Forced-air cooling of a heatsink

Application example coupled simulation with Code Saturne and SYRTHES

Brennan Sharp

Introduction

An electronic device producing heat during normal operation has an aluminium heatsink as part of the thermal management design. The heatsink fits into a cooling duct, through which ambient air is forced by a fan.

To conduct heat away from the electronic device quickly, it is mounted on a small copper plate, which is in turn attached to the base of the heatsink.

In reality, the air from the fan flows through a coupling, however for the purpose of demonstrating a coupled simulation, this additional geometry is omitted and the air speed blowing into the fluid domain of the simulation is uniform across the boundary plane.

The fluid domain outlet is surrounding air, which could also be included in the model, but again is omitted to retain focus on the coupling of the simulations.

Code Saturne can solve steady and unsteady flows, whereas SYRTHES works only with transients, so to find a steady state coupled result, the fluid part is set up as an unsteady flow.

The heatsink geometry was prepared using a third-party mechanical drawing software and transferred to Salome Platform to be prepared for the simulators, although Salome Platform can also be used to create geometry.

The mesh files are provided, but not the geometry. The meshes can be viewed by importing them into the Mesh module of Salome Platform.

For the preparation and descriptions that follow, I am using Code Saturne v2.00rc2, SYRTHES 3.4.2(5) and Salome Platform v5.1.4

Where I say to do something, you should see that it has been done already, so most times I am just noting what I would do and you need only observe what I've done.

Code Saturne must be built with SYRTHES support, which possibly requires SYRTHES to be installed before building Code_Saturne building.

Four cases are given for the study. Case 1 is a sanity check for the fluid flow, without SYRTHES coupling. In Case 2, coupling with SYRTHES is included. Boundary conditions are constant and we wait for a steady state solution to emerge. In Case 3, the heat flux changes in time, representing the situation where the electronic device is switched on and off. Case 4 is the fluid-domain counterpart to case three. The inlet fluid flow is stopped (or changed) for a while to see what happens to the temperatures in the solid domain.

The Code Saturne + SYRTHES studies are set up using the Code Saturne hierarchy. The SYRTHES-related files are simply added as needed.

Lastly, my background in fluid dynamics is minimal and long faded, so if you have questions or comments about my choice of settings relating to the physics, be gentle. The point of these notes is to illustrate my setup for coupled Code Saturne and SYRTHES simulations.

Input Meshes

In the Mesh module of Salome Platform import the MED mesh files from the Code_Saturne study MESH subdirectory.

Solid Domain

Select the object labelled 'solidDomain' and display it alone in the graphical window. Expand the object hierarchy so that the names of the grouped parts alre listed.

The heatsink is initially an extrusion. A rectangular open cavity has been made the base. The copper plate sits in that cavity in perfect contact with the heatsink body. A small rectangular area is visible on one face of the copper plate. This face will have heat flux applied due to the electronic device.

The solidDomain geometry has four named groups. 'heatsink' is the aluminium extrusion with the rectangular cavity. 'insert' is the copper plate. 'heatedFace' is the rectangular area where heat will be applied. 'fluidInterface' is the group of heatsink faces that are exposed to the fluid domain and which will therefore be a part of the coupling between Code Saturne and SYRTHES.

SYRTHES requires ten-node tetrahedrons, so the mesh for the solid domain is made only of tetrahedra and have been converted to quadratic.

The mesh density near the heated face has been increased because of the large thermal gradients in that region.

SYRTHES accepts a small number of mesh formats, though not MED directly. There is a conversion utility that will convert MED to SYRTHES format. Conversion can take quite some time and appears to be exponentially related to the number of mesh elements.

When it finishes, the converter also creates a description file that correlates SYRTHES mesh groups with the groups that were defined in Salome Platform.

Fluid Domain

In the Mesh module of Salome Platform select the object labelled 'fluidDomain' and display it alone in the graphical window.

