Introduction to open source CFD

Session 3/4

Gijsbert Wierink
gijsbert.wierink@jku.at

Christian Doppler Laboratory on Particulate Flow Modelling
Johannes Kepler University | Linz | Austria

Disclaimer

This course is offered non-commercially as part of course work at the Department of Particulate Flow Modelling (PFM) at the Johannes Kepler University (JKU) in Linz, Austria. This offereing is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OPENFOAM® and OpenCFD ® trade marks.

Introduction to open source CFD
Outline

- Think of a project
- Meshing with snappyHexMesh
- Vertical hot plate

PROJECT WORK

Project work

- Passing based on presence and project work
- Pick something you like and/or need
- Projects are presented in a small seminar
- We need to agree on a
  - topic
  - date

SNAPPYHEXMESH

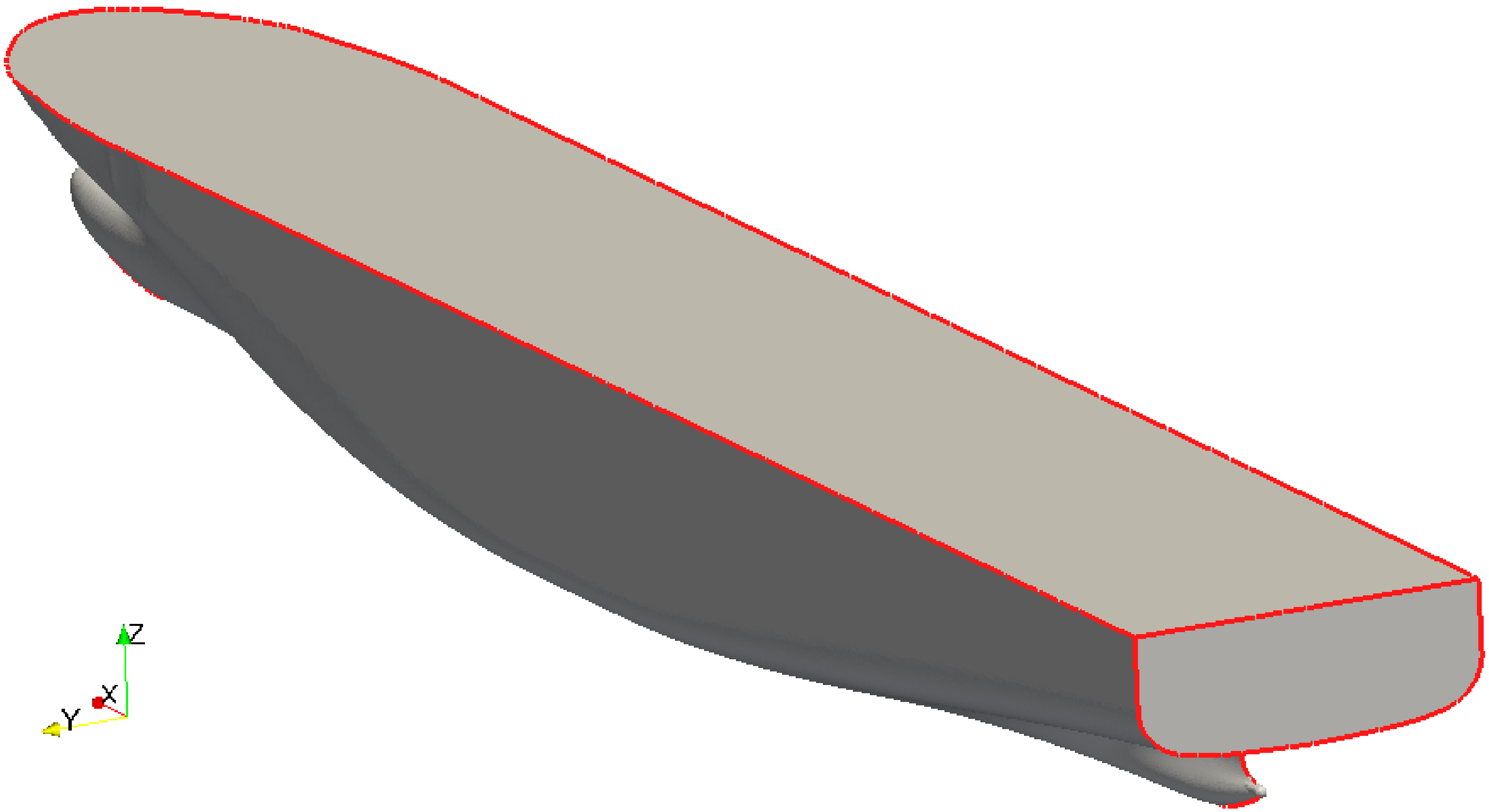
Advanced meshing
snappyHexMesh

- Meshing in OpenFOAM can be doen by
  - blockMesh
  - snappyHexMesh
  - Conversion

