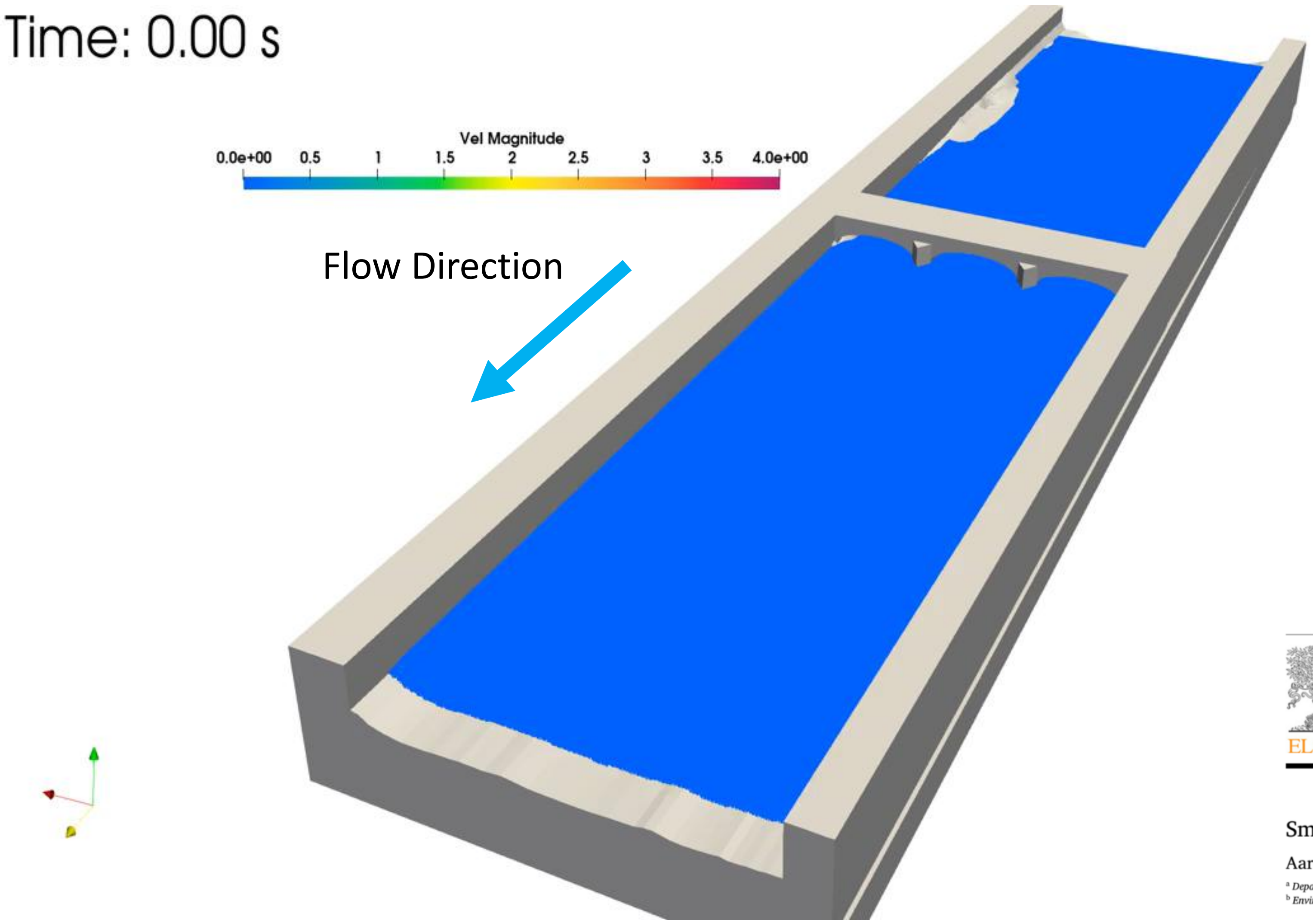




Defining initial conditions from previous simulations 3D (NEW)



Large scale example

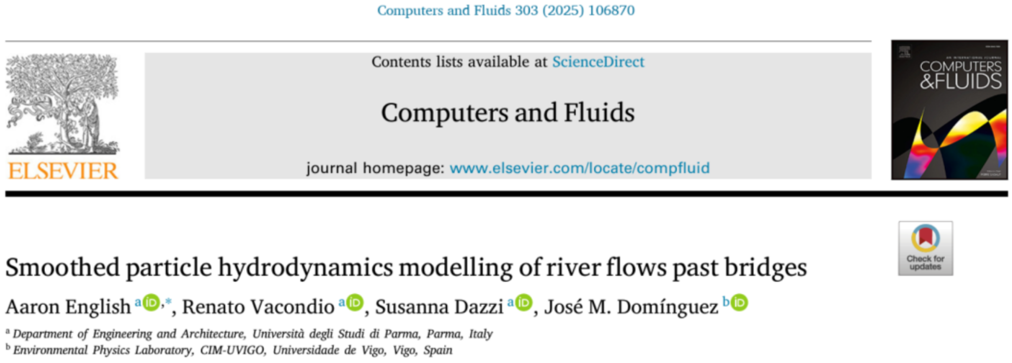


All run on NVIDIA RTX 4090
Arno test case Q=909m^3/s

Coarse resolution 8,730,146 particles 29.4 h to simulate 600 s, and reach steady state

