Enrico Nobile, Riccardo Zamolo
DIA - Dipartimento di Ingegneria e Architettura
Università degli Studi di Trieste

Esercitazioni di Termofluidodinamica Computazionale

Prediction of the thermal performance of a microchannel heat sink using ANSYS WB and Fluent

November 2018 (*reviewed May 2020*)

1 Abstract

In this tutorial it will be explained how to compute the thermal performance of a microchannel heat sink using ANSYS Workbench 18.0 (hereafter WB) and ANSYS Fluent 18 (Fluent for the sake of brevity) as CFD solver. The problem has been described by D. Liu and S.V. Garimella in [1], where they also performed an optimization of the device using several simplified, closed-form analytical models with increasing level of complexity.

2 Introduction

With heat dissipation levels of microelectronic components reaching 500 W/cm^2 (e.g. 5×10^6 W/m^2) and beyond, conventional air cooling systems are inadequate for removing excess heat. The research goal in this field is the development of more innovative, reliable and energy-saving chip cooling techniques. The ultimate goal is to reduce thermal resistance from the chip junction to ambient, and keep the chip's junction temperature as low as possible.
For high performance CPUs, GPUs, power amplifiers and other devices, air-cooling has proven ineffective at dissipating high heat fluxes. Heat transfer methods such as heat pipes, vapor chambers, nanomaterials, liquid cooling and miniature refrigeration systems have been attracting more interest.
Liquid-cooled microchannel heat sinks and *coolers* have been shown to be a very effective way to remove high heat load. A large heat transfer coefficient can be achieved by reducing the channel hydraulic diameter. In a confined geometry the small flow rate within microchannels produces laminar flow, which results in a heat transfer coefficient inversely proportional to the hydraulic diameter. In other words, the smaller the channels in the heat sink, the higher the heat transfer coefficient. This, however, is counterbalanced by an increase of the pressure drop. This will be shown next.

2.1 Hydrodynamic considerations

The previous statement can be easily proved by considering, for example, the equations for the fully developed (hydrodynamical and thermal) [1] flow and heat transfer in a cylindrical pipe in laminar regime under the assumption of constant heat flux boundary condition [2].
We first recall that the Moody (or Darcy) friction factor f for fully developed laminar flow in a cylindrical pipe is given by

$$f = \frac{64}{Re_D} \tag{1}$$

where Re_D is the *Reynolds number*

$$Re = \frac{\rho\, u_m\, D}{\mu} \tag{2}$$

[1] We recall that *fully developed flow and heat transfer* for an internal flow problem means, in practical terms, that both the friction factor f and the Nusselt number *Nu* no longer change with increased distance along the pipe.

in which ρ is the density of the fluid, u_m is the mean fluid velocity, D is the inner pipe diameter and μ is the dynamic viscosity.

For non-circular pipes, at least to a first approximation, eqn.(1) can be applied replacing the diameter D with the so-called *Hydraulic diameter* D_h

$$D_h \equiv \frac{4\,A_c}{P} \tag{3}$$

where A_c and P are the *flow* cross-sectional area and the *wetted perimeter*, respectively. It is this parameter that should be used in calculating the *Reynolds* and *Nusselt number*.

For a pipe of unit length, the pressure loss is given by

$$\frac{\Delta p}{L} = f\frac{\rho\,u_m^2}{2D} \tag{4}$$

where L is the pipe length.

Substituting (1) and (2) in eqn.(4), it results

$$\frac{\Delta p}{L} = \frac{128}{\pi}\,\frac{\mu\,\dot{V}}{D^4} \tag{5}$$

where $\dot{V}$ is the volumetric flow rate.

From eqn.(5) it results, therefore, that the pressure loss, for a given flow rate, decreases with the 4^{th} power of D. In other words, reducing the diameter of the pipe - or the *hydraulic diameter* for a pipe with a different cross-section shape - leads to a dramatic increase of the pressure loss.

2.2 Thermal considerations

What about the heat transfer ?
Assuming, for simplicity, fully developed flow and heat transfer, it results that, for *constant surface temperature*, the heat transfer coefficient h is defined such that:

$$q_{conv} = h\,A_s\,\Delta T_{lm} \tag{6}$$

where q_{conv} is the total heat flux, $A_s = P \cdot L$ is the heat transfer area and ΔT_{lm} - or *LMTD* - is the *logarithmic mean temperature difference*:

$$\Delta T_{lm} = \frac{\Delta T_o - \Delta T_i}{\ln\left(\Delta T_o/\Delta T_i\right)} \tag{7}$$

where

$$\begin{aligned} \Delta T_o &= T_s - T_{m,o} \\ \Delta T_i &= T_s - T_{m,i} \end{aligned} \tag{8}$$

In eqn.(8) T_s is the (constant) surface temperature, and $T_{m,i}$ and $T_{m,o}$ are the mean fluid temperature at the inlet and at the outlet, respectively.
We remind that the *mean* (or *bulk*) *temperature* is defined as

$$T_m = \frac{\int_{A_c} \rho\, u\, c_v\, T \ \mathrm{d}A_c}{\dot{m}\, c_v} \tag{9}$$

where $\dot{m}$ is the mass flow rate, and c_v is the specific heat at constant volume.
For incompressible flow in a circular tube with constant c_v, eqn. (9) simplifies in

$$T_m = \frac{2}{u_m\, r_o^2} \int_0^{r_0} uTr \,\mathrm{d}r \tag{10}$$

where $r_0 = D/2$.

For the case of *constant heat flux*

$$q_{conv} = q_s'' \,(P \cdot L) \tag{11}$$

where q_s'' is the constant surface heat flux.

For the case of a circular pipe with steady, laminar flow regime and constant surface temperature, the *Nusselt* number is

$$Nu_D \equiv \frac{h \cdot D}{k} = 3.66 \qquad (T_s = const.) \tag{12}$$

while for constant heat flux is

$$Nu_D \equiv \frac{h \cdot D}{k} = 4.36 \qquad (q_s'' = const.) \tag{13}$$

where k is the thermal conductivity. From (12) or (13)

$$h = \frac{k \cdot Nu_D}{D} \tag{14}$$

and, since Nu_D is constant, the heat transfer coefficient h is inversely proportional to D or, in general, is inversely proportional to D_h.

2.3 Splitting a single large channel in several smaller ones: is it convenient ?

Then, we may ask ourselves what happen if we split a single *large* channel in n *smaller* channels maintaining the same overall cross-section area and the same overall flow rate.
In the following we assume, for simplicity: circular pipe, fully developed laminar flow and constant temperature boundary condition.
We would like to find out, therefore, what happens if we split the original channel, with diameter D and volume flow rate $\dot{V}$, into n channels of diameter D_n - such that the overall cross-section is maintained - and volume flow rate $\dot{V}_n$ such that:

$$\begin{aligned} D_n &= D/\sqrt{n} \\ \dot{V}_n &= \dot{V}/n \end{aligned} \tag{15}$$

For the pressure loss, from (5)

$$\frac{\Delta p_n}{L} = n\,\frac{\Delta p}{L} \tag{16}$$

Therefore, replacing a single channel with n channels maintaining the same overall cross-section and flow rate, will increase the pressure loss by n. For the heat transfer, it is easy to show that the overall heat flux from the n smaller channels q_{conv_n} is given by

$$q_{conv_n} = n\,\frac{\Delta T_{lm_n}}{\Delta T_{lm}} \tag{17}$$

Therefore:

- If we maintain the same mean temperature difference ΔT_{lm}, e.g. $\Delta T_{lm_n} = \Delta T_{lm}$, the overall heat flux increases by n.
- If we are interested to have the same global heat flux, the mean temperature difference ΔT_{lm_n} reduces by n.

Other choices are however possible, like e.g. maintain the same mean temperature difference, the same overall heat flux and the same pressure loss, but reduce the length of the channels by n, thus obtaining a much more compact heat exchanger.

The use of microchannels as a viable cooling solution was first proposed in 1981 by Tuckerman and Pease [3], who designed and tested an integral, water-cooled heat sink by etching microscopic channels 50 μm wide and 300 μm deep on the silicon substrate. They reported achieving a high heat flux of 790 W/cm^2 with a temperature rise of 71°C above the inlet water temperature. Tuckerman's work was well received by the electronics community, and many extensive studies have since been conducted on different aspects of microchannels in electronic cooling.

Nowadays, microchannels heat sinks constitute an important building block for the energy-preserving approach of using hot water to cool HPC (High Performance Computer) systems [4].

3 Description of the problem

The problem studied in [1] is a microchannel heat sink, which is sketched in figure 1.

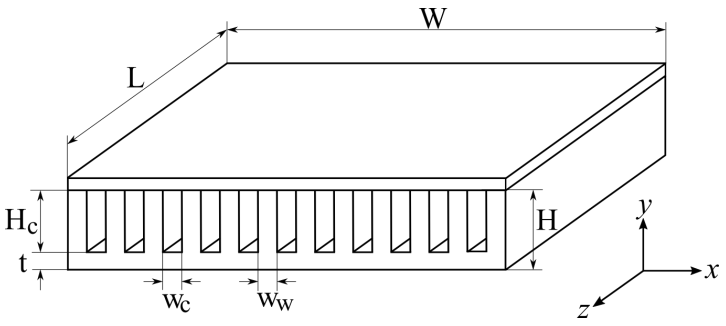

Figure 1: Scheme of a microchannel heat sink.

The thickness of the lid (top plate) is not indicated since, for conservative reasons, it may be assumed perfectly insulated and thus it will not be considered.

Among the cases considered in [1], we selected here the *Case 1*, which has been taken directly from the experiment in [3], thus allowing a reliable comparison between computed results and experimental measurements. For this case the geometrical and thermal characteristics are reported in table 1.

L [μm]	W [μm]	H [μm]	H_c [μm]	w_c [μm]	w_w [μm]	ΔP [kPa]	$q^{“}$ [W/cm^2]
10×10^3	10×10^3	533	320	56	44	103.42	181

Table 1: Geometric and functional data of *Case 1* in [1].

3.1 Computational domain

Since the width of individual microchannels and associated fins ($w_c + w_w$) is small compared to the overall heat sink width W, a large number of channels are distributed in parallel. For this reason, it is possible to consider, as a computational domain, just a single *repeating module* as depicted in figure 2, using appropriate symmetry boundary conditions.