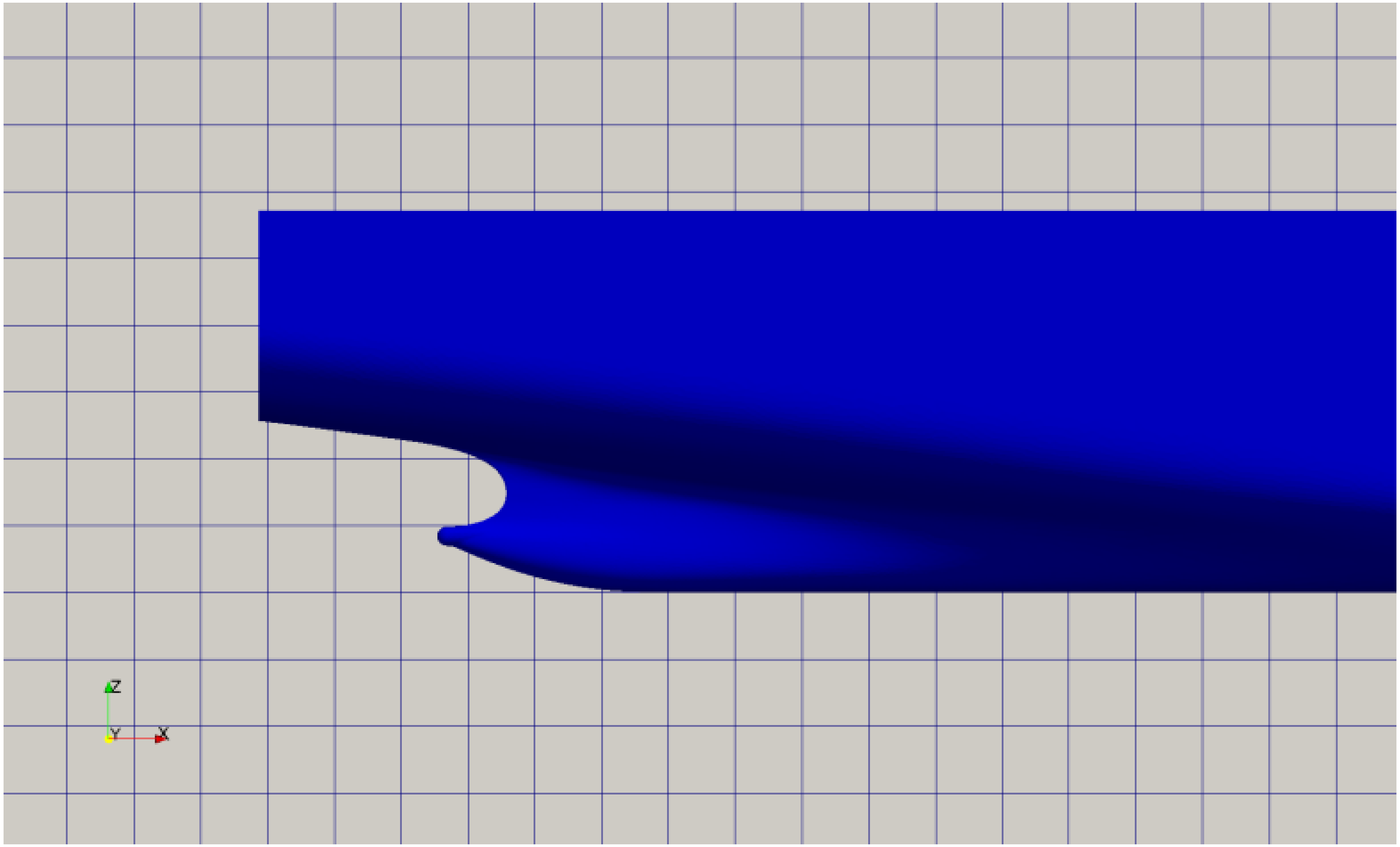
Advanced meshing
snappyHexMesh: process

- Get/create a triangulated surface (e.g. stl/obj)
- (Extract feature edges) (`surfaceFeatureExtract`)
- Gerenate a background mesh (`blockMesh`)
- Run snappyHexMesh in three stages
  - Generate castellated mesh
  - Snapping to surfaces and edges
  - Layer addition and clean-up

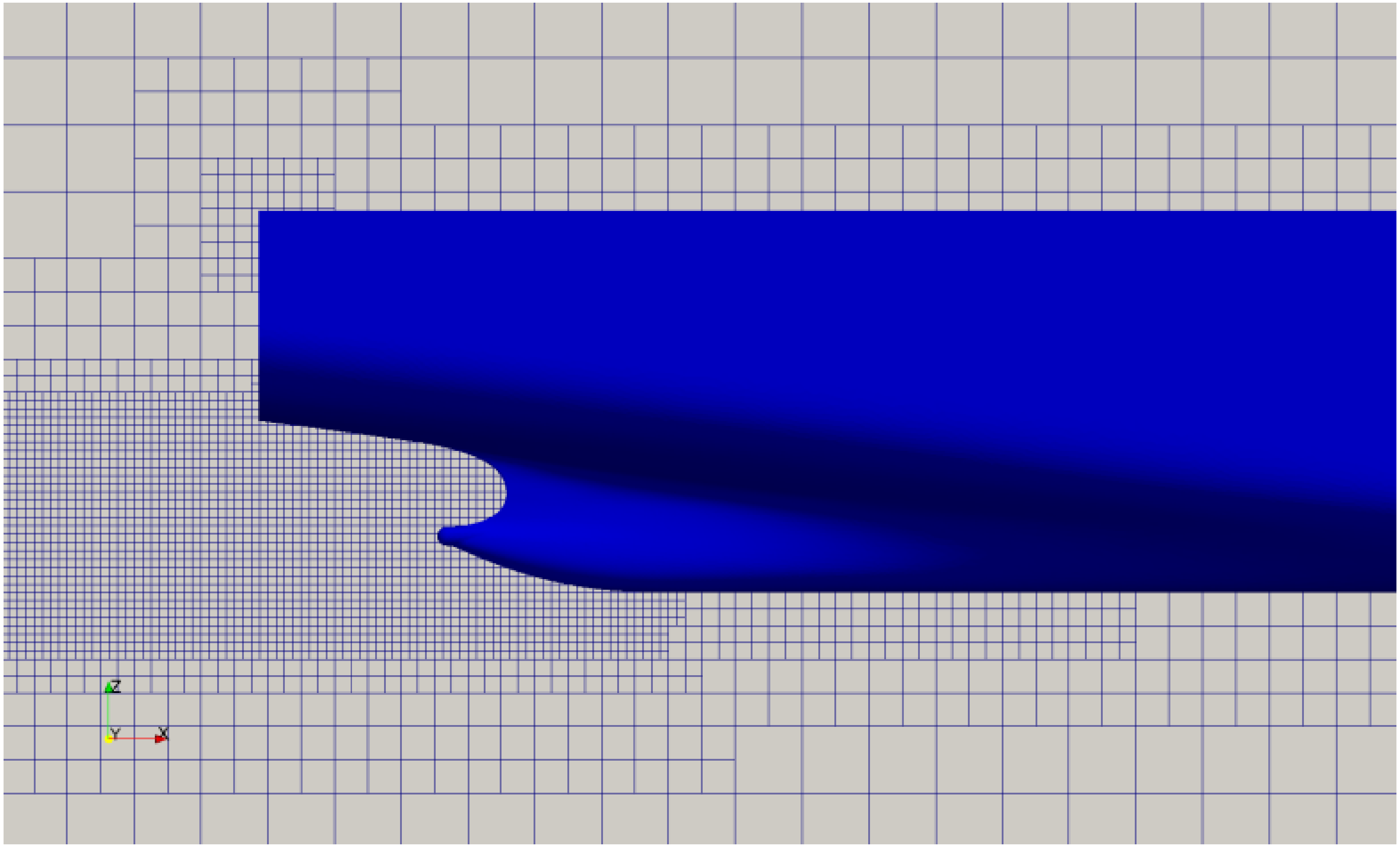
Advanced meshing
snappyHexMesh: feature edges



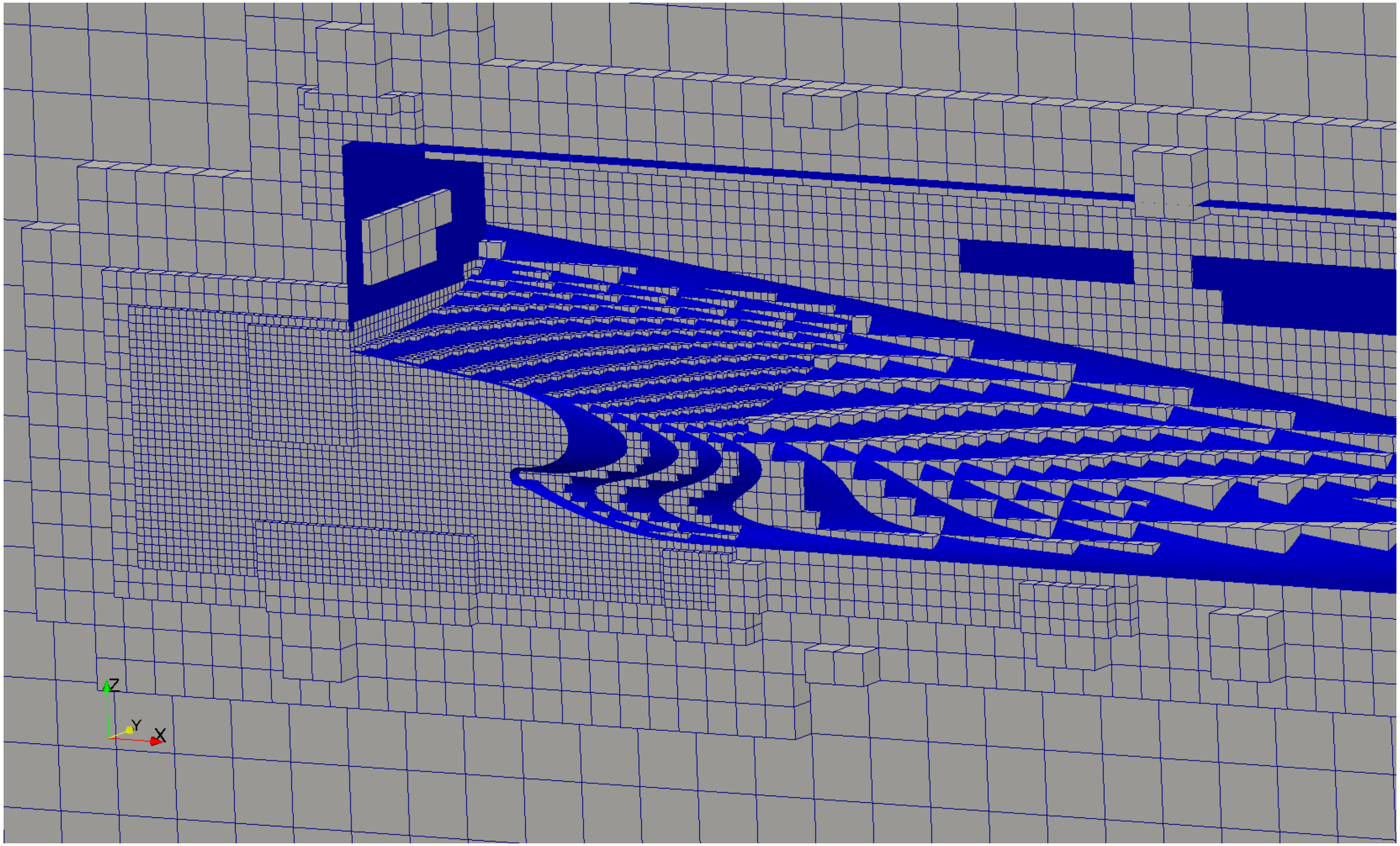
Advanced meshing
snappyHexMesh: background mesh



