Tutorial: BwUniCluster 2.0/HoreKa
CHT case: ChipHX (serial)

In this tutorial you will learn about the conjugate heat transfer (CHT) simulation of a plate (represents a chip) and some zylinders (= pins) in OpenFOAM serially by an Allrun-script. The Allrun-script could be modified to a submit-script for the bwUnicluster/HoreKa easily.

1. **Tutorial Case**

First, download the the zipped .tar-file chipHX_4Students_CHT.tar.gz from the course-webpage and unzip the file into your workspace :

$>   cd <workdir>
$>   tar –xvf chipHX_4Students_CHT.tar.gz
$>   module load cae/openfoam/v2406
$>   source $FOAM_INIT
$>   cd chipHX_4Students_CHT/

The case includes the geometry of a plate and some cylinders (pins). An air flow is passing the solid geometry. The plate is heated at the bottom. Via the pins and the other walls of the plate the heat is released to the passing air flow. The plate originally consists of aluminum.

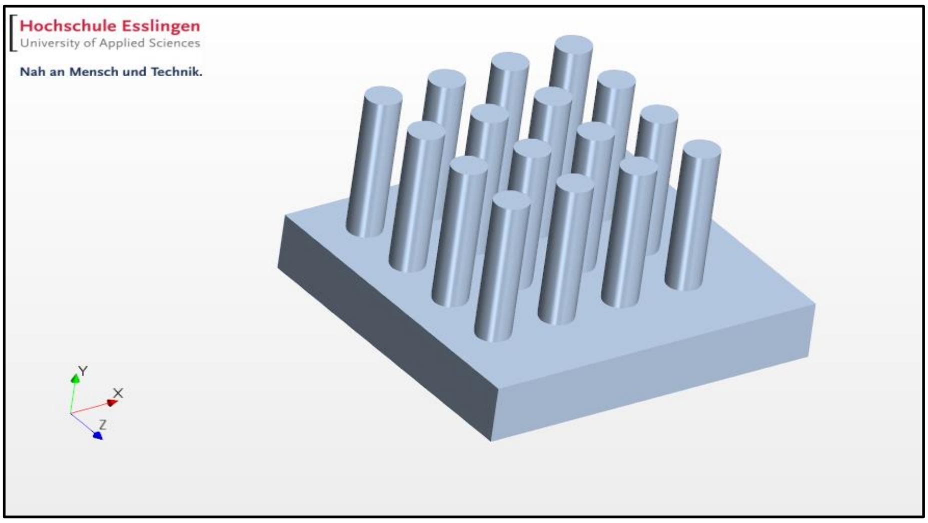

Fig.1 The chipHX geometry

2. **Learning content**

In this example you will get in touch and learn the following issues:

- Mesh a more complex geometry (more complex than rectangular.)
- Starting the CHT-solver of OpenFOAM

- Case directory structure of an CHT case
- Setting boundary conditions for CHT cases
- Analyzing residuals
- Setting up function objects and getting results from them
- Analyzing temperatures, heat fluxes and flows
- Setting solid material properties in OpenFOAM

3. **Running the case**

The mesh could be run serially by just calling the Allrun script.

$> cd chipHX_4Students_CHT/
$> ./Allrun

To repeat the run the case could be simply cleaned and rerun.

$> ./Allclean
$> ./Allrun

To run the case on bwUnicluster take the Allrun script without the first 2 lines (see also Fig.2) and add the typical header for an bwUnicluster job (from the Hot Room tutorial). Save the new file as my_dev_single_CHT_job.sh. Subsequently you can queue the job by:

$> sbatch my_dev_single_CHT_job.sh

```
# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions

# End of header
#------------------------------------------------------------------------------
# Select geometry for analysis
# Copy 0-template directory
cp -r 0.orig 0

#------------------------------------------------------------------------------
# Meshing
# Run blockMesh
runApplication blockMesh

# Extract Surface Feature Edges
runApplication surfaceFeatureExtract

# Run snappyHexMesh
runApplication snappyHexMesh -overwrite

# Run checkMesh
runApplication checkMesh
```

```
#------------------------------------------------------------------------------
# CHT Preparation
# Run splitMeshRegions
runApplication splitMeshRegions -cellZonesOnly -overwrite

# Run renumberMesh
runApplication renumberMesh -overwrite

#------------------------------------------------------------------------------
# Decomposing

#------------------------------------------------------------------------------
# Run solver
runApplication `getApplication`
#------------------------------------------------------------------------------
# Post-processing
# Reconstructing

# Foamlog
foamLog log.`getApplication`

# Create Paraview file
touch case.foam
#------------------------------------------------------------------------------
```

Fig.2 Allrun script without header

By the command runApplication `getApplication` the respective solver is taken from ./system/controlDict from the entry
application   chtMultiRegionSimpleFoam;

4. Structure of case directory

The structure of the case directory for a CHT case of OpenFOAM is different to a ordinary OpenFOAM case.

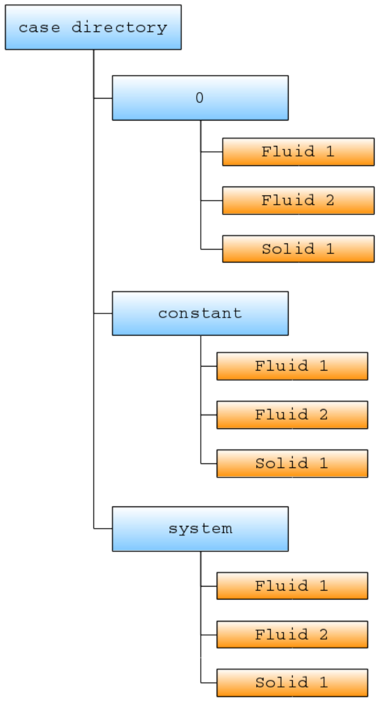

Fig.3 Case directory structure for OpenFOAM CHT cases

In this example there are two regions: one fluid region named *airFlow* and one solid region named *chipHX*.

5. Setting boundary conditions for CHT cases

Beside the normal setting of boundary conditions, like the inflow velocity in ./0.orig/airFlow/U for heat transfer problems a temperature of the inlet air has to be set. In this case the inlet air temperature is set to T_{in} = 293.15 K.
Furthermore, boundary conditions have to be set at the external boundaries of the solid region. In this case the (main) external boundary of the solid region is the *lowerWall*. There, a temperature or a heat flux or heat flow have to be set. In this case, a heat flux is set to q = 35200 W/m^2 via the boundary type *externalWallHeatFluxTemperature* (see also Fig. 4).
Internal boundaries between the fluid and the solid region do not need any specification of temperature or heat fluxes.

```
lowerWall
{
        type            externalWallHeatFluxTemperature;
        mode            flux;
        q               uniform 35200.0;
        kappaMethod     solidThermo;
        relaxation      0.99;
        value           $internalField;
}
```

Fig.4 Boundary conditions of lowerWall in ./0.orig/chipHX/T

6. Analyzing residuals

If you have run the simulation on the cluster you we can download the results onto our personal computer after the simulation run has been finished and visualize them in Paraview.

First, if you have installed gnuplot you can have a look on the residuals by:

$> ./gnuplot Residuals.plt

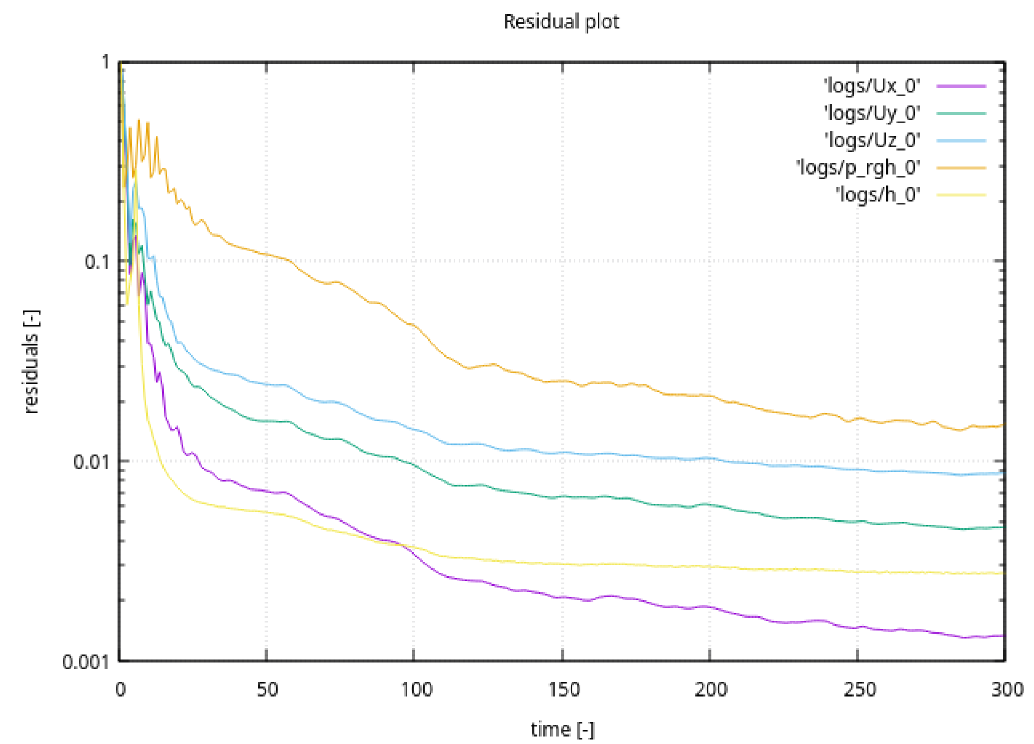

Fig.5 Residual plot of the chipHX case

7. Analyzing results from function objects

You can use function objects to analyze several physical data from the results of your OpenFOAM results.
Function objects are implemented within your *controlDict* in the block *functions*. A variety of physical quantities could be analyzed by function objects, as average temperatures of the air flow or heat flows via surfaces or interfaces (see Fig. 6).

```
T_outlet_mass
{
        type            surfaceFieldValue;
        libs            (fieldFunctionObjects);
        writeControl    writeTime;
        log             yes;
```

```
        writeFields     no;
        regionType      patch;
        region          airFlow;
        name            outlet;
        operation       weightedAverage;
        weightField     phi;
        fields          (T);
}

// Heat flow at lowerWall
myHeatFlow_lowerWall
{
        type            wallHeatFlux;
        libs            (fieldFunctionObjects);
        writeControl    writeTime;
        log             yes;
        writeFields     yes;
        region          chipHX;
        patches         ("lowerWall");
}
```

Fig.6 function objects in ./system/controlDict

The output of your function objects you can find in the log-file log.chtMultiRegionSimpleFoam or in the sub-directory *postProcessing/*.

```
Solving for solid region chipHX
DICPCG:  Solving for h, Initial residual = 0.0108167, Final residual = 0.000104534, No Iterations 74
Min/max T:408.465 456.197
ExecutionTime = 287.68 s  ClockTime = 288 s

surfaceFieldValue p_inlet write:
    areaAverage(inlet) of p = 100015

surfaceFieldValue p_outlet write:
    areaAverage(outlet) of p = 100000

surfaceFieldValue mdot_inlet write:
    sum(inlet) of phi = -0.0284567

surfaceFieldValue mdot_outlet write:
    sum(inlet) of phi = -0.0284567

    min/max/integ(airFlow_to_chipHX) = -2549.38, 52886.7, 351.891
wallHeatFlux myHeatFlow_chipSurface write:
    writing field wallHeatFlux
    min/max/integ(lowerWall) = 35200, 35200, 352
wallHeatFlux myHeatFlow_lowerWall write:
    writing field wallHeatFlux
surfaceFieldValue T_chipSurface write:
    areaAverage(chipHX_to_airFlow) of T = 430.912
```

```
volFieldValue T_chipHX write:
   volAverage(chipHX) of T = 438.156

surfaceFieldValue T_lowerWall write:
   areaAverage(lowerWall) of T = 447.017

surfaceFieldValue T_outlet_area write:
   areaAverage(outlet) of T = 320.222

surfaceFieldValue T_outlet_mass write:
   weightedAverage(outlet) of T = 305.954

writeObjects Further_IOObjects_airFlow write:
   writing object h
   writing object thermo:rho
   writing object wallHeatFlux
writeObjects Further_IOObjects_chipHX write:
   writing object h
   writing object thermo:rho
End
```