4 ANSYS Workbench project

For this analysis we will use Fluent, together with ANSYS Workbench (WB for brevity).
Start WB and, as shown in figure 3, drag& drop a *Fluid Flow (Fluent)* component into the main window and rename it *microChannelHeatSink* as in Figure 3 and save this new project as File → Save as... and name it *microChannelHeatSink*.

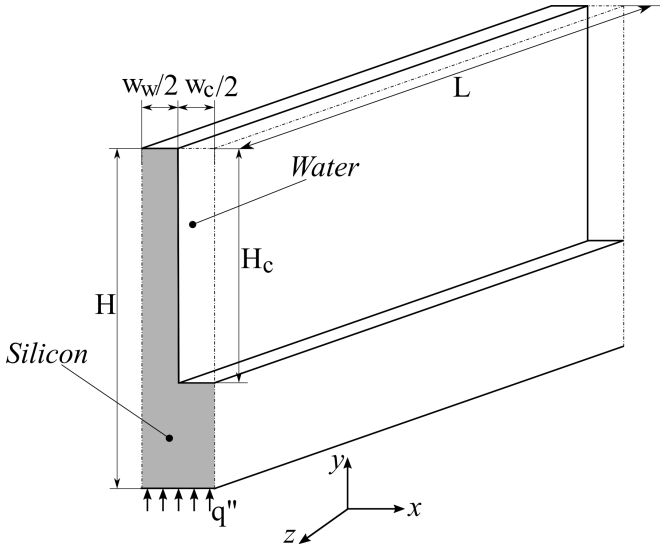

Figure 2: Computational domain for the microchannel heat sink.

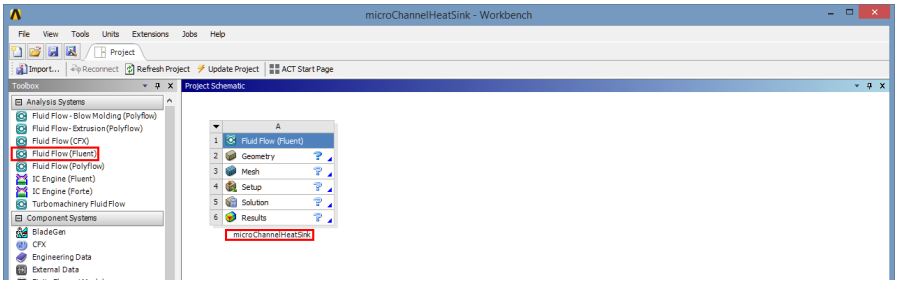

Figure 3: Workbench project - Fluid Flow (Fluent).

4.1 Geometry definition with Design Modeler

Within the Fluid Flow (Fluent) component in WB, right click on *Geometry* and click on *New DesignModeler Geometry...* to define a new geometry with Design Modeler (DM), as illustrated in figure 4.

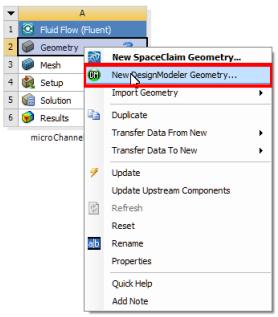

Figure 4: New geometry with DesignModeler.

Once the DM graphic window will open:

- Select the appropriate unit of length, in this case μm (micrometer), as in figure 5.

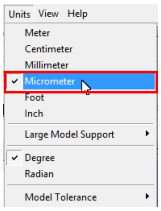

Figure 5: Selection of unit in DesignModeler.

- In the *Tree Outline* on the left, right-click on *XYPlane* and select *Look at* (figure 6).

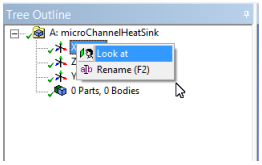

Figure 6: Selection of XY Plane in the Tree Outline of DesignModeler.

- Click on the *New Sketch* icon (), available on top, and right-click to rename the sketch as *Fin*. The DM window will look as in figure 7.

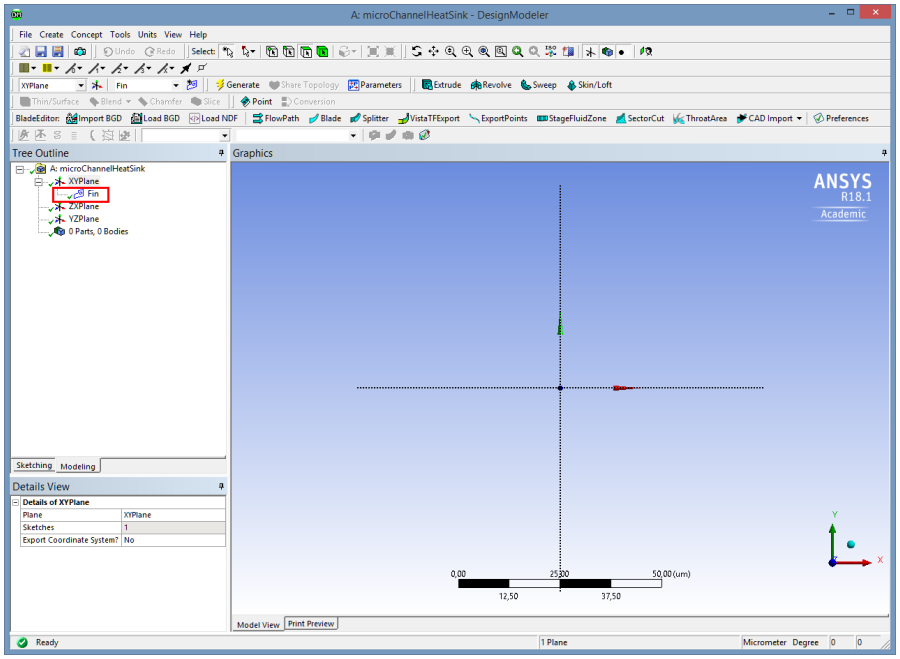

Figure 7: New sketch *Fin* in DM.

On this new sketch we have to draw the fin (solid) profile - see figure 2 - and extrude it to create the finned part of the heat sink:

- In the left tree, click on the *Sketching* tab and on the corresponding menu (the *Draw toolbox* is displayed by default when entering the Sketching mode) select *Line*, as in figure 8.

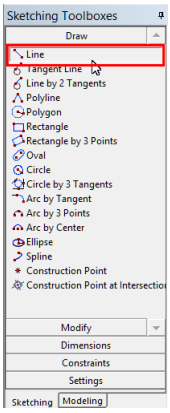

Figure 8: Selection of *Line* in the *Sketching* menu of DM.

- Starting, for convenience, at the origin of the axis, draw an approximate profile of the extruded finned profile (see fig. 2) illustrated in figure 9, paying attention to have all lines horizontal (the symbol *H* will appear while drawing) or vertical (the symbol *V* will appear while drawing).

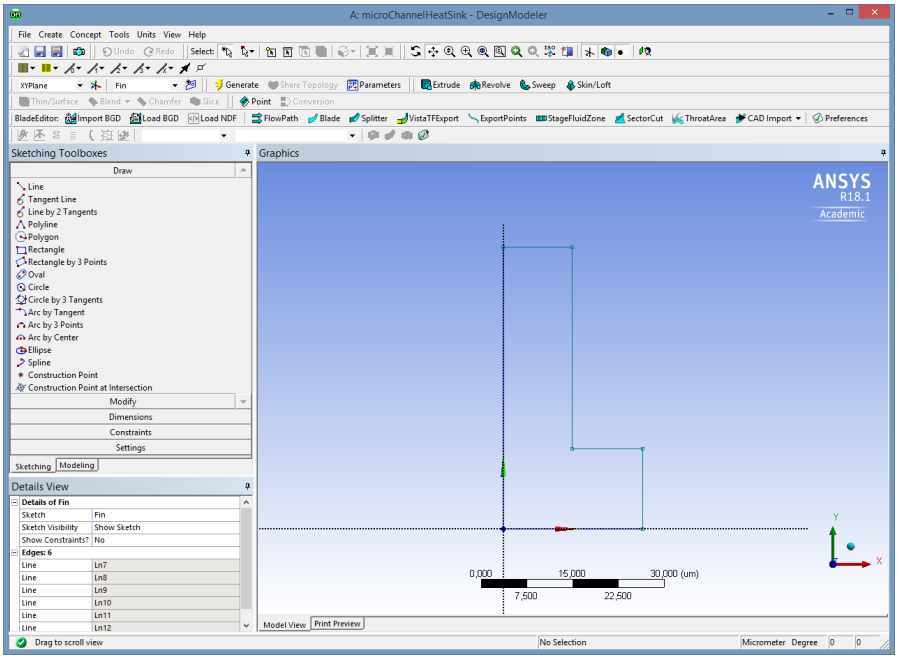

Figure 9: Rough sketch of the finned profile.

We now have to set the appropriate dimensions, as in table 1. For this:

- Click the *Dimensions* Toolbox in the Sketch menu, and select Horizontal ( ).

- Set the half-width of the water channel ($w_c/2 = 28\ \mu$m) and the half-width of the silicon fin ($w_w/2 = 22\ \mu$m), as in figure 10.

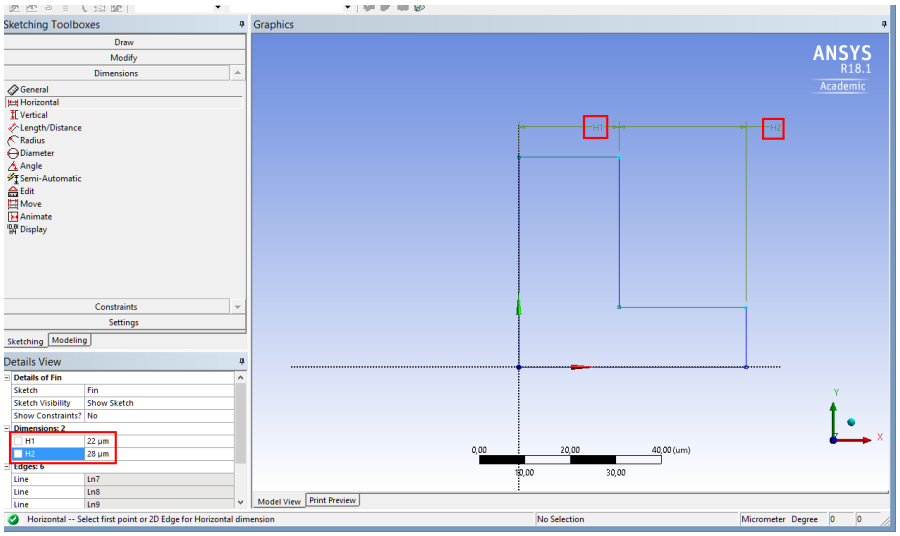

Figure 10: Set the horizontal dimensions of the finned profile.