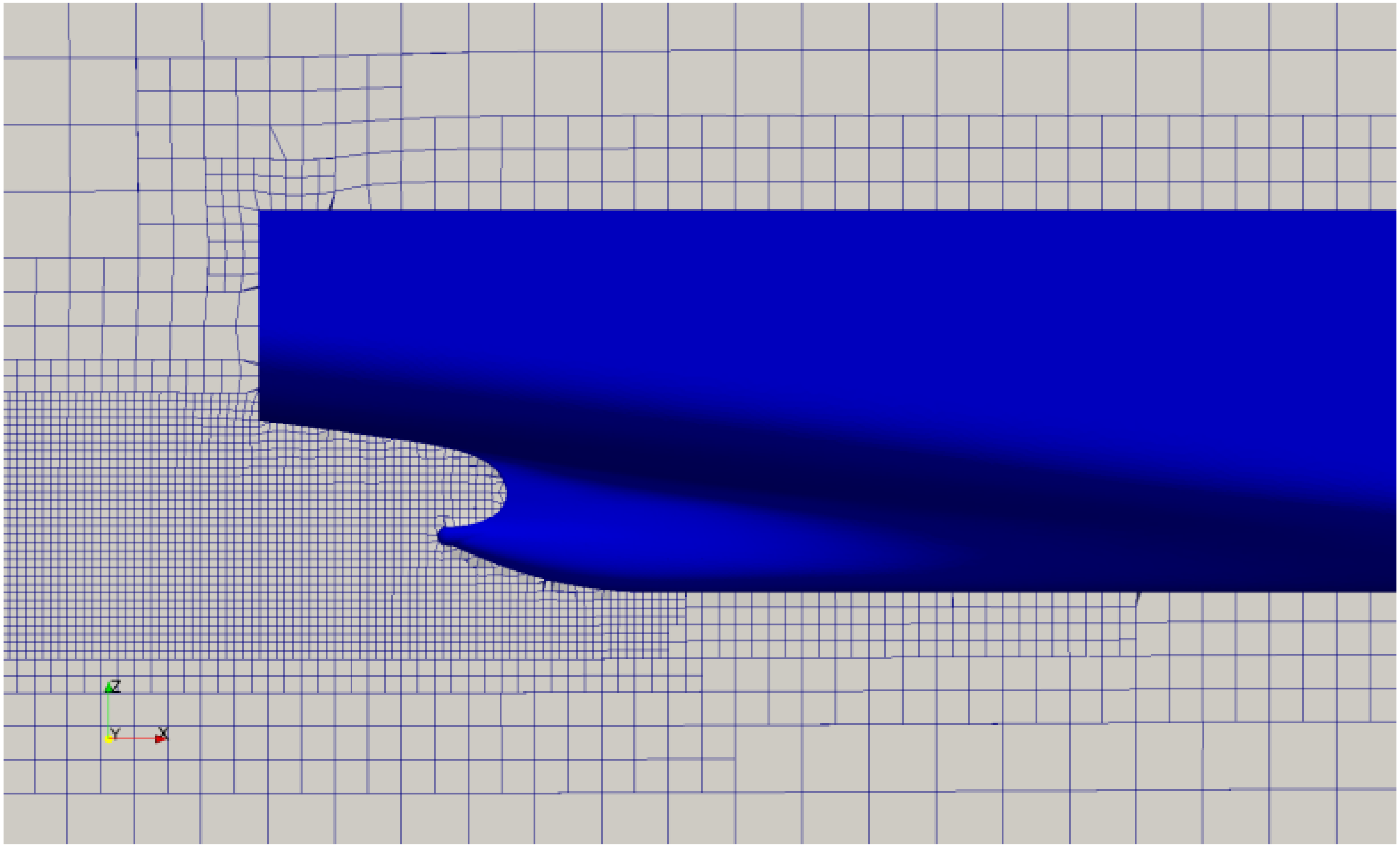
Advanced meshing
snappyHexMesh: castellated mesh