The geometry of the fluid domain is basically the space in the cooling duct not that isn't occupied by the heatsink. There are four boundary face groups defined: inlet, outlet, adiabaticWalls, and solidInterface. The face group labelled solidInterface is in 100% contact with the group labelled fluidDomain in the solidDomain.

The fluid domain mesh is an extrusion of triangles. An extrusion of quadrilaterals could be used, however triangular inlet and outlet face meshes seem to be of better quality and easy for minimal set-up effort.

I am not aware that there is any need for the fluid and solid domain interface to have conforming surface meshes, although there may be a limit to how disparate they can be.

Code Saturne will use MED meshes without conversion (and no requirement for quadratic elements).

CASE 1: Non-coupled Code Saturne Setup

It is generally a good idea to set up a non-coupled simulation in the first instance, to check fluid flow. The first case is therefore a non-coupled fluid domain setup.

Steady flow

As this is not coupled with SYRTHES, there is a choice of steady or unsteady flow setup. Steady flow was chosen for this case because then Code Saturne can find the peak fluid velocity without the user iteratively trying different time steps to get convergence without excessive iterations.

From the known maximum velocity a proper time step that keeps the Courant number < 1, can be calculated and used in the subsequent unsteady flow simulations.

For the turbulence model I've used k-epsilon. I've used the scalable wall function because of the fineness of the mesh at the boundary layer.

In the physical properties section I've used the defaults for this demo as they seem close to the values I normally use for air.

The initial volume is set to 3.0m/s, to match the inlet boundary velocity and hopefully get to the steady state solution faster.

Definition of boundary conditions is initiated by importing the pre-processor log for the fluid mesh.

The original labels were edited to match the names in the selection criteria, which in turn match the names of the groups in the Mesh. Also edit the boundary types.

In Boundary Conditions, only the inlet boundary needs editing. The inlet velocity is set to 3.0m/s and the hydraulic diameter for turbulence is set to 5mm, due to the heatsink fins defining multiple deep and narrow channels, about 6mm wide.

In steady flow management, the number of iterations has been increased to 100. That turns out to be ample.

In equation parameters, the solver precision has been reduced to 1e-5, as per suggestion on the Code Saturne forums.

Under prepare batch calculation, set the computer type and number of processors to suit your circumstances. My Code Saturne is built to take advantage of multiple processors in a simple Open MPI cluster and works as per the supplied settings.

Also check under advanced options that your temporary directory is set how you want it. I have a designated temporary directory, so you'll almost certainly want to change this.

Save the file and run the simulation.

While it's running, monitor the listing file in the temporary directory. I use tail -f for this. For my setup, this will be something like

```
tail -f /home/cluster_tmp/tmp_Saturne/heatsink/.CASE1.12091528/listing
```

Monitor the variables to see that things are running sensibly and that the variables are converging. It is evident that the convergence occurs well before 100 iterations.

From the last iteration in the listing file, the maximum velocity is 3.71m/s, in the Z direction (the direction of extrusion). This information is used in subsequent unsteady flow cases.

Unsteady flow

To check that the calculated time step is indeed suitable, the analysis is repeated as an unsteady flow, even though the flow will become steady.

The maximum steady state fluid velocity was observed to be 3.71m/s. All the prisms in the fluid mesh are 7.5mm long, so to satisfy the Courant condition, the time step for unsteady flow analysis should be no greater than 2.02ms, so a time step of 2.0ms seems good.

The advection time, based on an average Z-axis velocity of 3m/s is (300mm)/(3m/s)=100ms.

Allowing ten times the advection time to get close to steady state flow, the number of time steps needed is 1s/2.0ms=500 iterations.

It was observed that the maximum Courant number for such a time step was 1.05. Various possible reasons can be thought of, however the simple and inexpensive solution is still to reduce the time step by 5%, to 1.8ms and use more iterations.

In the CASE1 directory there is another file called unsteady.xml. It has been set up as a copy of steady.xml, but the calculation type has been changed and the time steps and iterations have been filled in with 1.8ms and 600 respectively.