- Within the *Dimensions* Toolbox in the Sketch menu, select now Vertical ( ), and set the dimensions of the overall height of the heat sink ($H = 533\ \mu$m) and of the height of the water channel ($H_c = 320\ \mu$m), like in figure 11.

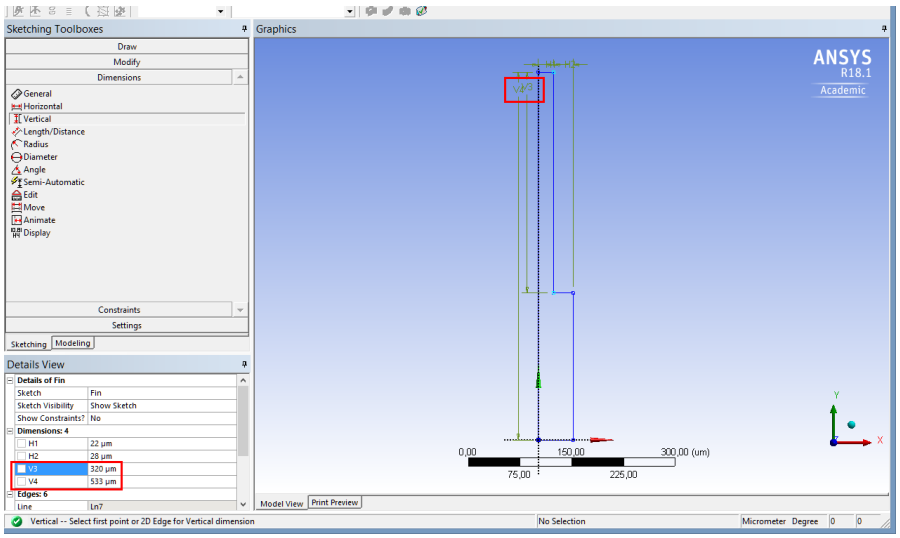

Figure 11: Set the vertical dimensions of the finned profile.

The next step is to *extrude* this sketch, in order to obtain the solid finned profile. Within *Modeling*, from the tree outline select the sketch to extrude (*Fin*) and:

- Click the *extrude* icon (Extrude) on the top.

- Rename it *FinnedChannel*.
- In the *Details Vew* menu for *Operation* select *Add Frozen* (it is necessary to select *Add Frozen* and not *Add Material* because, in this way, this solid will not be merged with other attached bodies which will be created later, like in this case the water channel).
- For the direction of extrusion select *Reversed*.
- For the Depth type *10000* μm.
- Click on the *Generate* icon (Generate).

The result is illustrated in figure 12, where also the previous choices are highlighted.

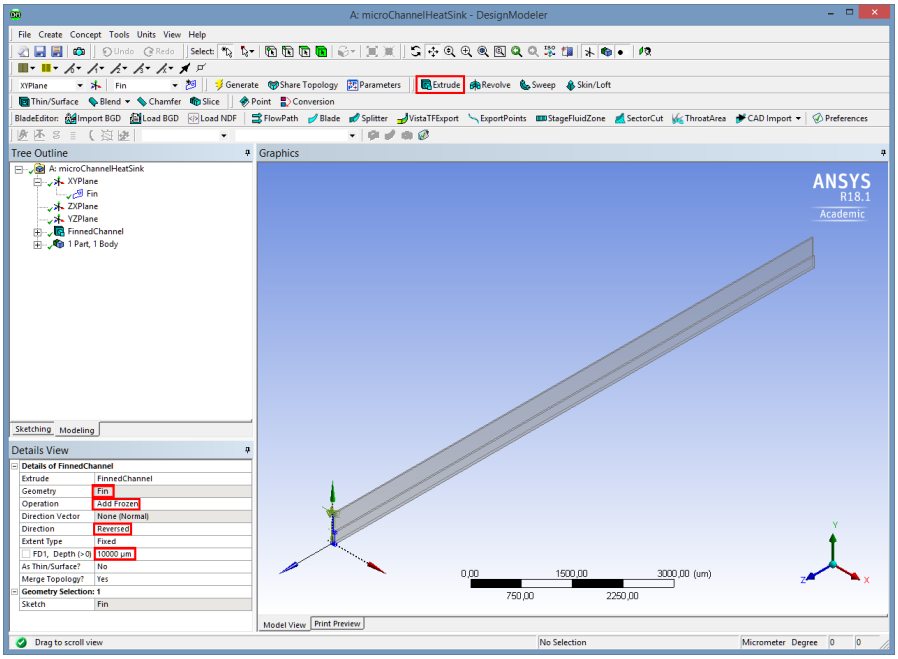

Figure 12: Extruded finned profile.

It is now necessary to create the water channel. In order to do this:

- In the *Tree Outline* on the left, right-click on *XYPlane* and select *Look at*.
- Click on the *New Sketch* icon (), available on top, and right-click to rename the sketch as *Water*.
- In the left tree, click on the *Sketching* tab and on the *Draw toolbox* menu select *Line*. Be sure to select the correct sketch where you want to draw.
- With the assistance of the snapping feature, draw the *rectangle* which represents the water channel, as depicted in figure 13 where, for clarity, the fin sketch has been hidden[2] (in the left tree, right-click on *Fin* and select *Hide sketch*).

[2] An easier way is to use the *Rectangle* tool () within the *Draw toolbox*.

- Extrude this sketch, as previously done for the fined channel, and rename it *WaterChannel*. Remember to select, also in this case, *Add Frozen* on the Details View, *Reversed* Direction of extrusion and again Depth of extrusion of 10000 μm. The result is shown in fig 14.

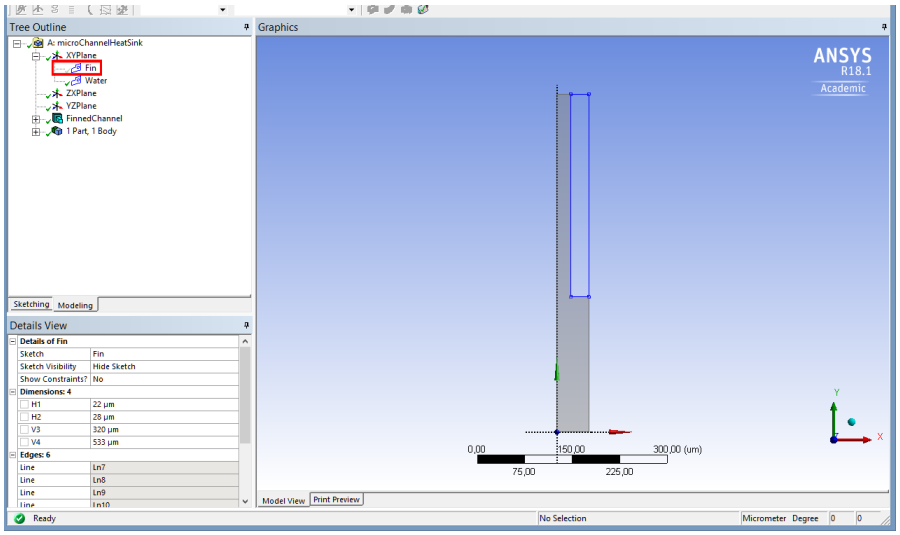

Figure 13: Sketch of water channel.

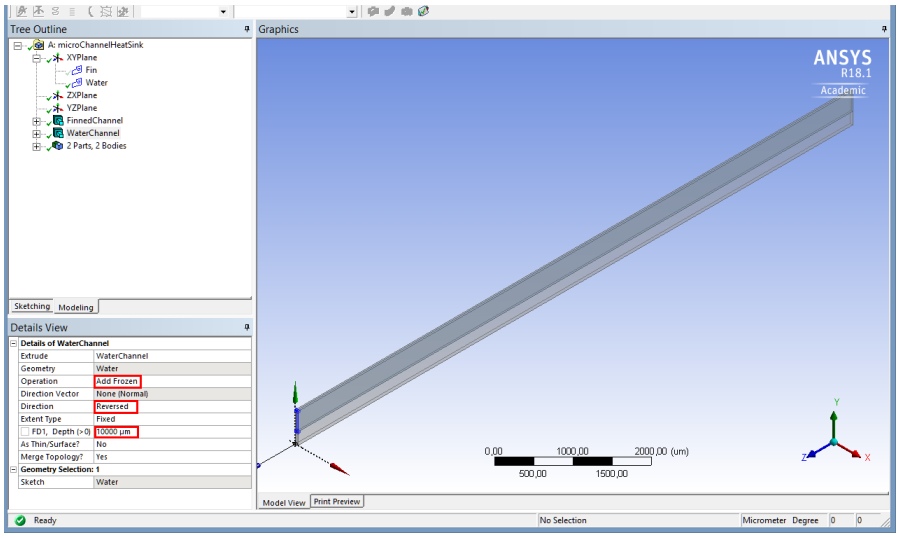

Figure 14: Extruded water channel.

An important step is now to create a *single part* from the the two *bodies*, e.g. the *FinnedChannel* and *WaterChannel*: in fact, after the creation of the *WaterChannel*, it should be noted that our computational domain is constituted, by default, by two *parts* and two *bodies*, as shown in figure 15. The fact that the two bodies belong to two different parts, means that the resulting mesh, at the *interface* of the two bodies, will be in general *non conformal*, e.g.

the nodes on the two surfaces will not match exactly, or at least with a tight tolerance. This leads to the need of some additional interpolation at the interface, which in turn can make the computation slower and less accurate.

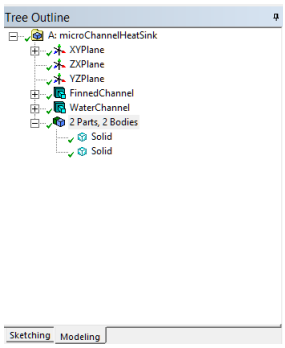

Figure 15: Parts and bodies by default in the Tree Outline.

In order to guarantee a *conformal* mesh at the interface, the bodies must belong to a single part. To do so, *shift-click* the two frozen bodies in the left-tree, and *right-click* to form a new part, as in figure16.

