This course material is free: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation. It is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

More details about the GNU General Public License can be seen at: http://www.gnu.org/licenses/>.

Tutorial: BwUniCluster 2.0/HoreKa Hot Radiation Room (parallel)

In this tutorial we will learn about the simulation of radiative heat transfer in OpenFOAM on multiple cores by submitting a job. We will do it using the example hotRadiationRoom from the series of built-in OpenFOAM tutorials.

1. Tutorial Case

First, we should copy the tutorial case from the FOAM_TUTORIALS directory and paste it into our workspace (denoted by a point):

\$> cd <workdir>

\$> module load cae/openfoam/v2106-impi

\$> source \$FOAM_INIT

\$> cp -R \$FOAM_TUTORIALS/heatTransfer/buoyantSimpleFoam/hotRadiationRoom/ .

 $\label{eq:spectrum} $ \ cp \ FOAM_TUTORIALS/multiphase/compressibleInterFoam/laminar/depthCharge3D/system/decomposeParDicthotRadiationRoom/system $ \ hotRadiationRoom/system $ \ hotRadiationRoom/s$

2. Creating the Mesh

The mesh in the hotRadiationRoom tutorial represents a 10m x 6m x 2m room equipped with a heat source in the corner of its floor.

In order to create the mesh using data from the blockMeshDict-file, we will use the following command:

\$> cd hotRadiationRoom \$> blockMesh

Apart from describing the basic structure of the mesh, the blockMeshDict file also defines boundaries, also known as patches, for which special conditions will be set. In our case these areas are: floor, ceiling, fixedWalls and box. To view the file you can use the command:

\$> more blockMeshDict

The mesh is shown in Fig. 1. Figures 2, 3 and 4 show the different boundary patches.



Fig.1 The hotRadiationRoom mesh created using blockMesh



Fig.2 Boundary patch "fixedWalls"

Fig.3 Boundary patch "ceiling"



In this case there will be a heat source with the temperature of 500 K in the box patch:



Fig.5 The heat source in the box patch

Since we will be conducting parallel computation, we also need to decompose our mesh into subdomains using the following commnad:

\$> decomposePar

This creates a directory for each of the eight processors (cores).

We want to conduct the simulation on 8 cores, therefore our decomposeParDict-file should look somewhat like this:

Now it is time to submit the job for calculation. Assuming we are conducting the simulation on a multiple cores the batch job file (name e.g. my_dev_multi_job.sh) should look something like this:

#!/bin/bash		header of the file
#SBATCHpartition dev_multiple #SBATCHnodes=2 #SBATCHntasks=8 #SBATCHtime=00:30:00 #SBATCHmem=8000mb #SBATCHjob-name=hotRadiationRoom		job will be submitted to queue dev_multiple two nodes will be used ntasks means ,,the number of cores" to use the job will run for maximum 30 minutes the job may use max. 8 Gb this name is given by the user
module purge module load cae/openfoam/v2106-impi source \$FOAM_INIT mpirun -n 8 buoyantSimpleFoam -parallel	}	loading OpenFOAM on the execute core the name of the OpenFOAM-solver

Note!: On HoreKa use dev_cpuonly instead of dev_multiple

We can then submit the job using the name of the job-file with the following command:

\$> sbatch my_dev_multi_job.sh

After the simulation has been completed we can combine the results from different cores into one using the command:

\$> reconstructPar

We can also continue without reconstructing. In that case we will be able to visualize results from different processors (decomposed case) as well as the reconstructed case in Paraview.

3. Results

After the simulation has been completed we can download the results onto our personal computer and visualize them in Paraview. This is the recommended way for visualizing results with small- or medium-sized grids which is the most common case.

We can now examine the steady state of the heat transfer in the room which has been achieved after 935 iterations. The following images depict the central plane of the control room from the top-down view with the box patch located in the bottom left corner:







As one could have expected, the heat transfer caused by radiation is prominent exclusively in and above the box patch due to absorption of the medium. The remaining heat transfer as well as the mass transfer take place due to convection.



Another place worth looking at is the Y-normal plane going through the box patch:







As expected, the radiation coming from the box causes the air arround it to heat up. Then, due to convection, the hot air moves towards the ceiling where it cools down which drives the circulation in the room.

Legend

a	Absorption coefficient	[1/m]
alphat	Turbulent thermal diffusivity	[m^2/s]
epsilon	Turbulent kinetic energy dissipation rate	[J/kg/s]
G	Incident radiation intensity	[W/m^2]
qr	Radiative heat flux	[W/m^2]
phi	Mass flow	[kg/s]
k	Turbulent kinetic energy	[J/kg]
nut	Turbulent viscosity	[m^2/s]

The names in the leftmost column are the variables in the 0 directory.