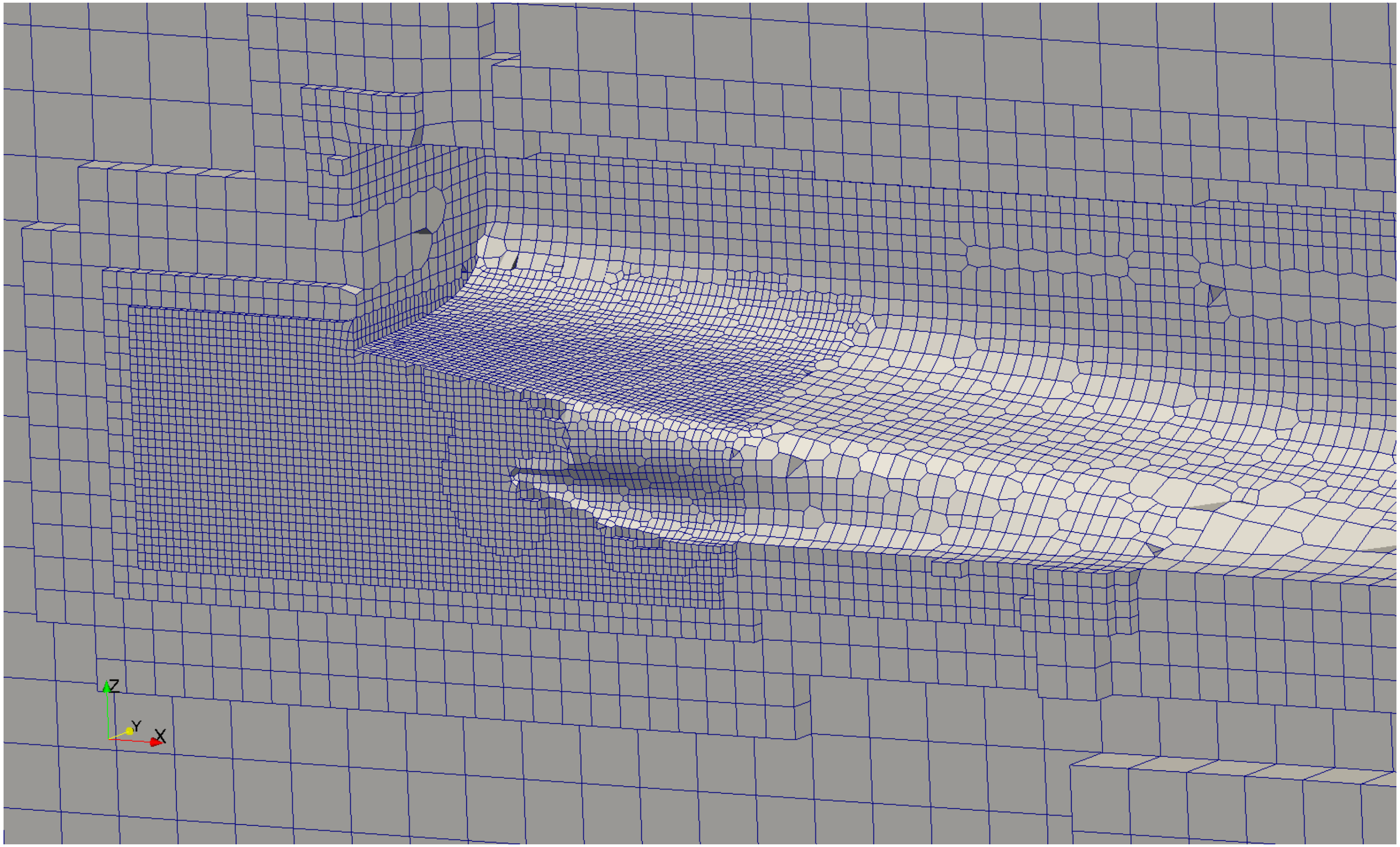
Advanced meshing
snappyHexMesh: castellated mesh



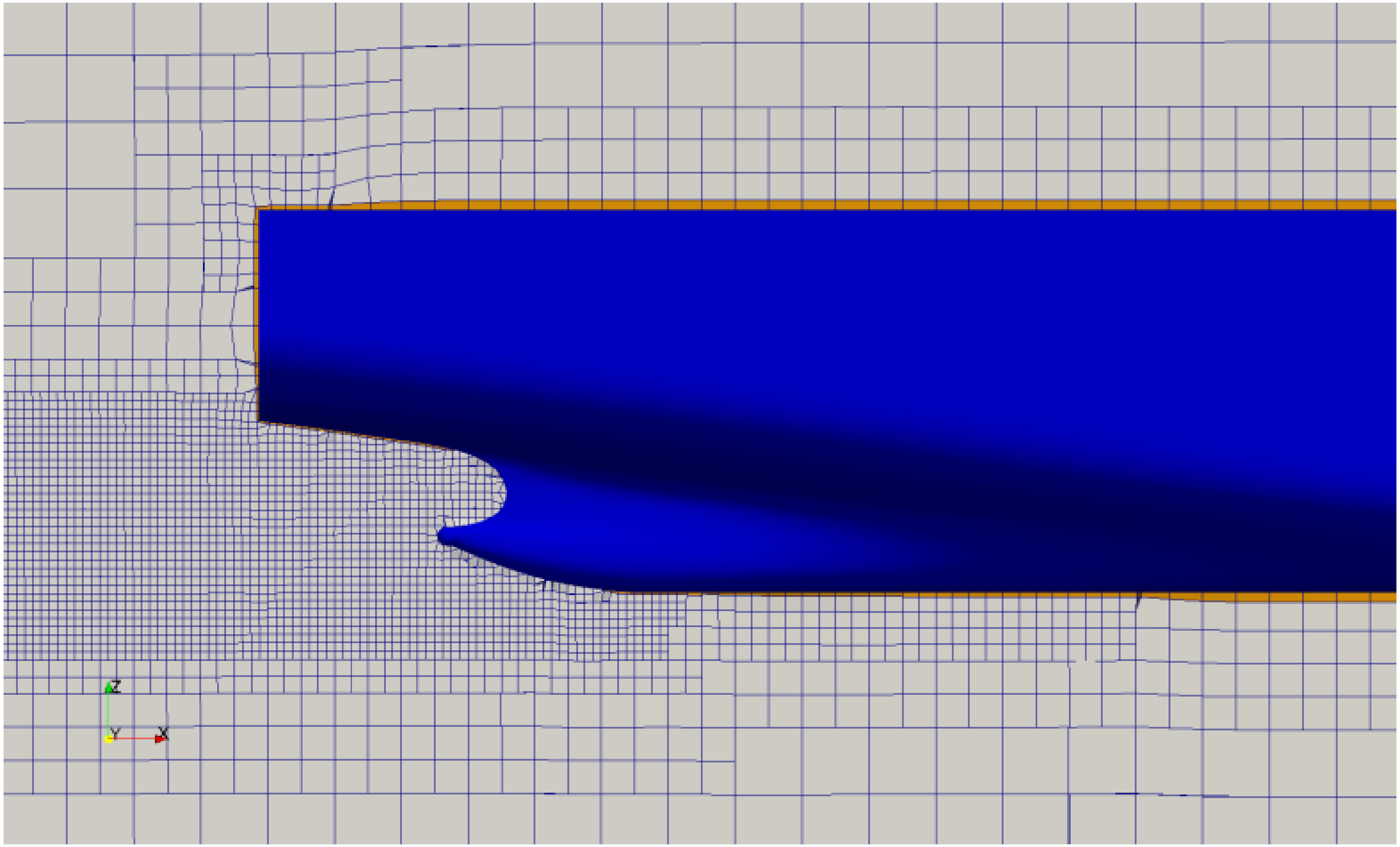
Advanced meshing
snappyHexMesh: snapping