The finer resolution 57,433,510 particles 167.7 h (one week) to simulate 200 s, restarting from the coarser steady state

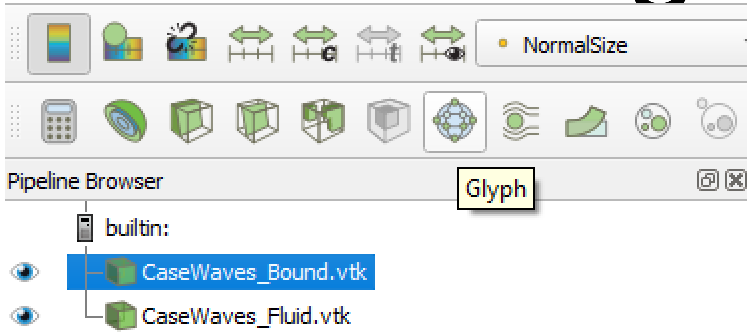
Completed in 40% of the time of a full high res simulation

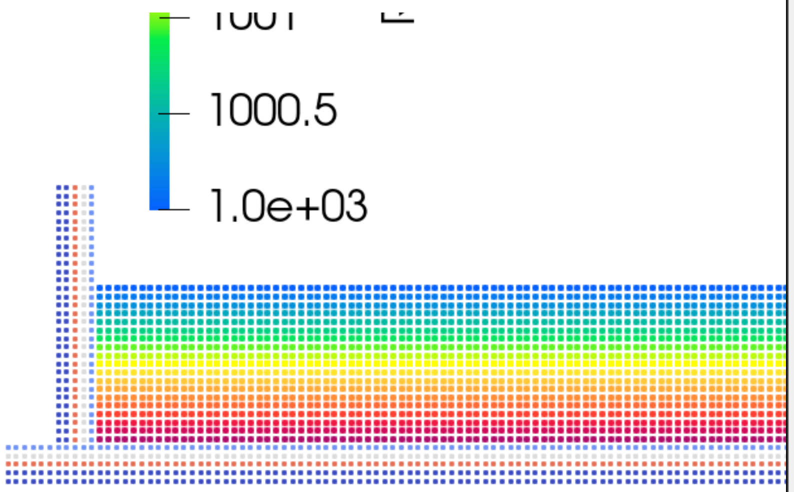

Computers and Fluids 303 (2025) 106870
Contents lists available at ScienceDirect
Computers and Fluids
journal homepage: www.elsevier.com/locate/compfluid

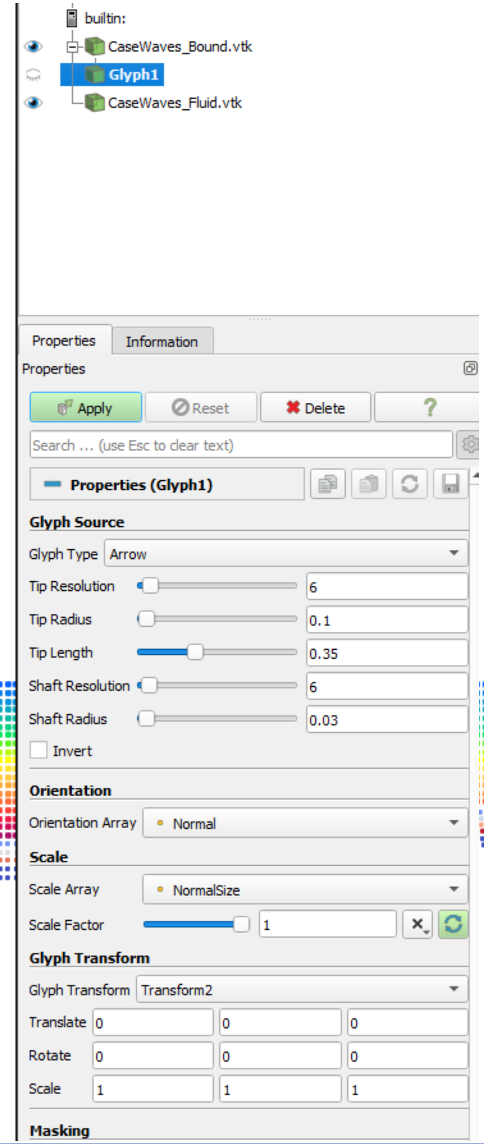
Smoothed particle hydrodynamics modelling of river flows past bridges

Aaron English [a,*], Renato Vacondio [a], Susanna Dazzi [a], José M. Domínguez [b]
[a] Department of Engineering and Architecture, Università degli Studi di Parma, Parma, Italy
[b] Environmental Physics Laboratory, CIM-UVIGO, Universidade de Vigo, Vigo, Spain