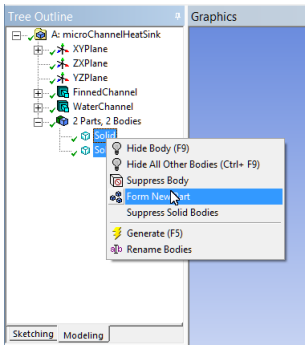

Figure 16: Merge all bodies in a single part in the Tree Outline.

The result will be like in figure17.

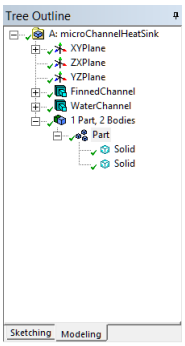

Figure 17: Single part in the Tree Outline.

Another useful step is to rename the two bodies as *Silicon* and *Water* and to select, in the Details View of the left tree, their Fluid/Solid property as Solid and Fluid, respectively, e.g. see figure18.

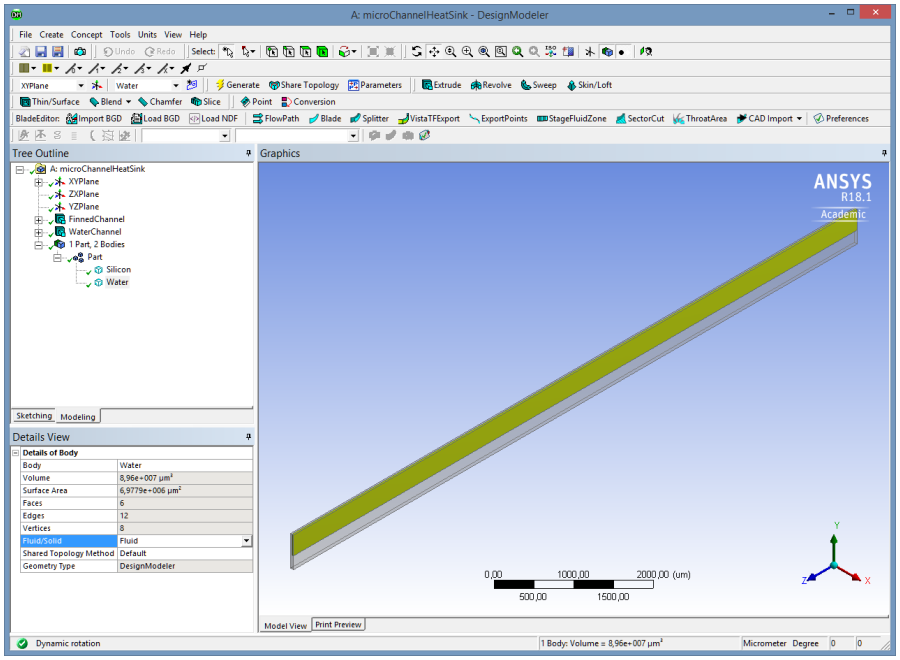

Figure 18: Rename bodies and set Fluid/Solid property.

The last task in Design Modeler is to *split* some of the edges in order to make a *purely orthogonal*, structured mesh in ANSYS Mesh. In fact, since the geometry can be thought as the sum of orthogonal hexaedra and is quite elongated in the main flow direction, it is extremely convenient to use a full Cartesian grid, as it is well known that this type of mesh provides better accuracy and faster convergence than any unstructured (e.g. tetrahedra) grids. In order to do this, we have to use, from the *Concept* menu, the command *Split Edges*, as shown in figure 19.

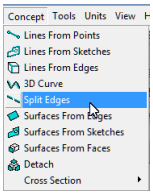

Figure 19: *Split Edges* command.

In the *Details Vew* menu select, among the different choices, *Fractional*, which specifies the ratio between the distance from the start point of the edge to the split location and the overall length of the edge. In order to determine the starting point of the edge, it is sufficient to activate, on the menu bar, the *Display Edge Direction* (see figure 20) option to display edges direction. The direction arrow, which appears at the midpoint of the edge, is proportional to the edge length.

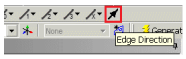

Figure 20: Display Edge Direction.

For the short edges at the base, it is necessary to zoom in, given their small length, as depicted in figure 21.

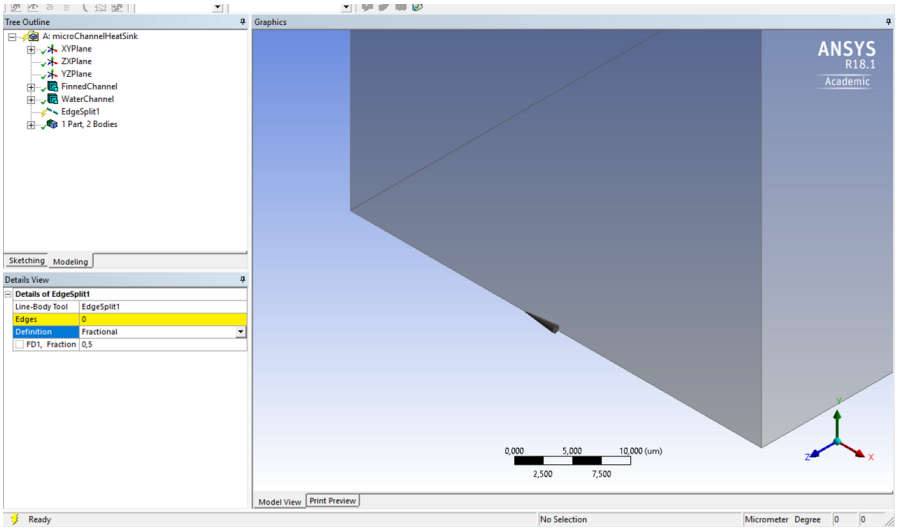

Figure 21: Direction of short edges at the base.

We have therefore to select the two short edges at the base as *Edges* within the *Details Vew* of the *Split Edges* command, and set the value of 0.56 as *Fraction* $(w_c/(w_c + w_w))$, as in figure 22.

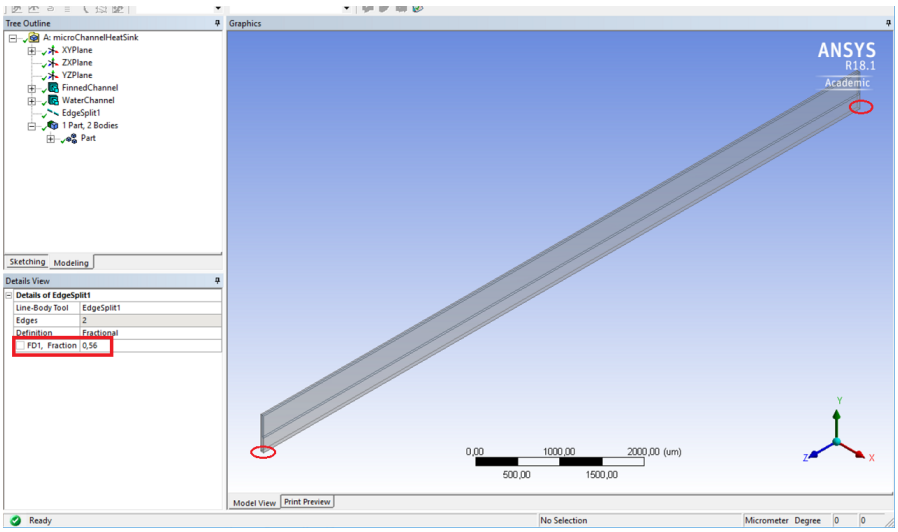

Figure 22: Short edges selection and setting of Fraction value.

Once this is set, click the *Generate* icon (Generate).

It is now necessary to do the same for the two longest vertical edges at the inlet and outlet sections, e.g. those indicated in figure 23.

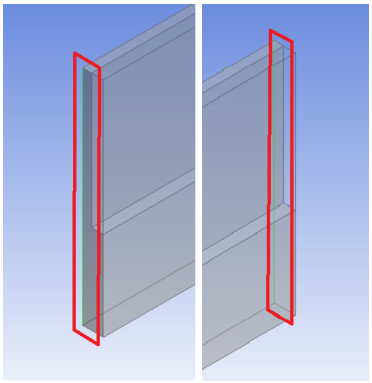

Figure 23: Long edges selection.

In this case we have to set the *Fraction* value as 0.3996, e.g[3] $((H - H_c)/H)$. Finally, it is convenient to rename the splits, in the Tree Outline, as *ShortEdgeSplit* and *LongEdgeSplit*.

Now we have to save our project, close DM and start the mesh generation process.

4.2 Mesh generation with ANSYS Mesh

Within the Fluid Flow (Fluent) component in WB, double-click on *Mesh* to start the mesh generation process.
Once the start of ANSYS Mesh is completed, the result will be as in figure 24. Clicking on *Mesh*, in the Outline tree on the left, the Details of Mesh will be available, in particular *CFD* as *Physics Preference* and *Fluent* as *Solver Preference* without further intervention from the user.

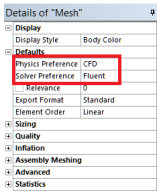

Figure 24: Physics and Solver Preferences in the start screen of ANSYS Mesh.

The first thing to do is to name all boundary surfaces - *patches* - in order to facilitate the setting of boundary conditions and the post-processing. As illustrated in figure 25, in the Outline Tree right click on *Model* $\rightarrow$ *Insert* $\rightarrow$ *Named Selection*.

[3]More precisely, should be 0.399624765478424.

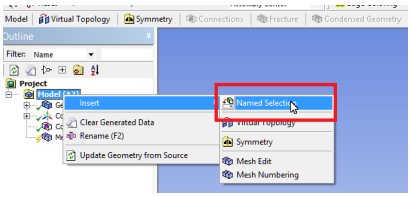

Figure 25: Insert Named Selection in ANSYS Mesh.

Afterwards:

- Rename it *Inlet*.
- Select the fluid, Inlet face (that with higher Z-value).
- Press *Apply*, as depicted in figure 26.