CASE2: Coupled simulation with static boundary conditions

Before starting the description of the SYRTHES coupling setup, it is convenient to consider the relative time scales of the fluid and solid domains.

The fluid flow analysis has been found to work well with the 1.8ms time steps. The time required for the heatsink to reach a steady temperature state is measurable in hours. Time steps of 1.8ms in the solid domain would require millions of iterations. A reasonable time step for the solid domain analysis might be in the range of 2 -- 10s. Let's suppose a 5s time step is sufficient and that the thermal transient will settle in two hours. That means 1440 iterations.

Every time step in the Code Saturne simulation coupled with SYRTHES is synchronised with one time step in SYRTHES, so the plan is therefore to use 1440 iterations in both fluid and solid domains, with 1.8ms and 5.0s time steps respectively.

For simulations like this, I usually allow additional fluid iterations anyhow, corresponding to about twice the advection time. That additional time is covered by the difference of 600 and 1440 iterations in this case.

To couple Code Saturne and SYRTHES

1. Code Saturne must have been built with SYRTHES support.
2. SYRTHES case file hierarchy must be embedded in the Code Saturne case hierarchy
3. Code Saturne must be told about it. This is done in a file called ussyrc.f90

Code Saturne setup

Set up the Code Saturne unsteady flow analysis as previously. The definiton of two wall-type fae groups in the mesh permits the distinction between coupled faces and non-coupled faces when defining wall-type boundaries in Code Saturne.

To the Code Saturne CASE directory, add DATA_SYR and SRC_SYR directories, counterparts to the existing DATA and SRC directories.

The solid-domain mesh can be put in the existing CS MESH directory with the fluid mesh.

In the CASE2/DATA directory the file static.xml is based on the unsteady flow setup of CASE1. The differences:

- The number of iterations that has been changed to 1440 as explained already
- Now use a temperature (celsius) thermal model
- Additional fluid properties can be set, though I've used the defaults
- Initial volume temperature must be set.
- Inlet boundary now has a temperature specified (in this case matching the initial volume temperature)

- The other boundary conditions also have thermal properties to set. These have been left as default.
- In equation parameters there is a new parameter. I leave it alone except for setting the precision.

Copy ussyrc.f90 from CASE2/SRC/REFERENCE/base/ to CASE2/SRC/. This FORTRAN code implicitly tells Code Saturne to use SYRTHES, and specifies what fluid boundary is involved, and the solid mesh it interfaces with. In the SYRTHES setup, SYRTHES will be told what boundaries are coupled.

SYRTHES setup

In the MESH directory, the 'convert2syrthes' utility was run on Mesh_solid.med. The result is Mesh_solid.syr and Mesh_solid.syr_desc, the former being the mesh in SYRTHES format, the latter being a mapping of SYRTHES group references to MED groups. In the SYRTHES setup, the SYRTHES group numbers will be used, not the MED group names, so the map file is important. The conversion utility is downloadable from EDF's SYRTHES pages.

In the CASE2/DATA_SYR/ directory there are two files: syrthes.env and syrthes.data.

syrthes.env WILL need to be edited if you want to use this demo because the paths to the study files will be different, however all that should need changing is AMONT, SUITE, and AVAL