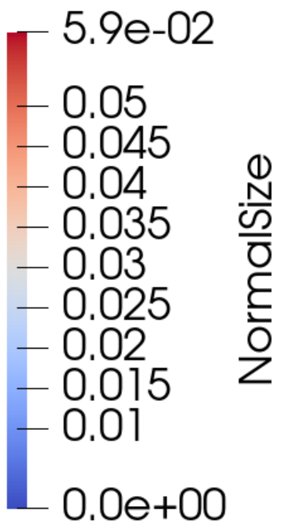
Defining and checking normals

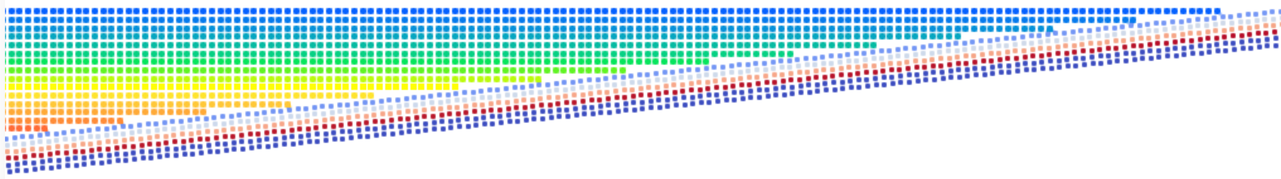




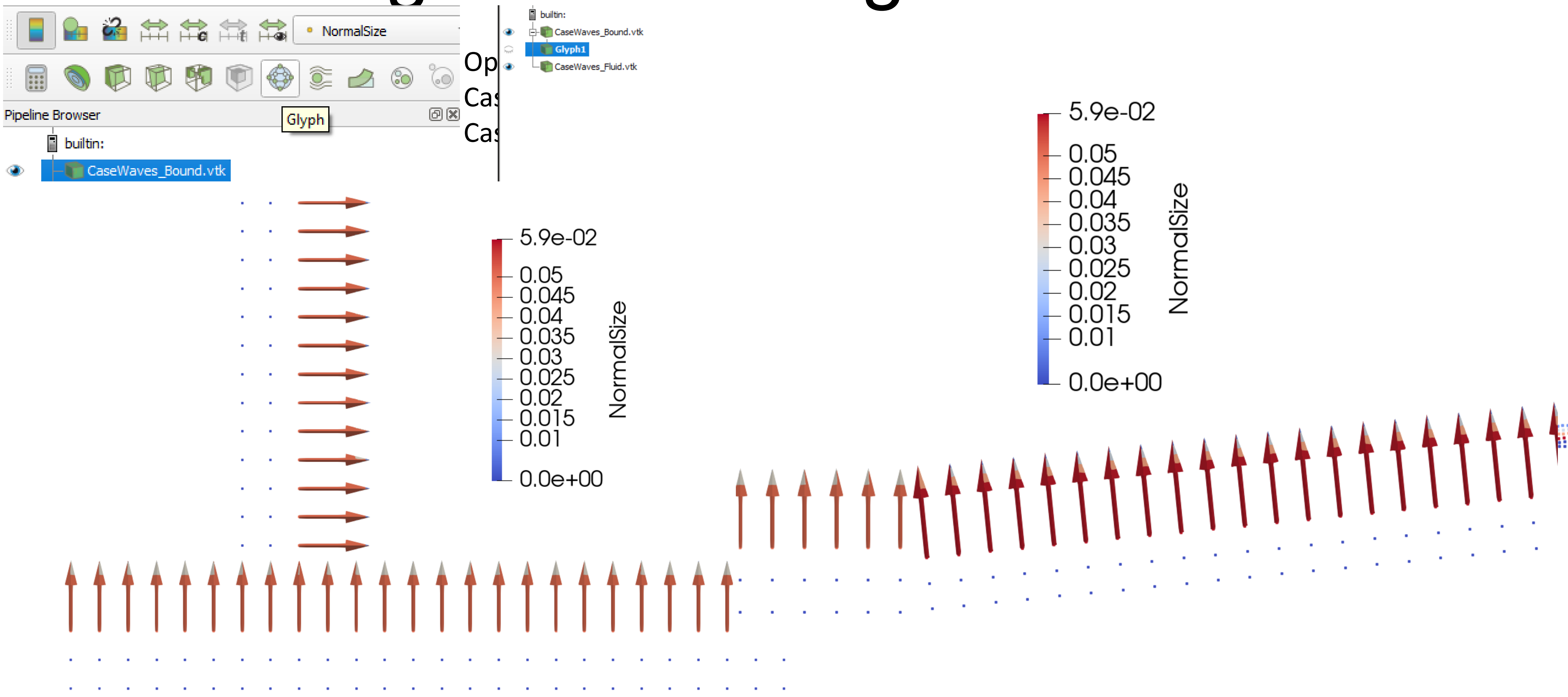


Open the files CaseWaves_bound.vtk and CaseWaves_Fluid.vtk in paraview

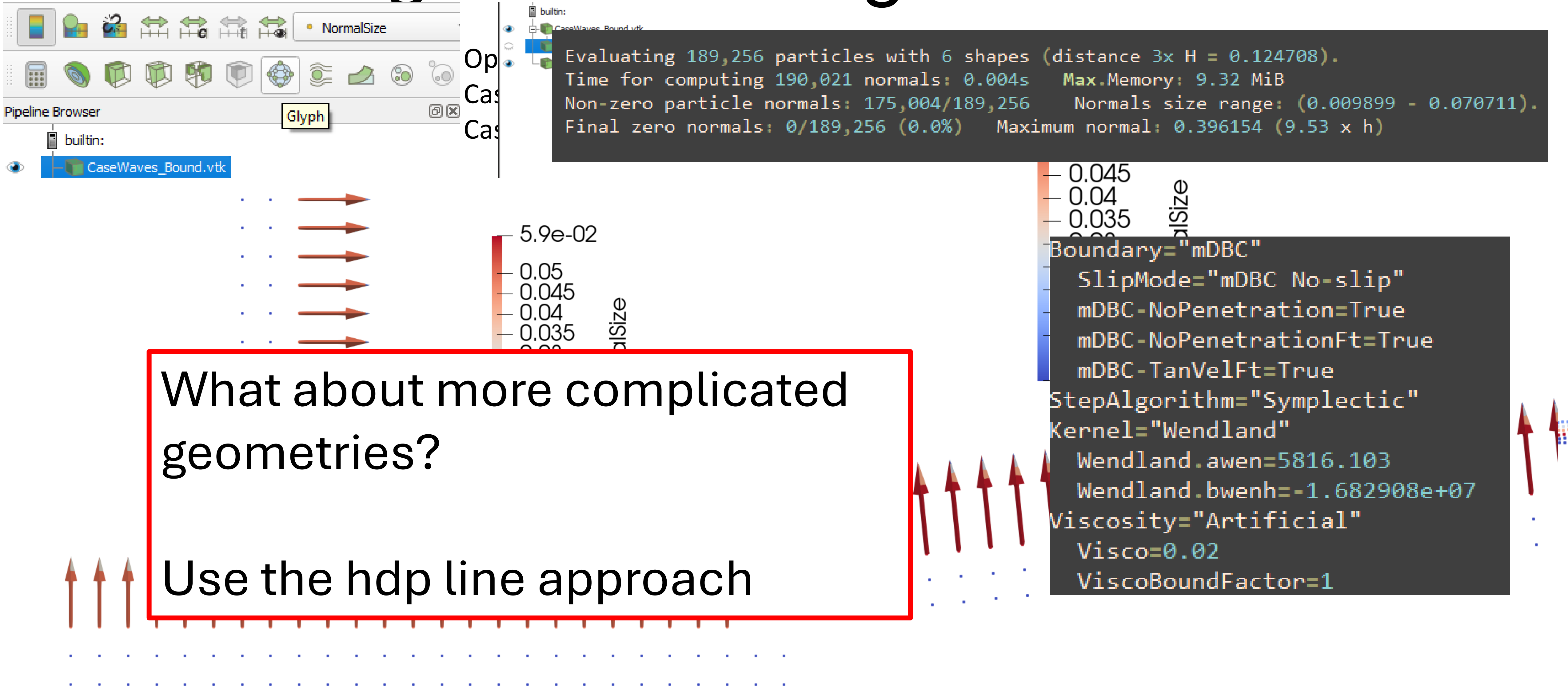




Defining and checking normals



Defining and checking normals



What about more complicated geometries?

Use the hdp line approach

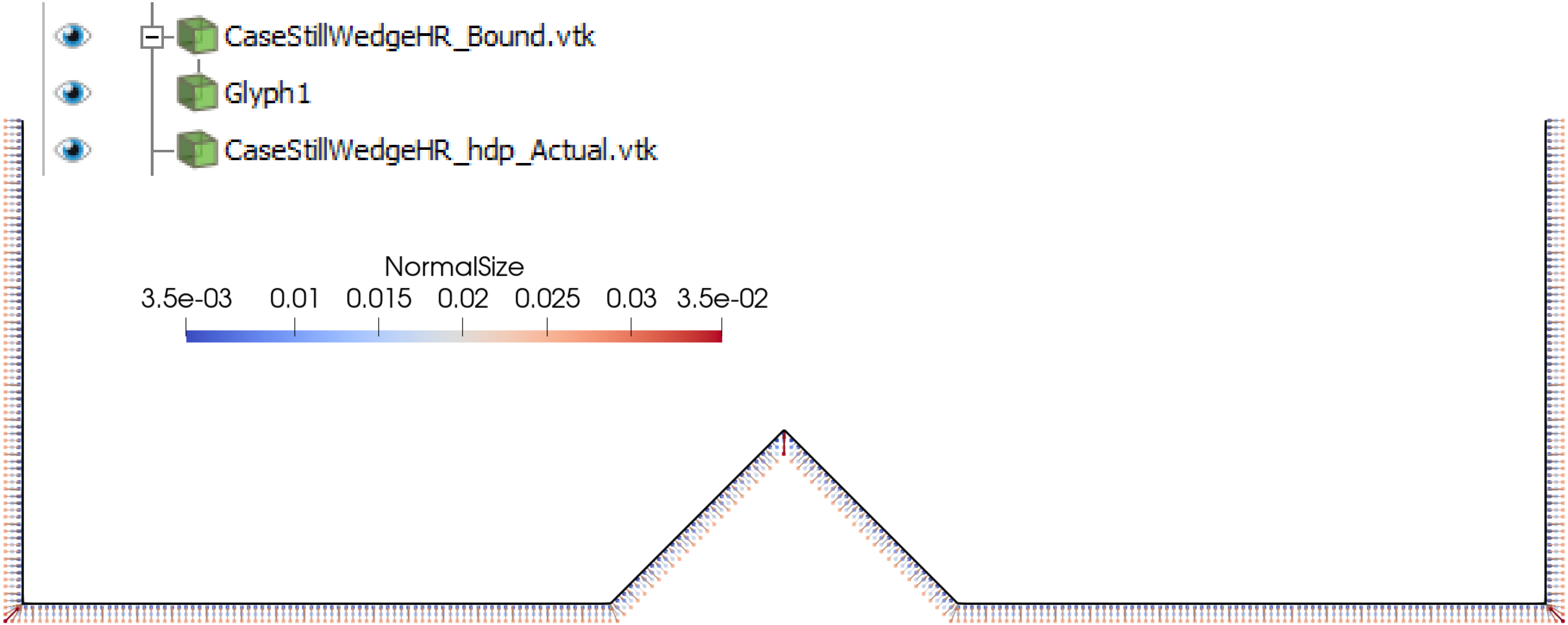
Defining and ch

```
<geometry>
    <definition dp="0.01" units_comment="metres (m)">
        <pointmin x="-1" y="0" z="-1" />
        <pointmax x="2.5" y="0" z="2" />
    </definition>
    <commands>
        <list name="GeometryForNormals">
            <setactive drawpoints="0" drawshapes="1" />
            <setshapemode>actual | bound</setshapemode>
            <setmkbound mk="0" />
            <drawextrude closed="false">
                <point x="0.005" y="-0.1" z="0.7" />
                <point x="0.005" y="-0.1" z="0.005" />
                <point x="0.85" y="-0.1" z="0.005" />
                <point x="1.1" y="-0.1" z="0.255" />
                <point x="1.35" y="-0.1" z="0.005" />
                <point x="2.195" y="-0.1" z="0.005" />
                <point x="2.195" y="-0.1" z="0.7" />
                <extrude x="0" y="0.2" z="0" />
            </drawextrude>
            <shapeout file="hdp" />
            <resetdraw />
        </list>
        <mainlist>
            <!-- Actual geometry at dp/2 -->
            <runlist name="GeometryForNormals" />
            <!-- Boundary particles -->
            <setdrawmode mode="full" />
            <setmkbound mk="0" />
            <drawextrude closed="false">
                <point x="0" y="0" z="0.7" />
                <point x="0" y="0" z="0" />
                <point x="0.85" y="0" z="0" />
                <point x="1.1" y="0" z="0.25" />
                <point x="1.35" y="0" z="0" />
                <point x="2.2" y="0" z="0" />
                <point x="2.2" y="0" z="0.7" />
                <extrude x="0" y="0.1" z="0" />
                <layers vdp="0,-0.5,-1,-1.5,-2" />
            </drawextrude>
            <!-- Fluid particles -->
            <setmkfluid mk="0" />
            <fillbox x="0.1" y="0" z="0.1">
                <modefill>void</modefill>
                <point x="-0.1" y="-1" z="-0.1" />
                <size x="2.3" y="2" z="0.6" />
            </fillbox>
        </mainlist>
    </commands>
</geometry>
```