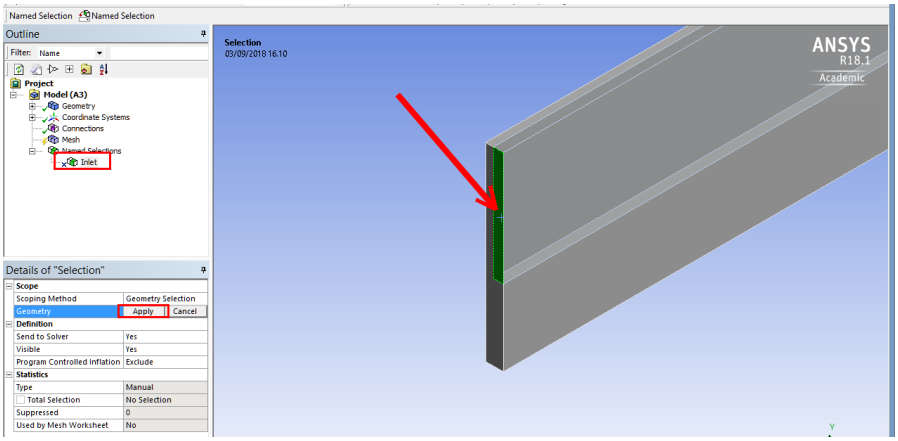

Figure 26: Set Inlet face in ANSYS Mesh.

Then proceed in the same way for all the other faces, which, with their obvious meaning, are: *Outlet, TopFin, TopFluid, Bottom, FrontSolid, BackSolid, SymmFluid, SymmSolidLeft, SymmSolidRight*.

At the end of this process, all named selections should be visible in the *Outline Tree*, as in figure 27.

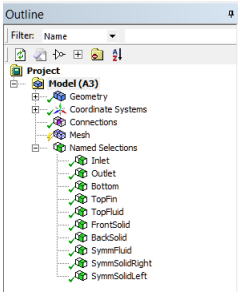

Figure 27: Named selections for the model.

Before proceeding with the mesh generation, it is necessary, in this case, to select an appropriate *Method* that will guarantee a high quality mesh, which in this case should be purely hexaedral-Cartesian, given the simple geometry of our problem. In fact, it is easy to see that, if we use the *default* method, the resulting mesh, no matter how we set the sizing parameters like *Max Face Size*, *Max Tet Size*, etc., will be either excessively fine along the streamwise direction, or too coarse in the cross-plane.
To do so, in the Outline Tree:

- Right click on *Mesh* $\rightarrow$ *Insert* $\rightarrow$ *Method*.
- In the Details of Method select *Geometry Selection* as *Scoping Method* and the two bodies, *Silicon* and *Water* as *Geometry*.
- For the Method select *Sweep*.
- For Source/Target select *Manual Source and Target*.
- For Source select *Inlet* and *FrontSolid*, for Target *Outlet* and *BackSolid*.
- For Free Face Mesh Type select *All Quad*, for Type leave *Number of Divisions* and set *Sweep Num Divs* as *100*.

All these choices are visualized in figure 28

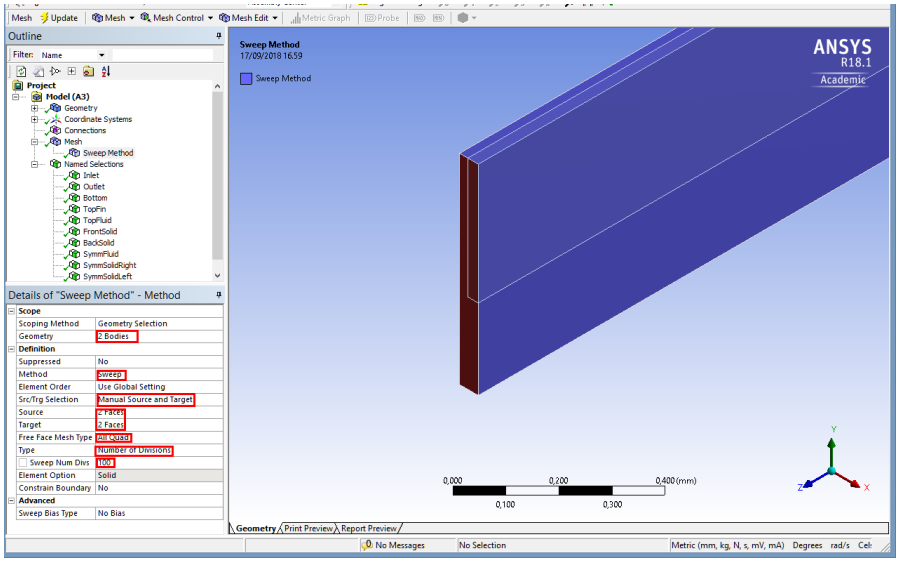

Figure 28: Parameters for the Meshing Method.

Once the Meshing Method has been selected, we should now set some *sizing*. The *sizing* should be selected taking into account the following points:

- This case is called a *conjugate heat transfer* problem, since the heat transfer occurs *simultaneously* in the solid and fluid region. Therefore, at the solid-fluid interface the heat flux will be the same and this, in turn, suggests to avoid large mesh size variation across the interface.

- The flow within the channel is a *developing flow*, therefore with axial gradients larger at the *entrance*.

- The discretization requirements for the solid are different than those for the fluid, since for the former there are no *boundary layers* and therefore the mesh size can be uniform.

In the Outline Tree right click on *Mesh* → *Insert* → *Sizing*.
In the *Details of "Sizing"*:

- in the *Scoping Method* leave *Geometry Selection*;

- for the *Geometry* select all 7 longitudinal (the longest) edges (use the *Edge* icon ( ));

- for *Type* select *Number of Divisions* and set the *Number of Divisions* to 100;

- for *Behavior* select *Hard*;

- for *Bias type* select – - · · · (see Figure 29);

- for *Bias factor* set 20.

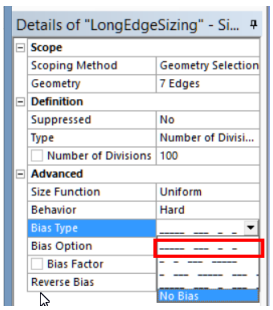

Figure 29: Bias type selection.

The edge sizing will be visualized in the graphic window as in figure 30 (it is suggested to use the wireframe visualization, in order to detect edges which require a *reverse bias*, available at the bottom of the *Details view*).

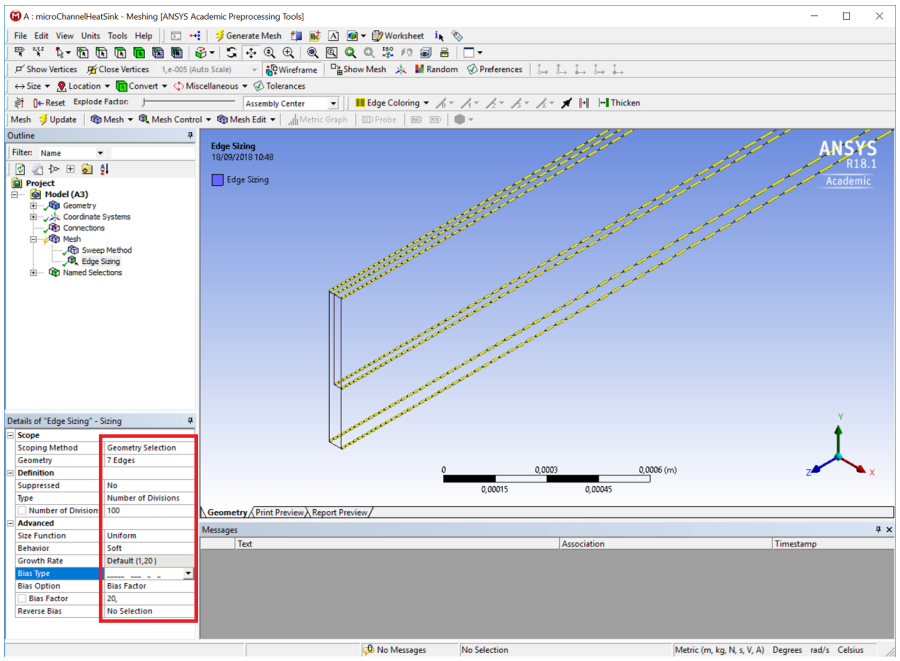

Figure 30: Long edges sizing.

For convenience, we rename the *Edge Sizing*, in the Outline Tree, as *LongEdgeSizing*. We should repeat the same sizing procedure for the other edges:

- set *Number of Divisions* to 8 and uniform sizing for the *fin thickness* (4 edges, since also the bottom horizontal splitted edges should be considered) and rename it as *FinThick-Sizing*;

- set *Number of Divisions* to 16 and *one-side bias* (smaller size towards the fin wall) with of *Bias factor* of 20 for the width of the fluid channel (6 edges, including the horizontal splitted edges) and rename it *FluidWidthSizing*;

- set *Number of Divisions* to 20 and *uniform sizing* for the vertical height of the base (4 edges, including the vertical splitted edges) and rename it *BaseHeightSizing*;

- set *Number of Divisions* to 50 and *two-side bias* (smaller size towards the boundaries) with *Bias factor*of 20 for the height of the fluid channel (6 edges, including the vertical splitted edges) and rename it *FluidHeightSizing*.

Once all the sizings have been inserted, we can generate the mesh and *send* it to the successive cell, e.g. *Fluent*: to do so, click on on the *Update* icon (Update).

To view the mesh, just click on *Mesh* in the Outline Tree: the results should look like in figure 31. The mesh, as expected, is purely *Cartesian*.

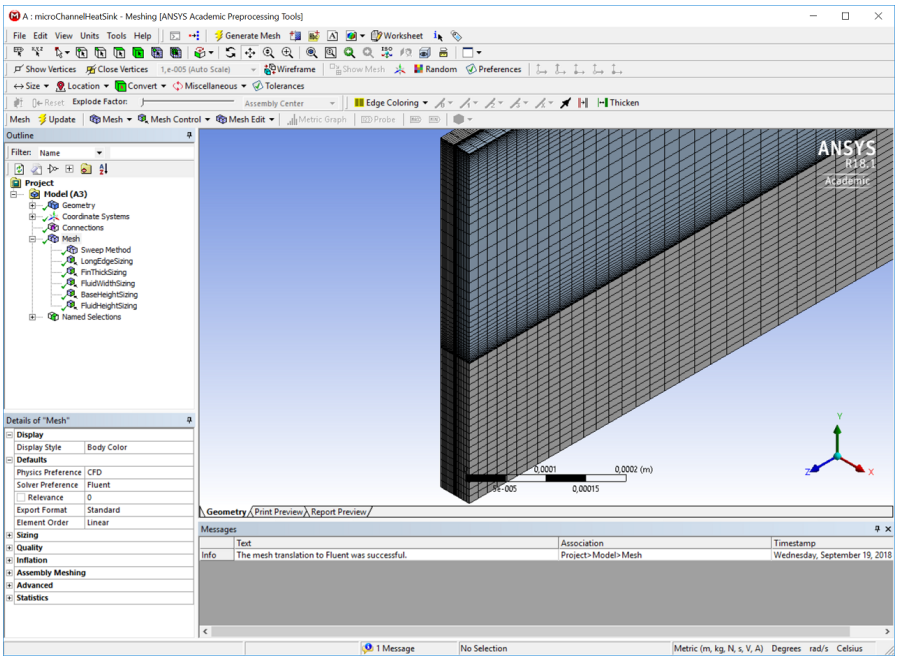

Figure 31: Overall view of the mesh.