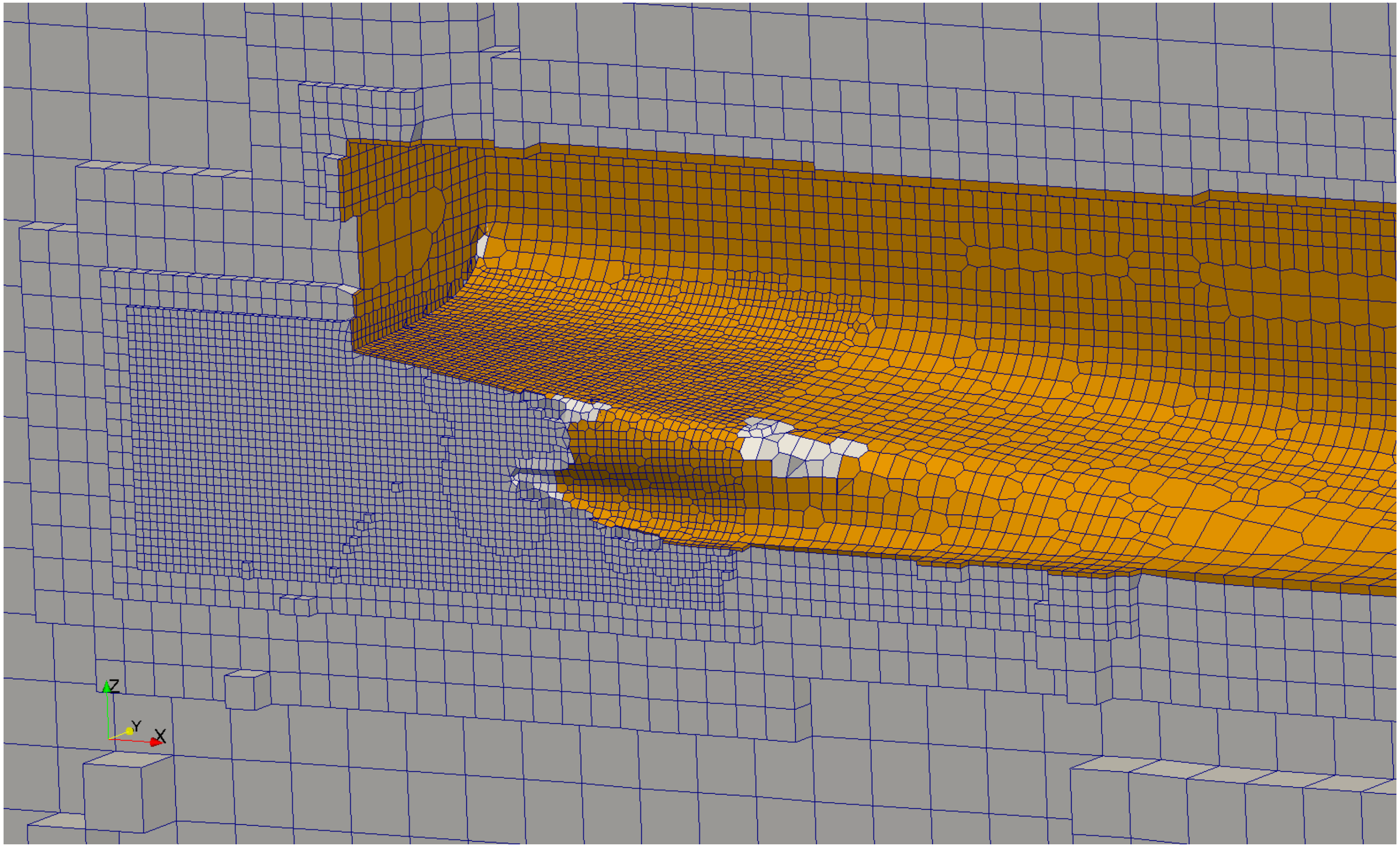
Advanced meshing
snappyHexMesh: snapping



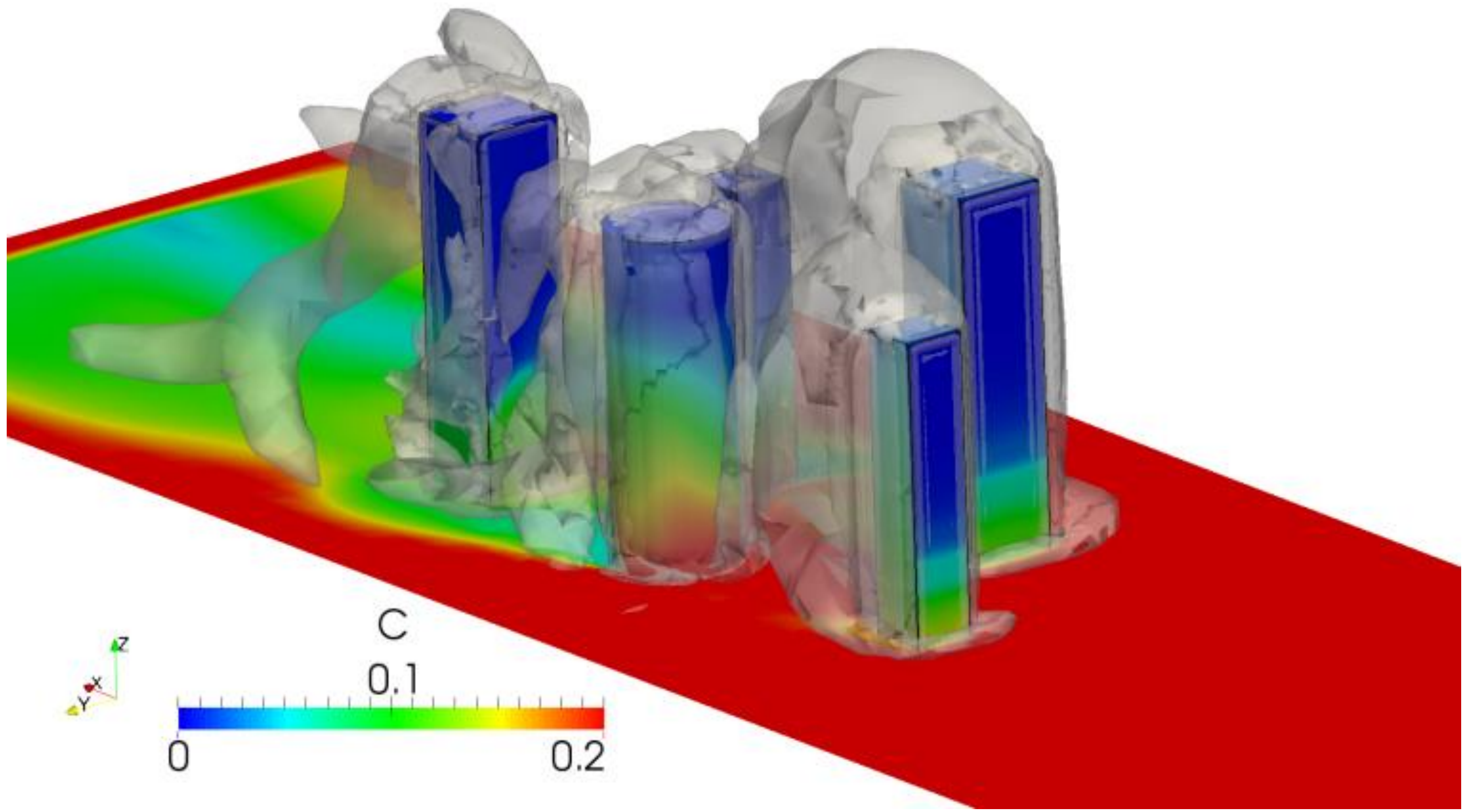
Advanced meshing
snappyHexMesh: layer addition