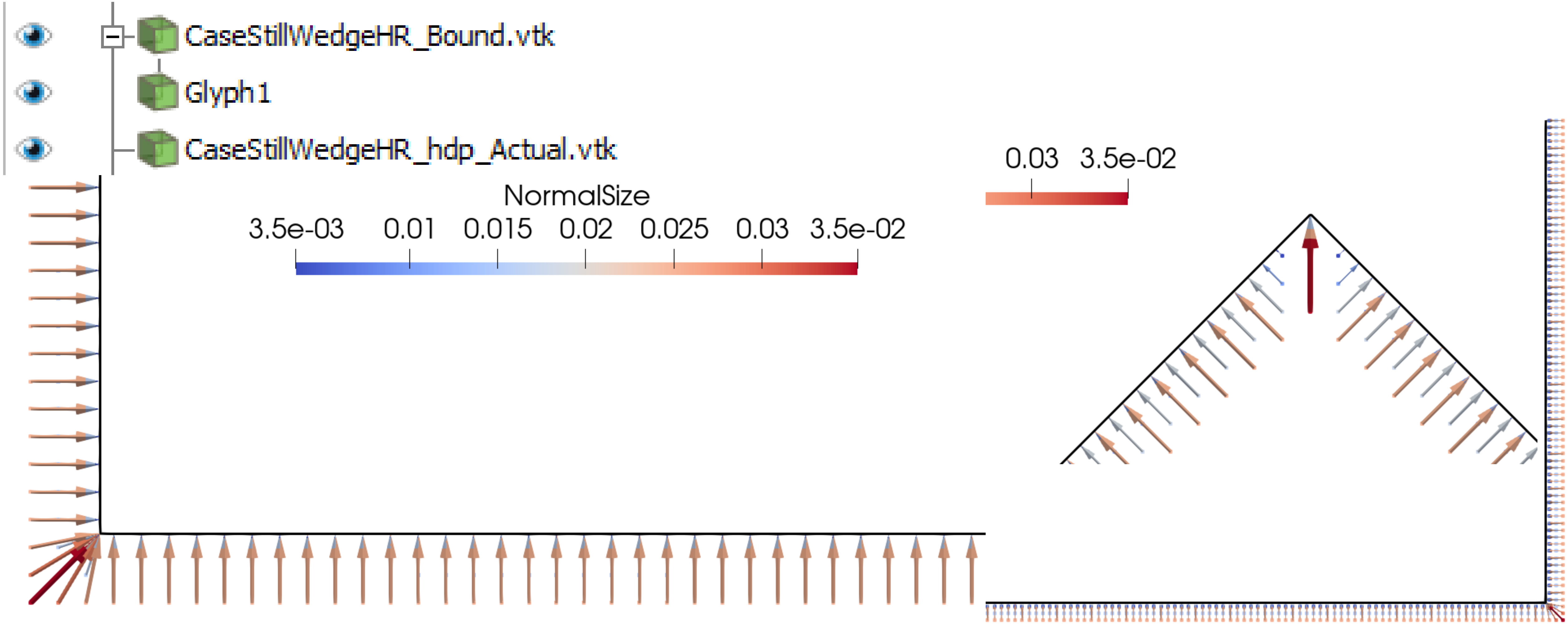

```
<list name="GeometryForNormals">
    <setactive drawpoints="0" drawshapes="1" />
    <setshapemode>actual | bound</setshapemode>
    <setmkbound mk="0" />
    <drawextrude closed="false">
        <point x="0.005" y="-0.1" z="0.7" />
        <point x="0.005" y="-0.1" z="0.005" />
        <point x="0.85" y="-0.1" z="0.005" />
        <point x="1.1" y="-0.1" z="0.255" />
        <point x="1.35" y="-0.1" z="0.005" />
        <point x="2.195" y="-0.1" z="0.005" />
        <point x="2.195" y="-0.1" z="0.7" />
        <extrude x="0" y="0.2" z="0" />
    </drawextrude>
    <shapeout file="hdp" />
    <resetdraw />
</list>
```

```
<mainlist>
    <!-- Actual geometry at dp/2 -->
    <runlist name="GeometryForNormals" />
    <!-- Boundary particles -->
    <setdrawmode mode="full" />
    <setmkbound mk="0" />
    <drawextrude closed="false">
        <point x="0" y="0" z="0.7" />
        <point x="0" y="0" z="0" />
        <point x="0.85" y="0" z="0" />
        <point x="1.1" y="0" z="0.25" />
        <point x="1.35" y="0" z="0" />
        <point x="2.2" y="0" z="0" />
        <point x="2.2" y="0" z="0.7" />
        <extrude x="0" y="0.1" z="0" />
        <layers vdp="0,-0.5,-1,-1.5,-2" />
    </drawextrude>
    <!-- Fluid particles -->
```