Now we have to save our project, close the Mesh generator and start ANSYS Fluent.

4.3 Problem setup and solution with ANSYS Fluent

Within the Fluid Flow (Fluent) component in WB, double-click on *Setup* to start ANSYS Fluent [4]. When ANSYS Fluent is started, the Fluent Launcher is displayed: it enables to view and/or set certain ANSYS Fluent start-up options, as depicted in figure 32

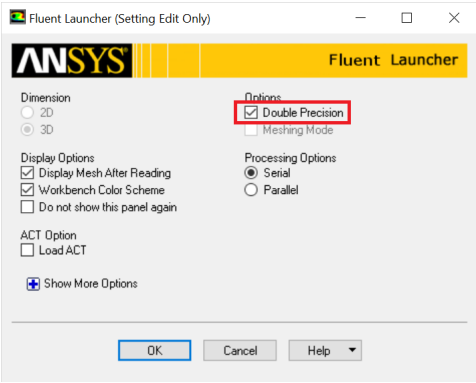

Figure 32: Fluent launcher.

[4]Please note that when Fluent is launched from the *Setup* cell, it loads only the mesh and settings that served as the starting point for the analysis and are associated with the *Setup cell*. In order to load the *current case file* (files that contain the mesh, boundary and cell zone conditions, and solution parameters for the problem) and *data files* (files that contain the values of flow field quantities and the convergence history) or the *initial data file*, it is necessary to launch Fluent from the *Solution* cell.

Ensure that the proper options are enabled:

- Verify that *Serial* from the Processing Options list is enabled.
- Select *Double Precision* under Options.
- Ensure that the *Display Mesh After Reading* and *Workbench Color Scheme* options are enabled.

Note that the *Dimension* setting is already filled in and cannot be changed, since ANSYS Fluent automatically sets it based on the mesh or geometry for the current problem.
Click *OK* to launch ANSYS Fluent. The Fluent application will start, and it will appear as in figure 33

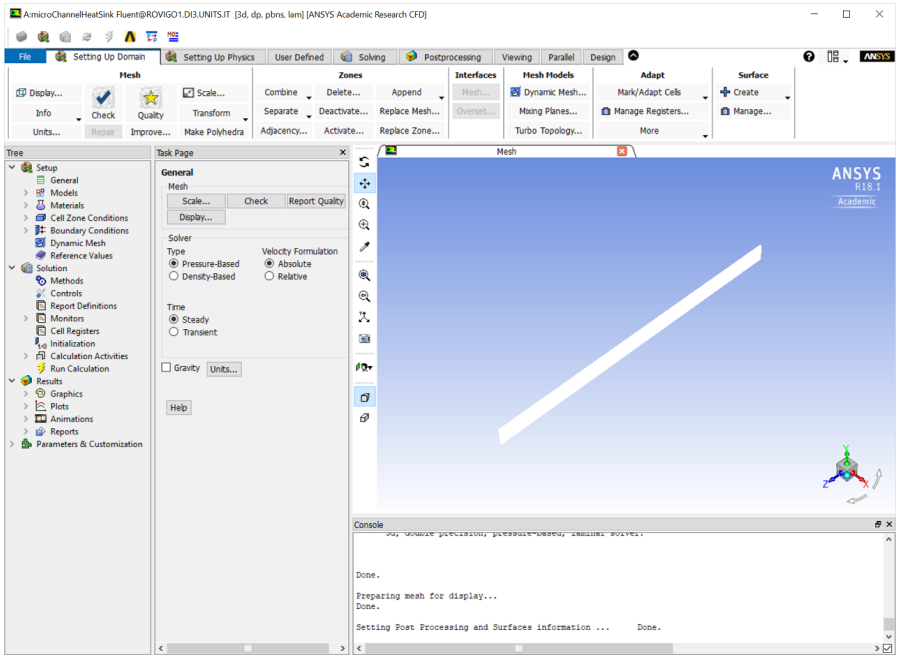

Figure 33: ANSYS Fluent in WB.

Setting up the domain

Ensure that the *Setting Up Domain* ribbon tab is active. It is now convenient to change the units for length within ANSYS Fluent from meters (the default) to millimeters:

1. Select *Units* under the *Mesh* group.
2. The *Set Units* dialog box will be displayed, as in figure 34.
3. Select *length* in the Quantities list.
4. Select *mm* in the Units list.
5. Close the dialog box.

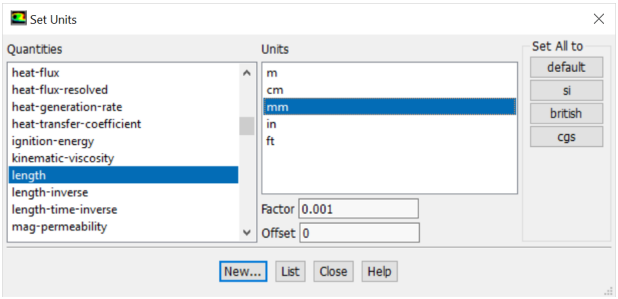

Figure 34: Fluent *Set Units* dialog box.

The second step is to check the mesh: under the *Mesh* group, click on the *Check* icon ( ). The mesh check will list in the Console window, among several other mesh features, the minimum and maximum x, y and z values of the mesh in the default SI units of meters. It is important to verify that the minimum volume is not negative, or otherwise Fluent will not be able to begin the calculation.

The next, suggested step, is to check the quality of the mesh: under the same *Mesh* group, click on the *Quality* icon ( ). The quality check results will also be reported in the Console window. The quality of the mesh is of paramount importance for the accuracy and stability of the calculation. Therefore, checking the quality of the mesh is an important step in order to guarantee a successful and accurate simulation. Values for *orthogonality* can vary between 0 and 1, with lower values indicating poor quality of the cells. In this case, if the mesh generation process was correct, the minimum orthogonal quality should be identically equal to 1 since all cells are pure orthogonal hexaedra, e.g. rectangular cuboid.
The high aspect ratio of some cells in this case - up to 1.2×10^3 - does not pose particular problems of *stability* or *numerical diffusion*, since these very elongated cells are aligned with the flow.

Setting up physics

Select the *Setting Up Physics* ribbon tab and then:

- In the *Solver* group leave the default selection of *Steady*, *Pressure-Based* solver with *Absolute Velocity Formulation*, as shown in figure 35.

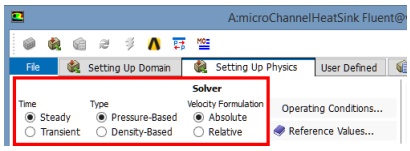

Figure 35: Solver options selection.

- In the *Models* group, select *Energy* in order to take into account the heat transfer, as illustrated in figure 36.

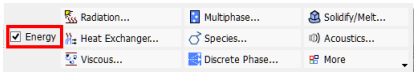

Figure 36: Activation of Energy option.

Then click *Viscous* and in the *Viscous Model* dialog box select *Laminar*[5] as depicted in figure 37.

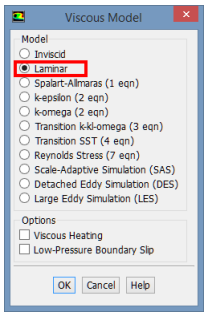

Figure 37: Selection of the *Laminar* Viscous Model.

Click *OK* to accept the model and close the *Viscous Model* dialog box.

- We have now to set up the materials of the heat sink. Since we model the case 1 of [1], we have to use the same values of the thermophysical properties the authors used, in particular:

  1. *Water*: $\rho = 997$ kg/m^3, $c_p = 4.179$ kJ/kg K, $k_w = 0.613$ W/m K, $\mu = 8.55 \times 10^{-4}$ kg/m s.
  2. *Silicon*[6] : $\rho = 2719$ kg/m^3, $c_p = 871$ J/kg K, $k_s = 148$ W/m K.

  In the *Materials* group of the *Setting Up Physics* ribbon tab, click on *Create/Edit ...* icon( ). In the *Create/Edit Materials* dialog box that will appear, type *water* for *Name* and in the *Properties* group enter the values of density, specific heat, thermal conductivity and dynamic viscosity just indicated. Click *Change/Create* and the result will be as in figure 38

[5] In this problem, given the very small dimensions of the channel, the *Reynolds* number is rather low (of the order of 300, based on the *Hydraulic Diameter*), and therefore the flow regime is steady laminar.

[6] Although for steady state analysis the only property required for solids is the *thermal conductivity*, for completeness we also insert the values of *density* and *specific heat* for silicon, taken from [2].

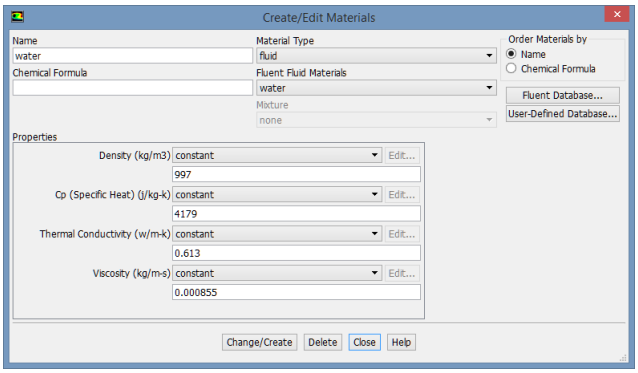

Figure 38: Inserting water properties in the *Create/Edit Materials* dialog box.

A *Question* dialog box will open, asking if you want to overwrite air. Click *Yes*, in order to avoid any possible confusion later, e.g. only water will be used as a fluid (note that this will affect only the *local* copy of material properties).

We have now to do the same for the solid material, *silicon*, inserting the values of density, specific heat and thermal conductivity given before. Also in this case, at the question, we will accept to overwrite aluminum, in order to leave only silicon as solid material.