Advanced meshing
snappyHexMesh: layer addition



Dust in desert city
Modelling desert dust as massless species



Dust in desert city

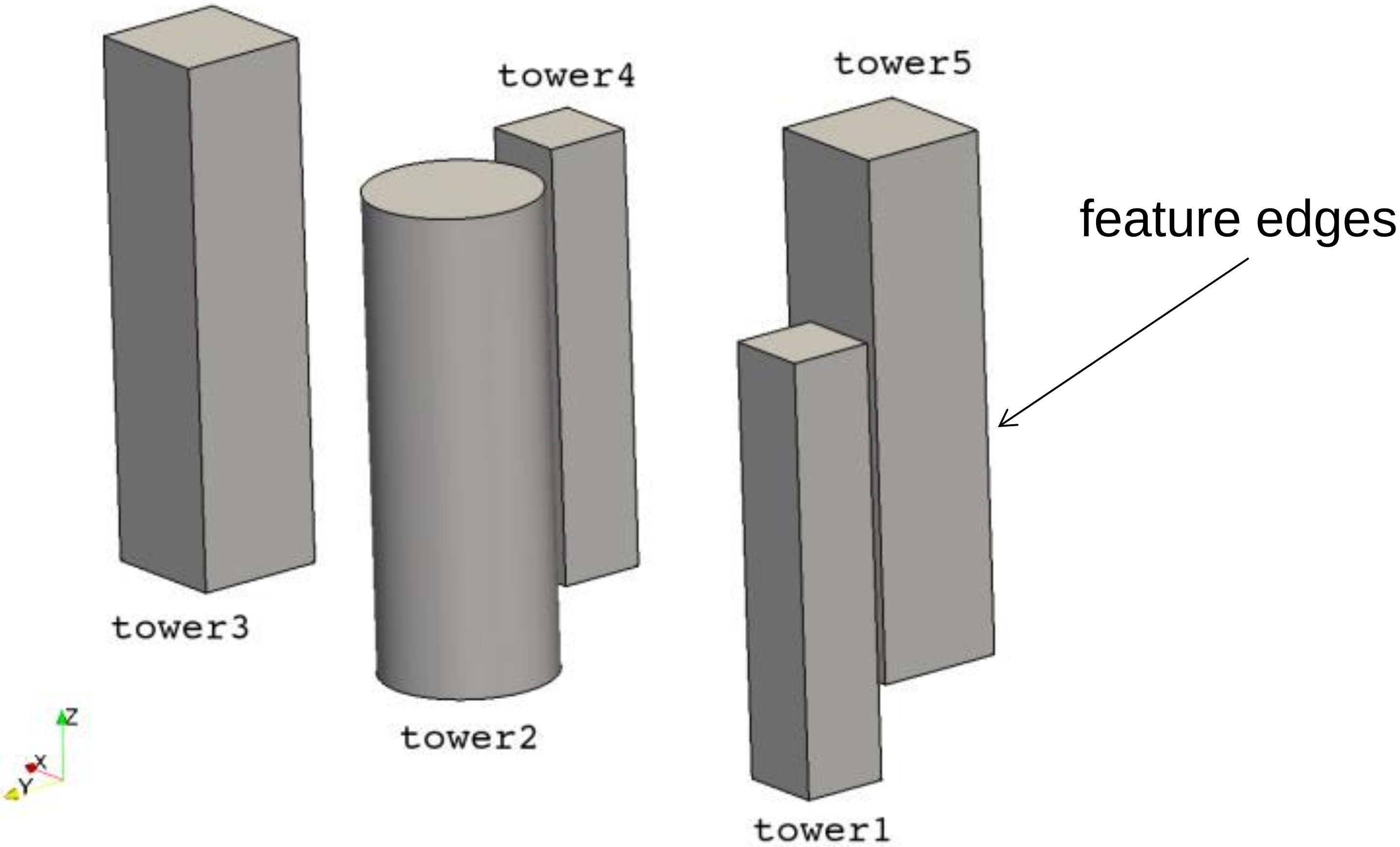
Background mesh and feature edges

- Run `blockMesh` to generate the background mesh (see `constant/polyMesh/blockMeshDict`)
- Run `surfaceFeatureExtract` to extract surface features from the triangulated surface. These edges can be used by `snappyHexMesh` for edge snapping. (see `system/surfaceFeatureExtractDict`)

```
> blockMesh
> surfaceFeatureExtract
```

Dust in desert city

Background mesh and feature edges

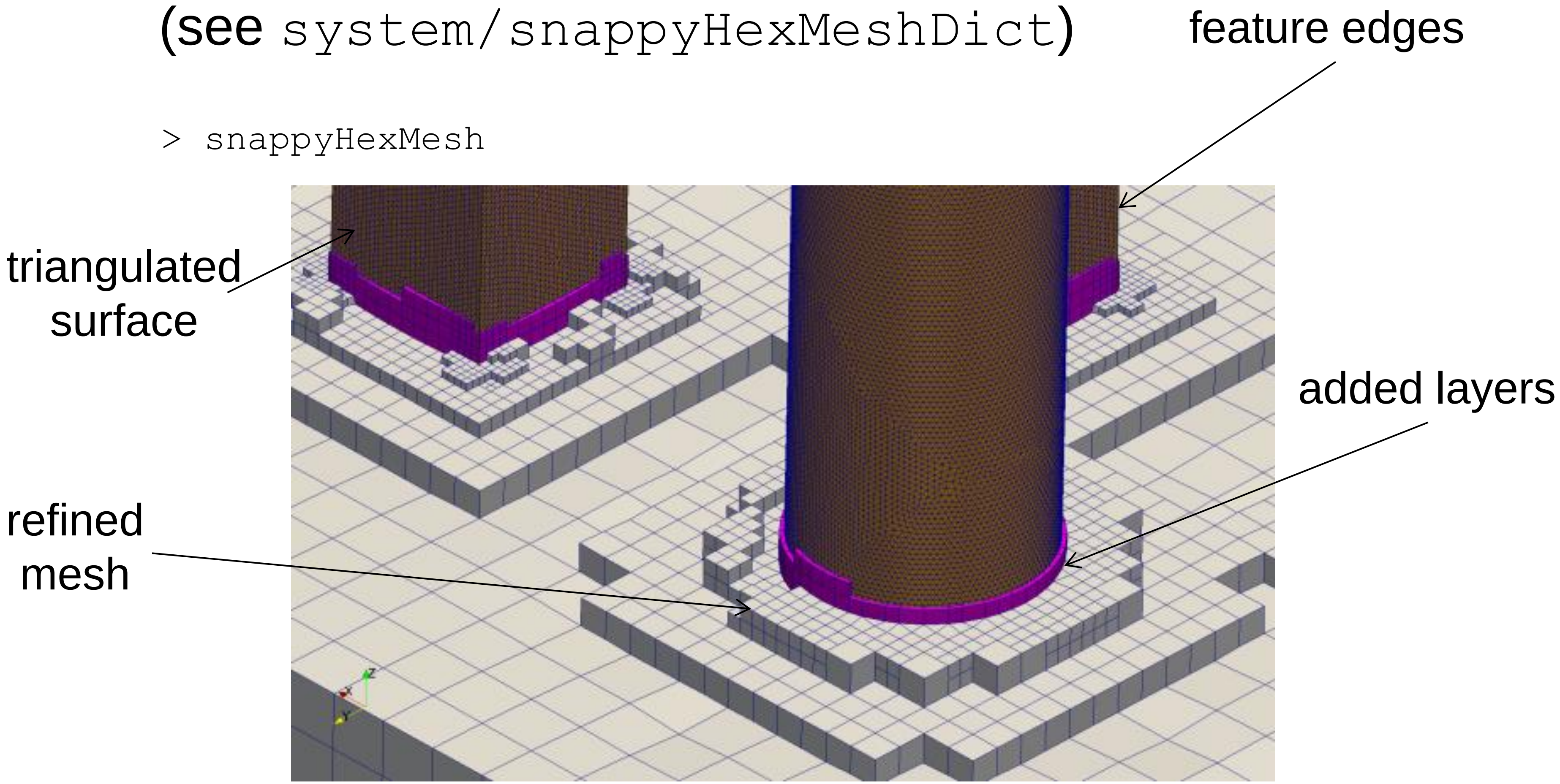


Dust in desert city
Background mesh and feature edges

- Now run `snappyHexMesh`

  (see `system/snappyHexMeshDict`)

```
> snappyHexMesh
```



Dust in desert city
Rough wall function

- To model the desert floor we use `nutkRoughWallFunction` on the ground patch (see file `0/nut`)

$$\frac{U_P}{u^*} = \frac{1}{\kappa} \ln\left(\frac{Ey}{C_s K_s}\right) \qquad u^* = C_\mu^{0.25}\sqrt{k}$$

```
ground
{
    type      nutkRoughWallFunction;
    Ks        uniform 0.003; // sand-grain height (m)
    Cs        uniform 0.5;   // rougness constant (0.5-1)
    value     uniform 0;     // dummy value
}
```