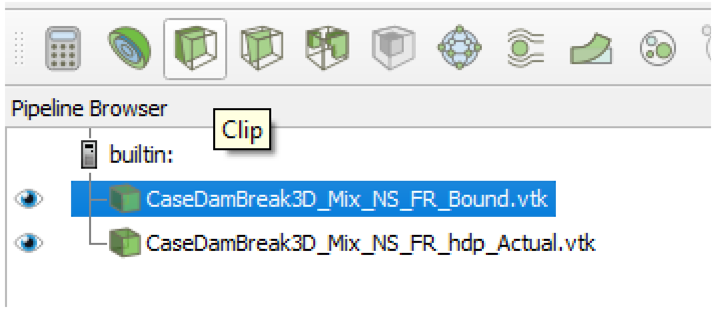
Defining and checking normals

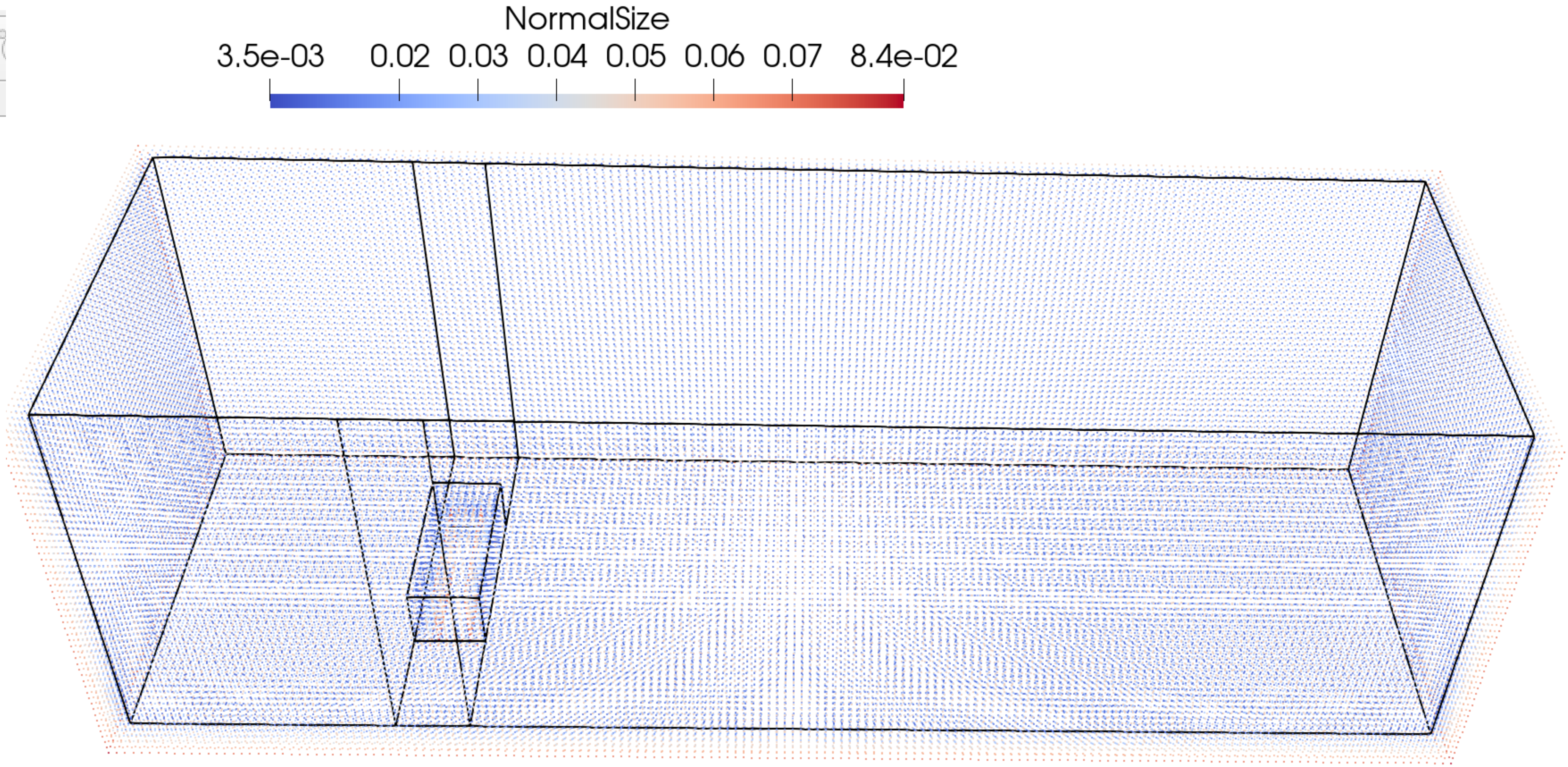


Defining and checking normals

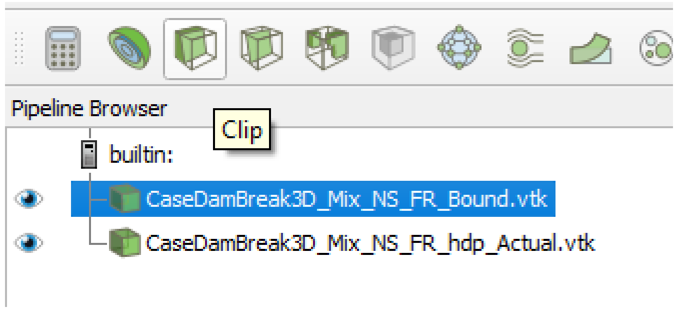


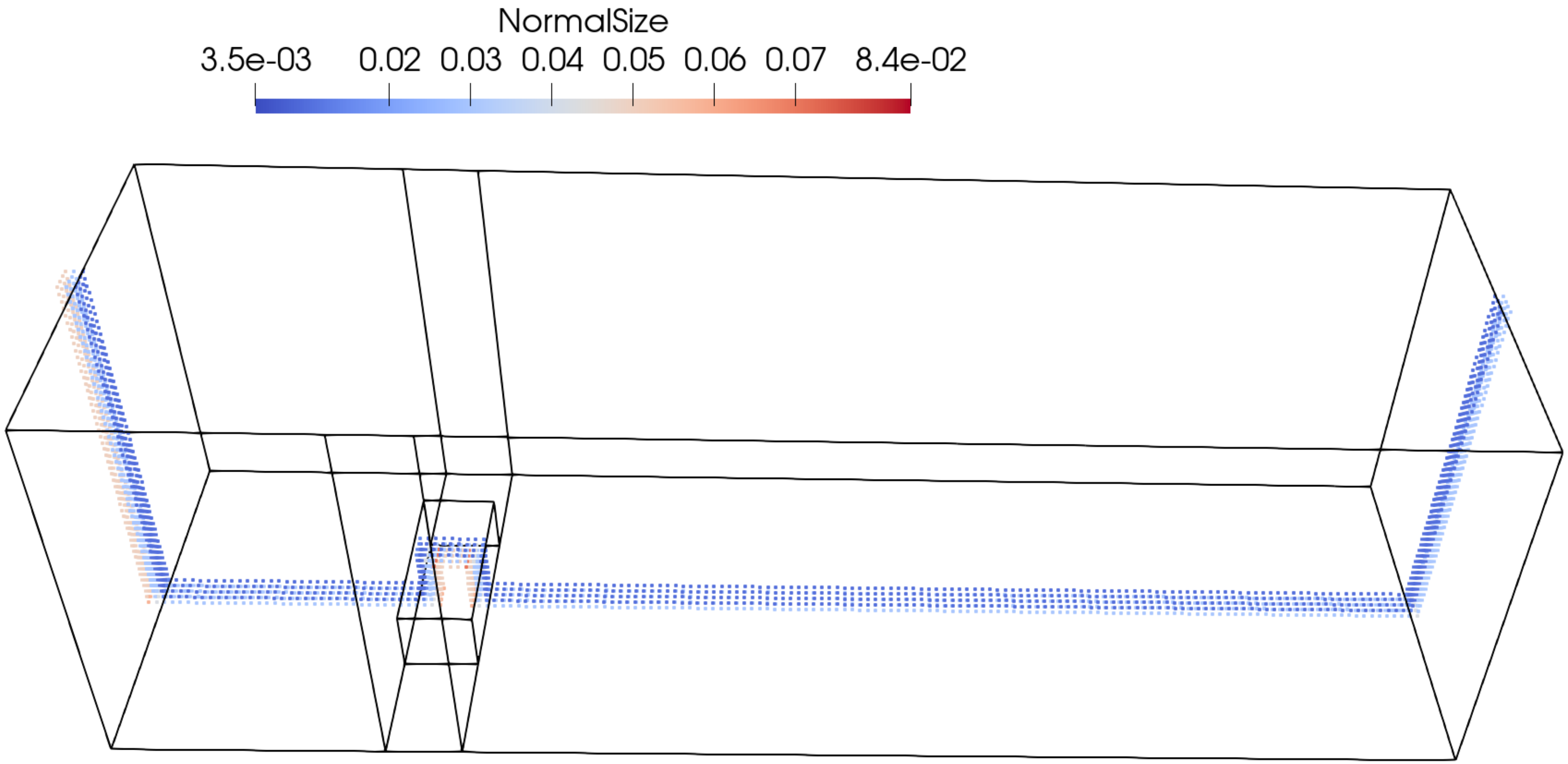
Defining and checking normals 3D