- In the *Zones* group of the *Setting Up Physics* ribbon tab, click on *Cell Zones*. This will open the *Cell Zone Conditions* dialog box[7]. In the dialog box, two zones should be visible:

  1. *part-silicon*.
  2. *part-water*.

  Select *part-silicon* - in the *type* drop-down list it should be set to *solid* - and click *Edit* to open the *Solid* dialog box. Check that, on the *Material Name* drop-down list, *silicon* is selected. Click *OK* to close the *Solid* dialog box.
  Select *part-water* - in the *type* drop-down list it should be set to *fluid* - and click *Edit* to open the *Fluid* dialog box. Check that, on the *Material Name* drop-down list, *water* is selected. Click *OK* to close the *Fluid* dialog box.

- It is now necessary to setup the boundary conditions for the problem, using those indicated in [1]. In the *Zones* group of the *Setting Up Physics* ribbon tab, click on *Boundaries*. This will open the *Boundary Conditions* task page[8].
  It is convenient to display the boundary zones grouped by *Zone Type*: this can be

[7] Please note that the same dialog box can be opened by double-clicking the *Cell Zone Conditions* under the *Setup* tree branch on the left.

[8] Please note that the same task page can be opened by double-clicking the *Boundary Conditions* under the *Setup* tree branch on the left.

achieved by clicking the *Toggle Tree View* button in the upper right corner of the *Boundary Conditions* task page, and under the *Group By* select *Zone Type* from the drop-down menu, as shown in figure 39.

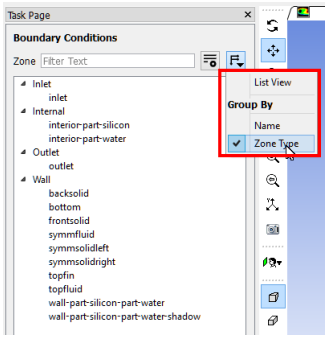

Figure 39: Boundary conditions grouped by *Zone Type*.

(a) Set the boundary conditions at the (fluid) *inlet*.
From the *Zone* selection list, select *inlet*, for the *Type* select *pressure-inlet*[9], as shown in figure 40 and click *Edit*.

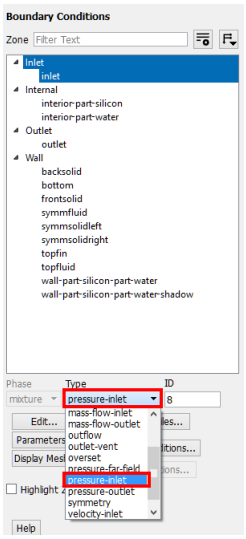

Figure 40: Inlet boundary condition as *pressure-inlet*.

As illustrated in figure 41, in the *Pressure Inlet* dialog box, set the *Gauge Total Pressure* equal to 103420 *Pascal*, and leave all other parameters for the *Momentum* tab as the default.

[9]In this case we use *pressure-inlet* and *pressure-outlet* as boundary conditions for the *inlet* and *outlet*, respectively, since in [1] the pressure drop ΔP = 103.42 kPa is given, rather than the inlet velocity or mass flow rate.

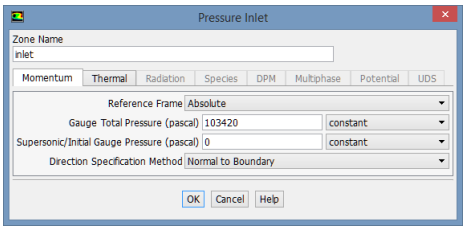

Figure 41: Pressure value at the *pressure-inlet*.

Click the *Thermal* tab, and set the *Total Temperature* equal to $300\ K$[10], as depicted in figure 42.

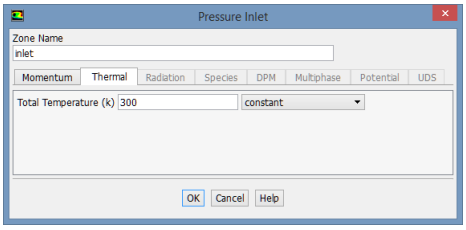

Figure 42: Temperature value at the *pressure-inlet*.

Click *OK* to close the *Pressure Inlet* dialog box.

(b) For the (fluid) *outlet*, select *outlet* from the *Zone* selection list, for the *Type* select *pressure-outlet* and click *Edit*. As illustrated in figure 43, in the *Pressure Outlet* dialog box, leave the *Gauge Pressure* equal to 0 *Pascal*, and leave all other parameters for the *Momentum* tab as the default.

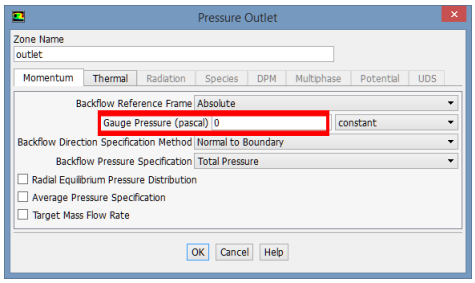

Figure 43: Pressure value at the *pressure-outlet*.

Click the *Thermal* tab, and set the *Backflow Total Temperature* equal to $300\ K$, as depicted in figure 44.

[10]Since in this problem the physical properties of the materials are assumed independent of temperature, and in addition all the other thermal boundary conditions are of Neumann type, e.g. imposed value of heat flux, the value of the *inlet temperature* has no relevance, therefore any convenient value can be given.

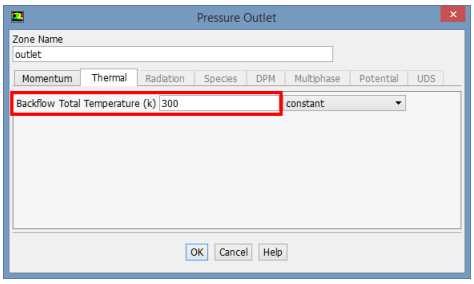

Figure 44: Backflow Temperature value at the *pressure-outlet*.

Click *OK* to close the *Pressure Outlet* dialog box.

(c) We have now to set all the following zones as *symmetry*: *symmfluid, symmsolidleft, symmsolidright*. Select each of them from the *Zone* selection list, for the *Type* select *symmetry* and click *Edit*. In the *Symmetry* dialog box just click *OK*.

(d) We have to set the *walls*:

(1) Select first *bottom* from the *Zone* selection list, for the *Type* select *wall* and click *Edit*. In the *Wall* dialog box, since it is a solid wall, only the *Thermal* tab will be active. In the *Thermal Conditions* list select *Heat Flux*, and set its value, as in [1], to 1818000 (1.81×10^6) W/m^2, as shown in figure 45.

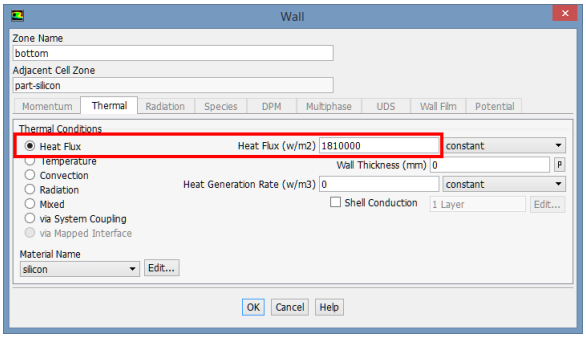

Figure 45: Setting heat flux value at the bottom of the microchannel.

Click *OK* to close the *Wall* dialog box for the *bottom* zone.

(2) Select now *topfluid* and verify that, in the *Momentum* tab, it is set as a *No Slip, Stationary Wall* and that in the *Thermal* tab it is set as an insulated wall, e.g. *Heat Flux* equal to zero. Click *OK* to close the *Wall* dialog box for the *topfluid* zone.

(3) For the *wall-part-silicon-part-water-shadow* verify that, in the *Momentum* tab, it is set as a *No Slip, Stationary Wall* while in the *Thermal* tab the *Thermal Conditions* should be set to *Coupled*, as depicted in figure 46.

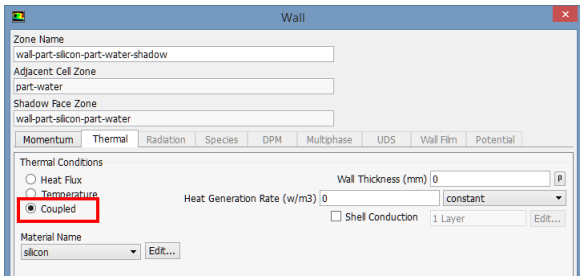

Figure 46: Coupled thermal boundary condition at the solid-fluid interface.

Click *OK* to close the *Wall* dialog box for the *wall-part-silicon-part-water-shadow* zone.

(4) In a similar way, verify that also for *wall-part-silicon-part-water* (the *Momentum* tab is inactive) the *Thermal Conditions* are set to *Coupled* in the *Thermal* tab.

(5) For all other walls - *topfin, frontsolid and backsolid* - check that in the *Thermal* tab the *Heat Flux* is set to zero.

Solution

Select the *Solving* ribbon tab and then:

- Click on the *Solution Methods* icon ( ).
- This will open the *Solution Methods* task page: leave all the choices as their default values, like in figure 47 (for this case, with a perfect orthogonal, Cartesian mesh aligned with the main flow direction, also the simpler - and more economical - *Green-Gauss Cell Based* gradient reconstruction method could have been used without adversely affecting the accuracy of the calculations).

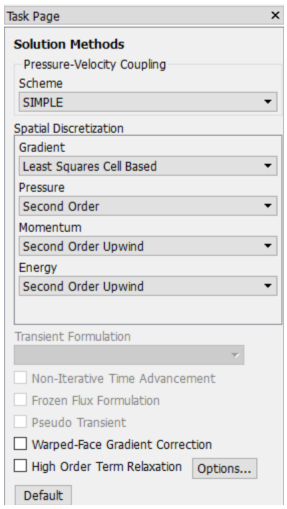

Figure 47: *Solution Methods* selection.

- In the *Reports* group, click the *Residuals* icon (Residuals...). This will open the *Residual Monitors* dialog box and on this:

  (a) Check that, in the *Options* group, *Plot* is enabled.

  (b) Change the values for the Convergence Absolute Criteria from 0.001 to 1×10^{-6} for the continuity and velocity components, and from 1×10^{-6} to 1×10^{-8} for the energy equation, as illustrated in figure 48.