Dust in desert city

Dust in Desert City

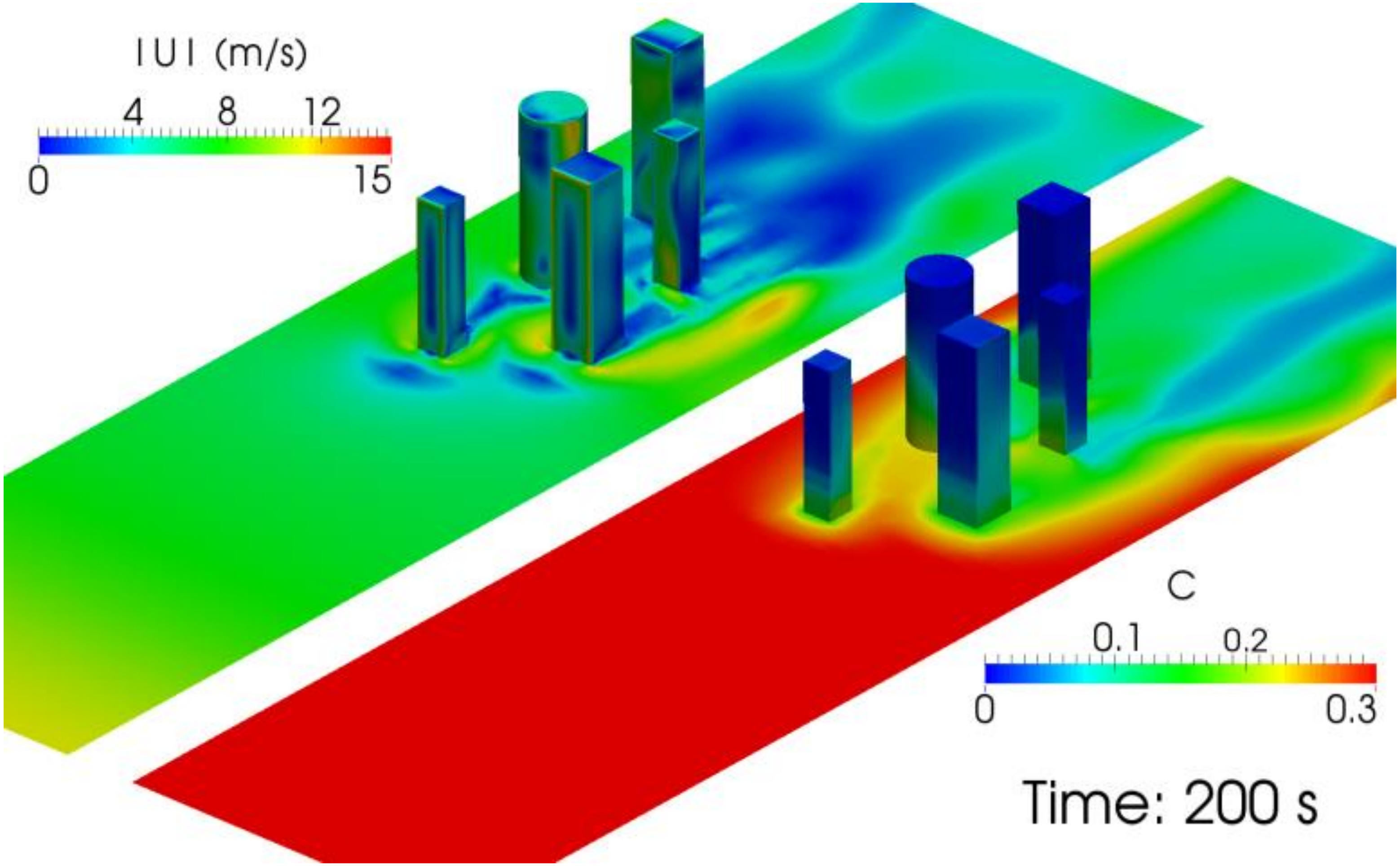
- Copy the initial fields

```
> cp -r 0.org 0
```

- Run the solver

```
> pimpleSpeciesTransportFoam
```

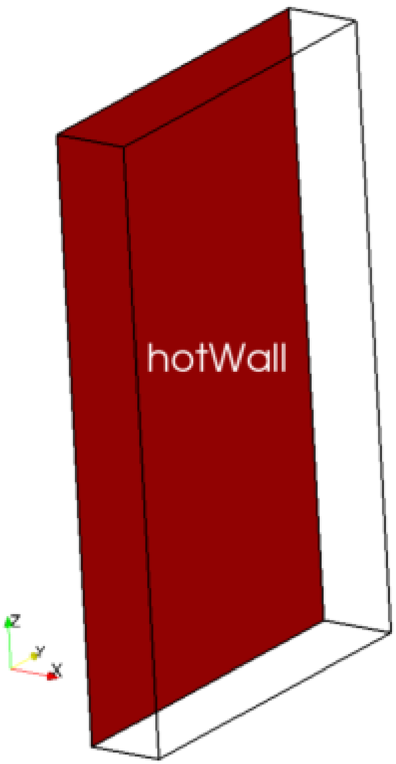
Dust in desert city
Dust in Desert City



HEAT TRANSFER

Heat transfer
Hot plate

- Worked example in VDI Wärmeatlas
- Vertical wall with open domain
  - Plate is 0.5 m wide, 0.8 m high
- Wall temperature fixed
  - High density boundary layer
  - Parametrization
  - Robust and simple meshing
- Copy the case and create the mesh




```
> run
> export HT="$OSCCAR_TUTORIALS/heatTransfer"
> cp -r $HT/buoyantBoussinesqPimpleFoam/veritcalHotPlate .
> cd verticalHotPlate
> blockMesh
```

Heat transfer

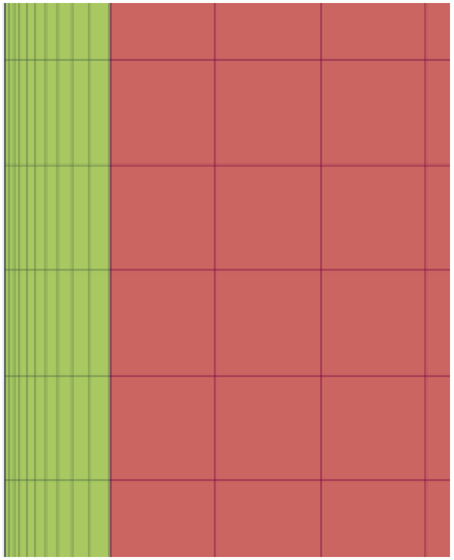
Hot plate: blockMeshDict

```
convertToMeters 1;

vertices
(
    (0     0.25 -0.4) //  0
    (0.01  0.25 -0.4) //  1
    (0.1   0.25 -0.4) //  2
    (0    -0.25 -0.4) //  3
    (0.01 -0.25 -0.4) //  4
    (0.1  -0.25 -0.4) //  5
    (0     0.25  0.4) //  6
    (0.01  0.25  0.4) //  7
    (0.1   0.25  0.4) //  8
    (0.1  -0.25  0.4) //  9
    (0.01 -0.25  0.4) // 10
    (0    -0.25  0.4) // 11
);

blocks
(
    hex (3 4 1 0 11 10 7 6) (10 50 80) simpleGrading (5 1 1) // Boundary layer
    hex (4 5 2 1 10 9 8 7)  (9 50 80)  simpleGrading (1 1 1) // Main domain
);

edges
(
);
```