Fig.7 Output of function objects in the log-file log.chtMultiRegionSimpleFoam

In Fig.6 you can see the heat flow of Q = 200 W passing the surface *lowerWall* as well as the same heat flow passing the interface *airFlow_to_chipHX*. The correct outlet temperature of the passing air is T_{out} = 320.22 K.
For checking the results the heat absorbed by the passing air flow could be calculated by
$\dot{Q} = \dot{m} \cdot c_p \cdot (T_{out} - T_{in}) = 366.2$ W.
This is a deviation of 4 % to the input heat flow of 352 W. The deviation is due to the insufficient position of the outlet boundary, the insufficient mesh and the insufficient number of iterations (which have all been selected for the sake of achievemant of short turn around time).

Furthermore, temperatures, like average surface temperatures or average volume temperatures, could be analyzed by function objects.

Local temperature fields and the spatially resolved temperatures could be post-processed using ParaView.

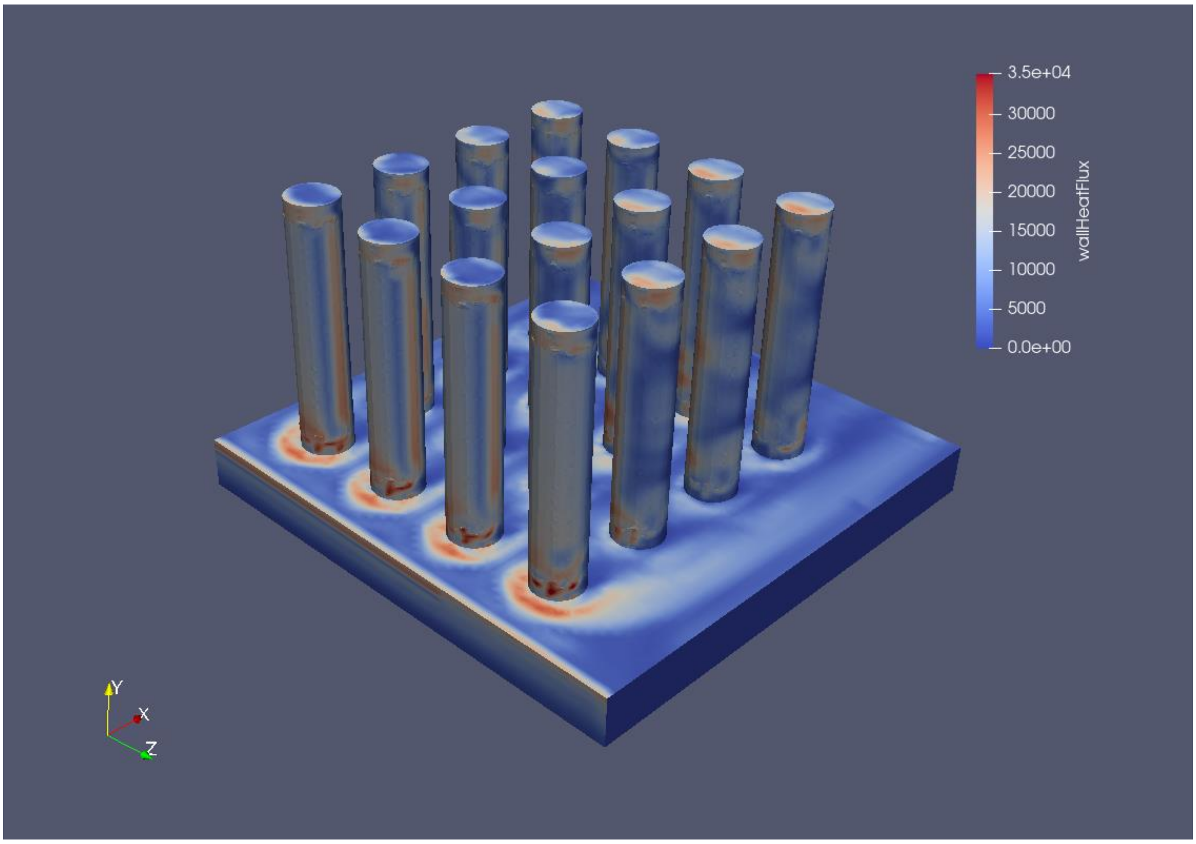

Fig.8 Wall heat flux at surface of pins and plate (in W/m²)

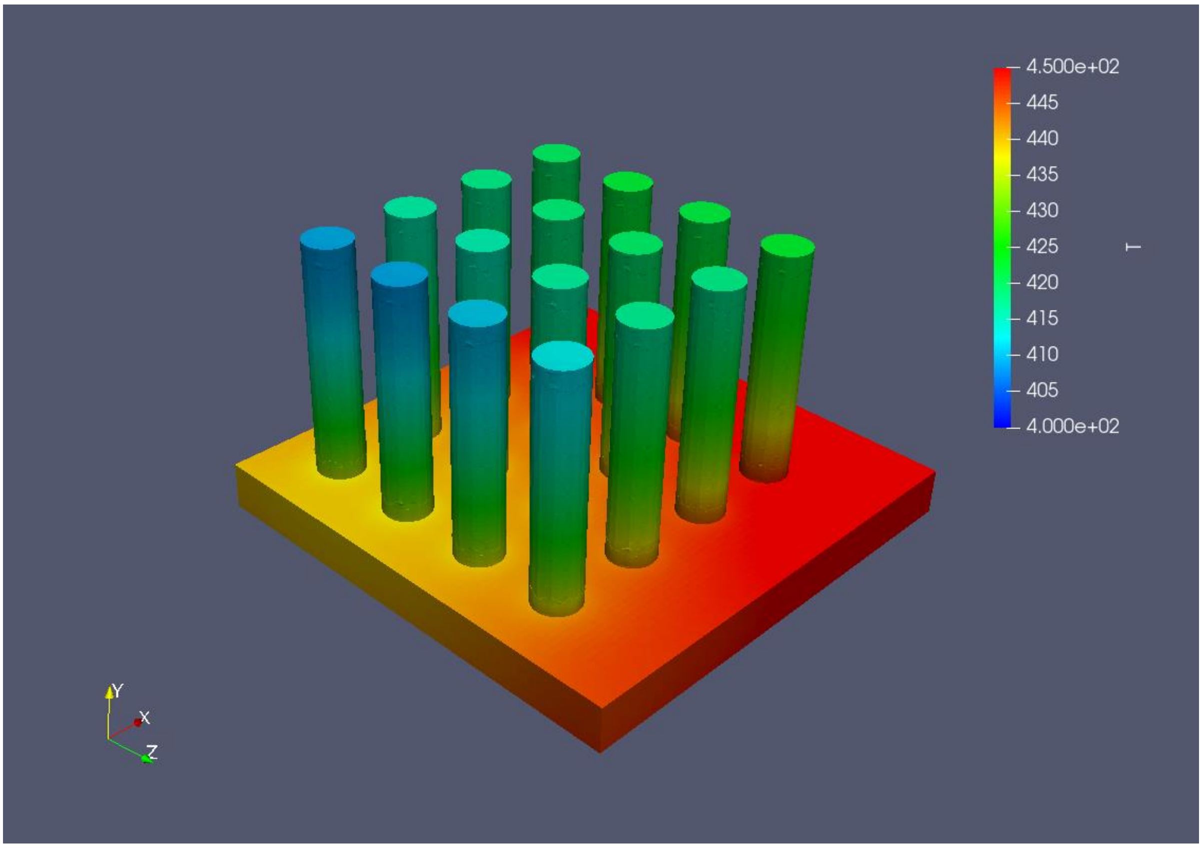

Fig.9 Temperature distribution at surface of pins and plate (in K)