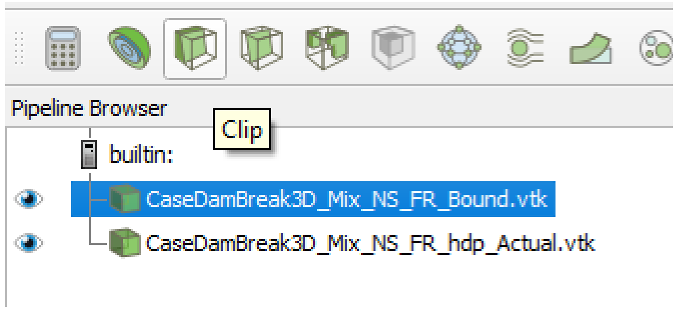


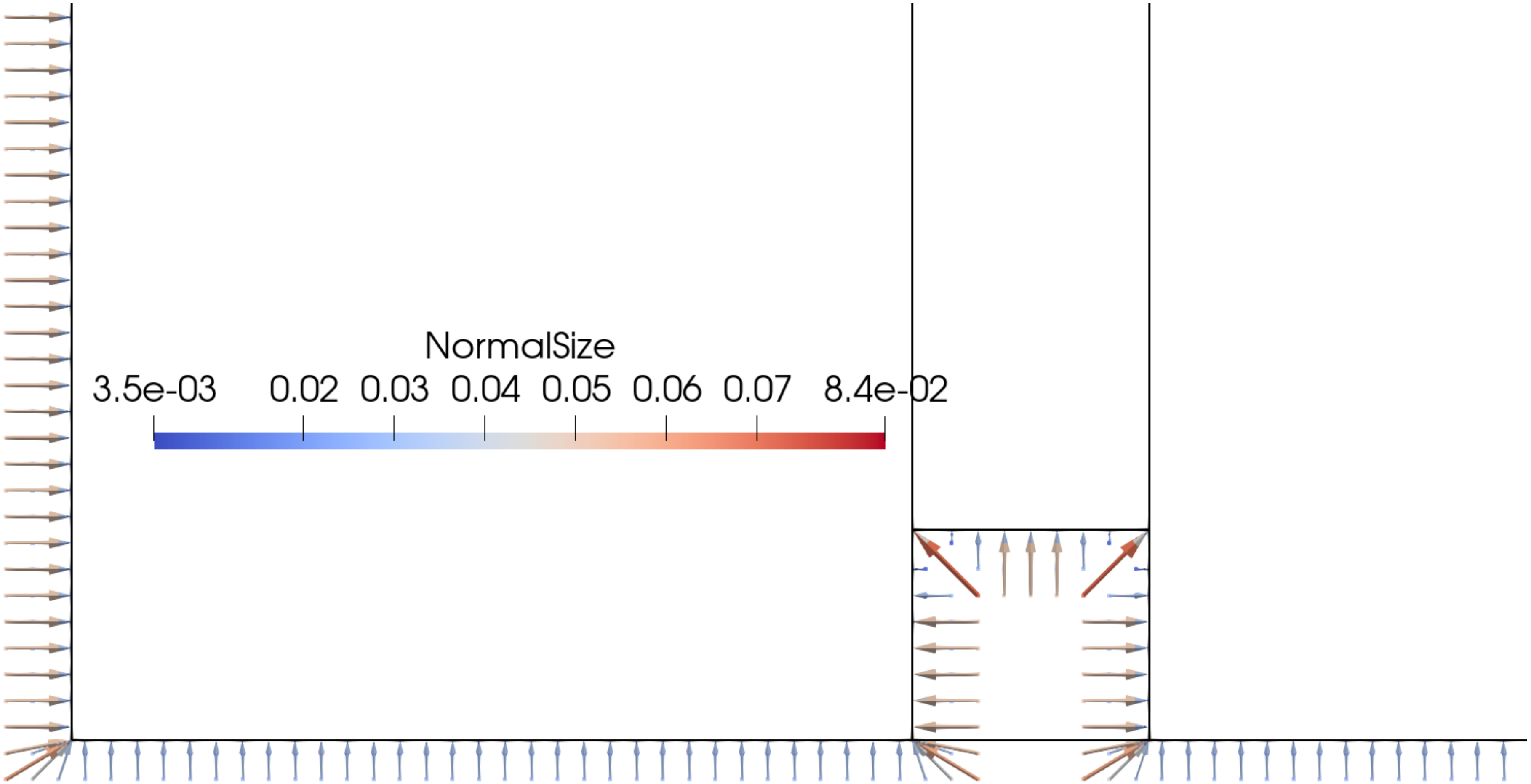
Defining and checking normals 3D





Defining and checking normals 3D





Conclusions

- There are now 5 boundary options to choose from in DualSPHysics
- It is not possible to use more than one boundary condition in simulation
- Previous simulation outputs can be used as initial conditions for easier set ups and reduced overall run times
- Checking boundary normals and boundary set up has never been easier