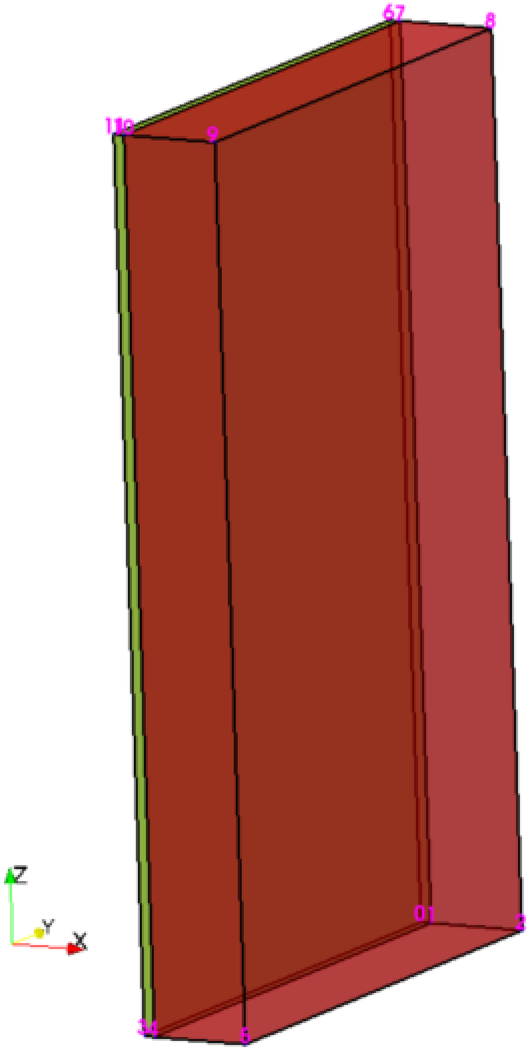
Heat transfer

Hot plate: blockMeshDict

```
boundary
(
    hotWall
    {
        type wall;
        faces
        (
            (0 3 11 6)
        );
    }

    sides
    {
        type patch;
        faces
        (
            (3 4 10 11) // Front BL
            (0 6 7 1)   // Back BL
            (4 5 9 10)  // Front main domain
            (1 7 8 2)   // Back main domain
            (2 8 9 5)   // Right side main domain
        );
    }
     ...
);

mergePatchPairs
(
);
```



Heat transfer
Run the solver and sample U and T

- Now run the solver
  `> buoyantBoussinesqPimpleFoam`

- View the results in ParaView
  `> paraFoam`

- Sample velocity and temperature
  `> gedit system/sampleDict`

```
interpolationScheme cellPoint;

        setFormat       raw;

        sets
        (
            sampleLine
            {
                type    face;
                axis    x;
                start   ( 0 0 0 );
                end     ( 0.1 0 0 );
            }
        );

        fields ( U T );
```

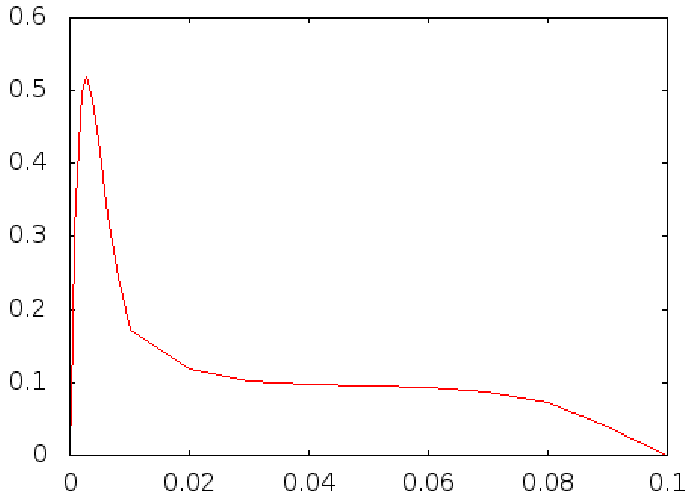
Heat transfer
Post-processing

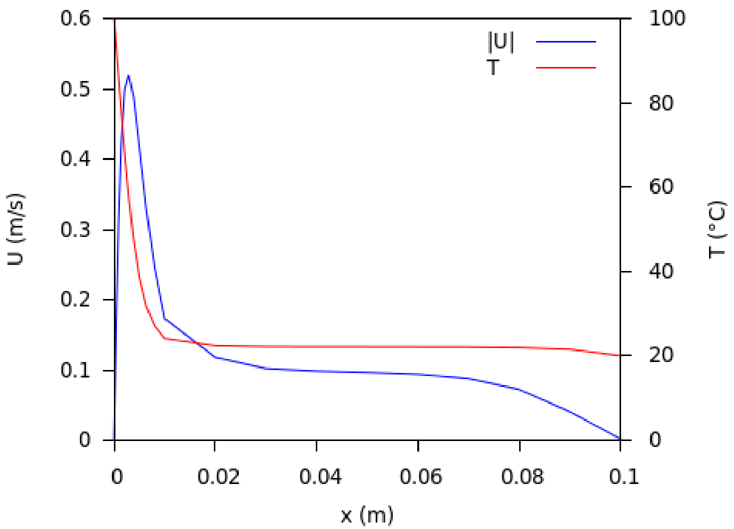
- `sample` writes data to `postProcessing/<setFormat>/<time>`

```
> ls postProcessing/sets/5
sampleLine_T.xy  sampleLine_U.xy
```

- Plot e.g. Uz using gnuplot

```
> gnuplot
gnuplot> plot "postProcessing/sets/5/sampleLine_U.xy" \
         using 1:4 with lines notitle
```





Heat transfer
Post-processing

- Use OSCCAR‘s incompressible wallHeatFlux utility.
  Make sure it is compiled:

```
> wmake $OSCCAR_UTILITIES/postProcessing/wall/wallHeatFluxIncompressible
```

- Add thermophysical poperties to the case

```
> export OUTIL=$OSCCAR_UTILITIES/postProcessing/wall/wallHeatFluxIncompressible
> gedit \
    $OUTIL/thermophysicalProperties constant/thermophysicalProperties &
```

- Make sure thermopyhsical properties are in right range:

```
CpRef  = 1005  [J/kgK]
RhoRef = 1.2   [kg/m3]
Pr     = 0.715 [-]
Prt    = 0.85  [-]
```

Heat transfer
Post-processing

- Use the incompressible wallHeatFlux utility and write to log file

```
> wallHeatFluxIncompressible |& tee log.wallHeatFlux
```

- Filter the log file, create data file with time and heat flux

```
> grep "Time =" log.wallHeatFlux | cut -d' ' -f3 > time.tmp
> grep hotWall log.wallHeatFlux | cut -d' ' -f2 > Q.tmp
> paste time.tmp Q.tmp > wallHeatFlux.dat
> rm *.tmp
```

- Now plot from the data file

```
> gnuplot
gnuplot> plot "wallHeatFlux.dat" \
         with lines title "CFD",\
         32 with lines title "Correlation"
```