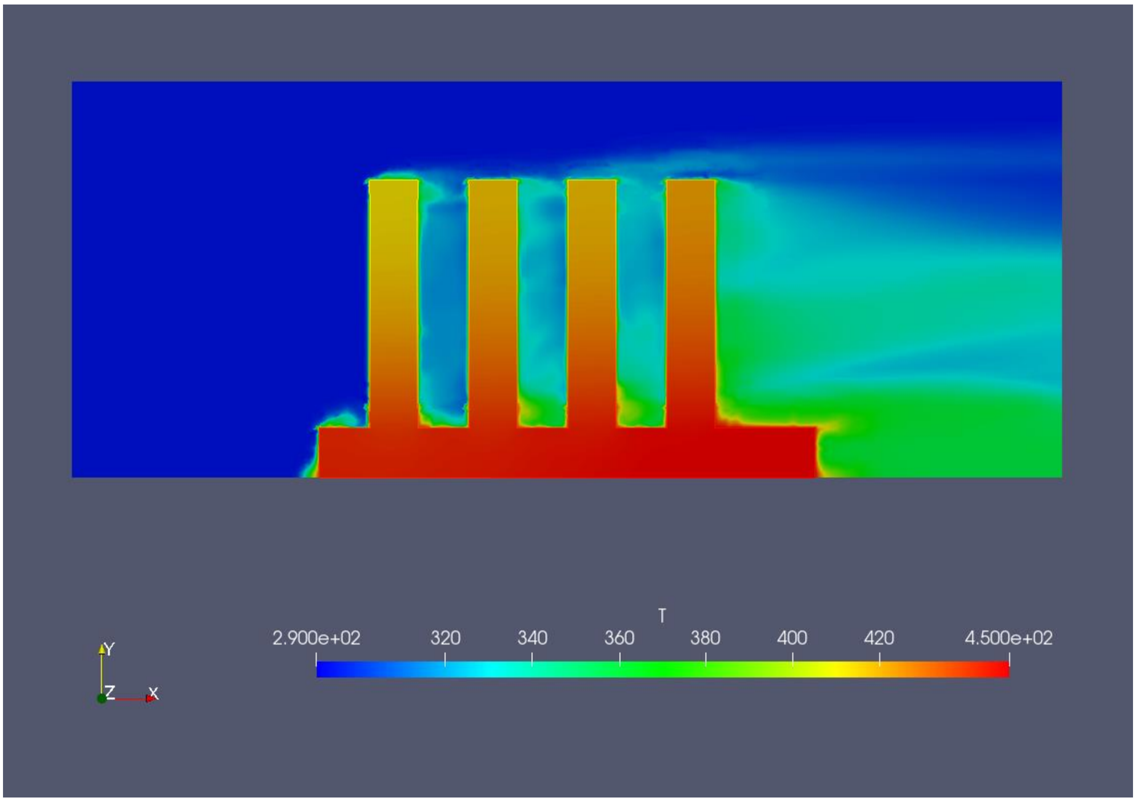

Fig.10 Temperature distribution of passing air flow (in K)

8. Setting solid material properties

In *./constant/chipHX/thermophysicalProperties* the material properties of the solid region chipHX could be specified.
You can substitute the pre-set values of the solid material aluminum with the material properties of stainless steel given in Fig. 11.

```
mixture
{
        transport
        {
                kappa  13;
        }
        thermodynamics
        {
                Hf      0;
                Cp      470;
        }
        equationOfState
        {
                rho     7980;
        }
}
```

Fig.11 Solid material properties of steel