The template syrthes.data has been edited. The SYRTHES documentation describes the settings in this file, however here is a discussion of anything special relevance to this analysis.

```
'DIMENSION DU PROBLEME=' 3                         / sets 3D for problem dimensionality
'AXE D AXISYMETRIE (AUCUN,OX,OY)=' 'AUCUN'         / there is no symmetry

'PAS DE TEMPS SOLIDE='  5.                         / the size of the time step
'NOMBRE DE PAS DE TEMPS SOLIDES=' 1440             / the number of time steps (matches the number in Code Saturne)

'PAS DES SORTIES CHRONO SOLIDE='    72             /  Every 72 iterations (6minutes simulated) , results will be
written to the chronological results
'PAS DES SORTIES LISTING SOLIDE='   -1             / The listing file will have every time step.


'NOMBRE ITERATIONS SOLVEUR SOLIDE=' 1000           / Maximum solver iterations per time step
'PRECISION POUR LE SOLVEUR SOLIDE=' 1.E-6          / Solver precision


/'REFERENCES NOEUDS OU FACES SOLIDES COUPLE(E)S' 4  /  Syrthes face reference 4 is fluidInterface in the MED
file (refer back to the Mesh_solid.syr_desc file). This face group is coupled to fluid domain.

'REFERENCES NOEUDS OU FACES SOLIDES AVEC FLUX' 5  /  equivalent to heatedFace in MED file, this face has heat
flux

'CINI'           20                 -1  / Initial temperature is 20 Celsius everywhere

'CLIM'    'FLUX'          1.5625e6     5   /  Face reference 5 has flux of 1.5625MW/m^2

/ Aluminium
 'CPHY'     'RHO'          2700.       6    / Ref group 6 is the heatsink elements (aluminium)
 'CPHY'     'CP'           900.        6
 'CPHY'     'K ISOTROPE'   200.        6
/ Copper insert
 'CPHY'     'RHO'          8960.       7    / Ref group 7 is the insert elements (copper)
 'CPHY'     'CP'           385.        7
 'CPHY'     'K ISOTROPE'   385.        7
```

You may choose to other output, such as history on selected nodes, however the SYRTHES setup is now complete.

And the coupled simulation setup is also complete.

You run the coupled simulation from the Code Saturne GUI (or command line if you prefer). Code Saturne takes care of launching SYRTHES.

While the simulation runs, I monitor the fluid and solid listing files (listing and listsyr respectively) to check that things run as intended.

As the thermal transient settles, the iterations will get faster, almost reaching the simulation speed of CASE1.

CASE3: Thermal cycle

This is a simple variation of CASE2 in that the heat flux applied in the solid domain is changed at a set time during the simulation.

The syrthes.env file is updated to refer to the CASE3 directory, but syrthes.data is not changed.

The heat flux change (or boundary condition changes in general) is implemented quite simply in a fortran file called limfso.F.

The template for this file is in the SYRTHES installation. It is copied into CASE3/SRC_SYR/ and changed to suit.

In this example, the heated device produces 1.5625MW/m^2 heat flux when operating. It will run for one hour and then be switched off. The operating flux was originally used in the syrthes.data file. The controls applied in limfso.F will take precedence.

In section 2 of limfso.F on lines 197--203 you will see the time dependent condition is specified. for face reference 5 (heatedFace)

This is all that is required to make the heat flux change. You can make more complex changes if you wish, perhaps applying flux at different times and places.

CASE4: Interrupted fluid flow

Another simple variation on CASE2, this time changing the fluid inlet boundary condition, such that the inlet fluid flow is temporarily stopped.

The inlet velocity is initially 3.0m/s. After one hour (in solid domain), the inlet velocity abruptly drops to zero for a period of ten minutes before being restored.

Lines 533 -- 554 define the boundary condition in terms of fluid domain times. One hour solid domain is 720 iterations, which is 1.296s in fluid domain.

One hour and ten minutes is 840 iterations in solid domain, so 1.512s in fluid domain.

Note that in this example I did not consider the advection time in my timing calculations. The advection time was estimated earlier to be 100ms. The fluid inlet flow was stopped for only 216ms, so the effect is that heatsink is being blown the entire time. To achieve a realistic stopping of flow I would use extra iterations in the fluid domain and change the time scale in SYRTHES for the period of interruption, but that's going further off topic.

Post-processing the results

The text output can be processed as you wish.

I configured Code Saturne to produce post-processing data in MED format, which I can use directly in Salome Platform.

SYRTHES produces data in its native format. There is code with SYRTHES to convert to MED or

Ensight, however the last time I tried building the syrthes2med converter was unsuccessful, so I use a less convenient but workable method:

syrthes2ensight (from the SYRTHES code)

ensight2med (included with Salome Platform)

You can get Code Saturne to give temperature information from the solid domain, but this appears to be limited to the coupled boundary.

The end.