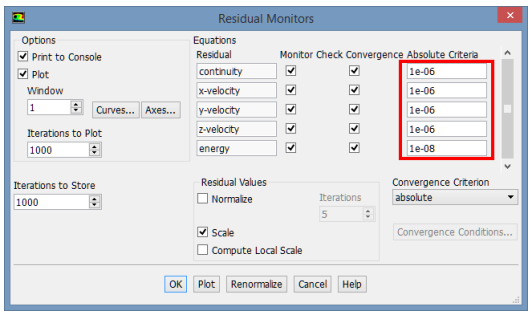

Figure 48: Convergence criteria in the *Residual Monitors* dialog box.

Click *OK* to close the *Residual Monitors* dialog box.

It is a recommended practice to monitor physical quantities relevant to the problem at hand, in addition to equation residuals, when judging convergence. These physical quantities can be, for example, the *force* or *drag* for an *external aerodynamics* problem, or the *pressure loss* for an *internal flow* problem et. In this case, as we will see later, it is convenient to monitor and plot the *highest temperature* in the heat sink.
For this purpose, let's create a surface report definition at the surface of the heat sink: in the *Reports* group, click *Definitions* → *New* → *Surface Report* → *Facet Maximum...*, see figure 49.

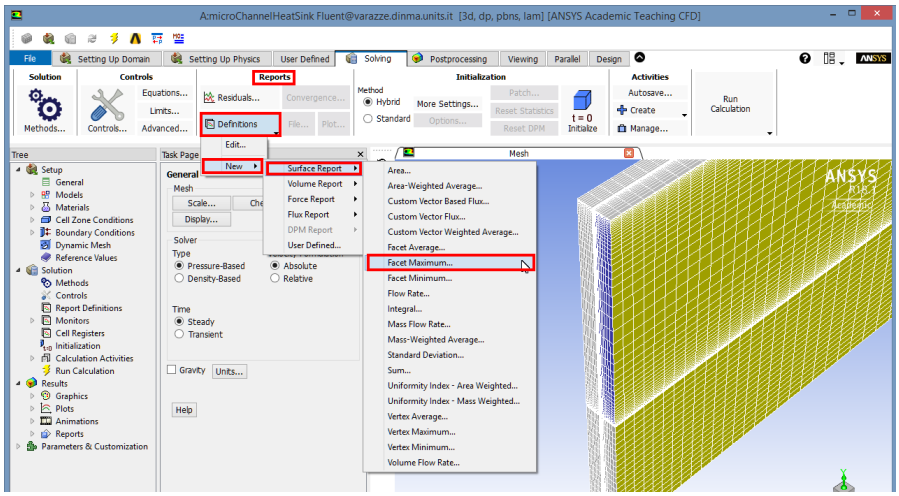

Figure 49: New Surface Report Definition.

In the *Surface Report Definition* dialog box, illustrated in figure 50:

(a) For the *Name* insert *max_temp_heat_sink*.

(b) Under the *Create* group, enable *Report Plot* and *Print to Console*.

(c) In the *Field Variable* select *Temperature, Static Temperature*. In the *Surfaces* group, toggle the tree view by *Surface Type* and select all the surfaces of *Wall* type.

(d) Click *OK* to save the surface report definition and close the *Surface Report Definition* dialog box.

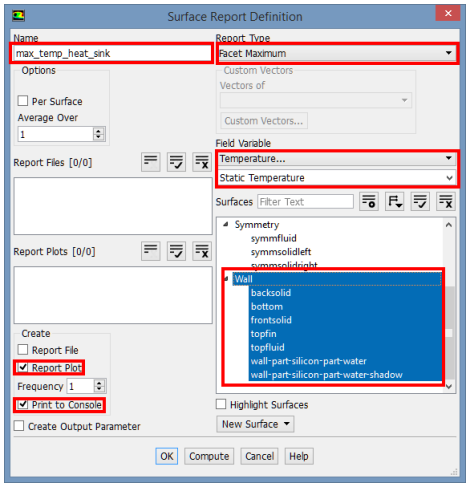

Figure 50: Surface Report Definition dialog box.

In the left tree, double-click *max_temp_heat_sink-rplot* under *Solution* → *Monitors* → *Report Plots* and, in the report plot settings in the *Editor Report Plot* dialog box shown in figure 51, insert *Max Temp. Heat Sink* in the *Plot Title* field and *Temperature* as *Y-Axis Label*.

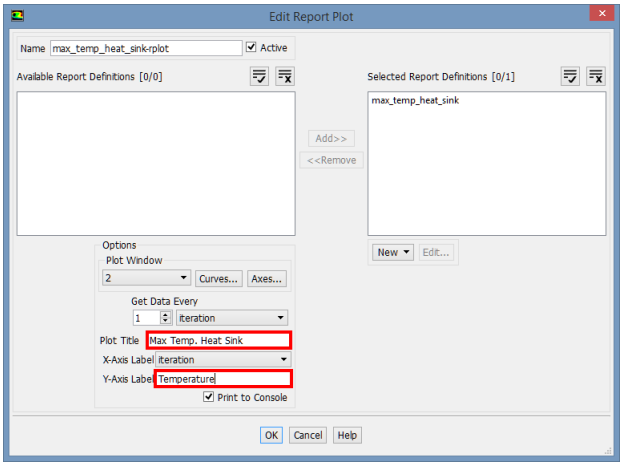

Figure 51: *Edit Report Plot* dialog box.

- Click *OK* to save and close the *Edit Report Plot* definition dialog box.
- In the *Initialization* group, verify that the *Method* selected is *Hybrid*, like indicated in figure 52.

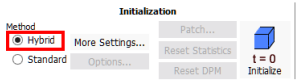

Figure 52: *Hybrid* initialization of the flow field.

Click *Initialize* (t = 0 Initialize).
It is now convenient *to patch* the entire domain - fluid *and* solid - with the *inlet temperature* value in order to speed up convergence. For this purpose, in the *Initialization* group, click *Patch...*: this will open up the *Patch* dialog box. In this, as shown in figure 53:

(a) Select *Temperature* in the variables list on the left.

(b) Insert 300 in the *Value (K)* field.

(c) Select both *part-silicon* and *part-water* as *Zones to Patch*.

(d) Click *Patch* and then *Close*.

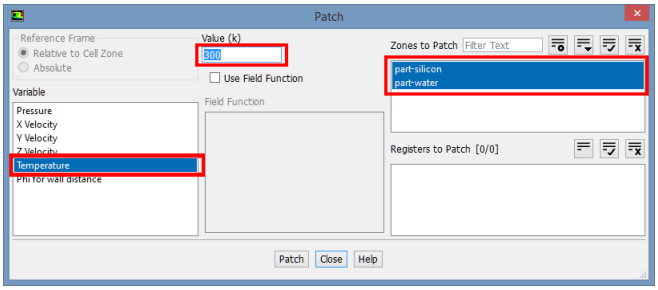

Figure 53: *Patch* initialization of the temperature field.

- In the *Run Calculation* group of the *Solving* Tab, enter 500 for *No. of Iterations*, as shown in figure 54.

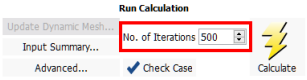

Figure 54: Set the maximum number of iterations for the calculation.

In the same group click the *Check Case* icon (Check Case). The response should be like in figure 55.

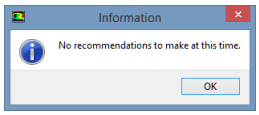

Figure 55: Result of the *Check Case*.

- In the *Run Calculation* group, click the *Calculate* icon ( ). The residuals history, during the calculation, will be plotted in the *Scaled Residuals* window, figure 56, while the highest temperature in the heat sink will be plotted in the *Max Temp. Heat Sink* window, shown in figure 57.

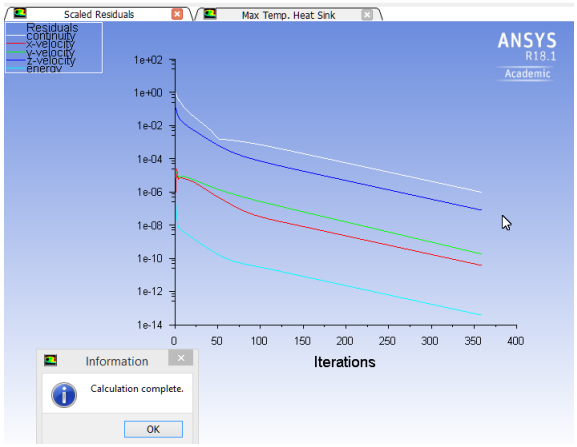

Figure 56: Residuals for the converged solution.

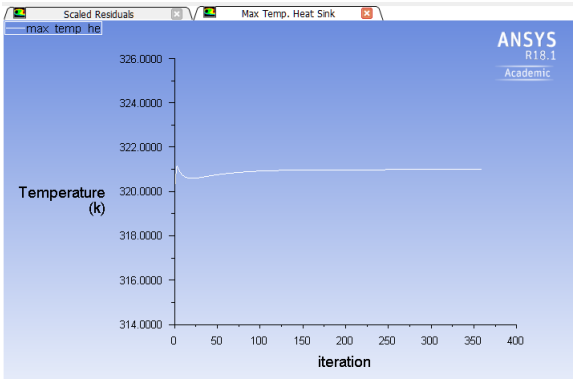

Figure 57: Maximum temperature of the heat sink during the solution.

The calculation will terminate when either the residuals reach their specified minimum values, or the maximum number of iterations, 500 in this case, will be reached. In this case the solution will converge in about 360 iterations [11]. For the maximum temperature

[11] Since the residual values vary slightly by computing platform (hardware, OS), the plot that appears on your screen may not be exactly the same as the one in figure 56.

in the heat sink, however, it is easy to verify - from both the plot and the print-out in the *Console* window - that it converges in about 90 iterations, reaching a value $T_{Max} = 321$ [K].

4.4 Postprocessing of the results with CFD-Post

Although it is possible to post-process the results directly within ANSYS Fluent, in this case we use the general purpose tool ANSYS CFD-Post.

Close Fluent and, within the Fluid Flow (Fluent) component in WB, double-click on *Results* to start *CFD-Post*. Once the start of ANSYS CFD-Post is completed, the result will be as in figure 58.

The microchannel geometry is already loaded and is displayed in *outline* mode. ANSYS Fluent case and data files are also automatically loaded into CFD-Post.

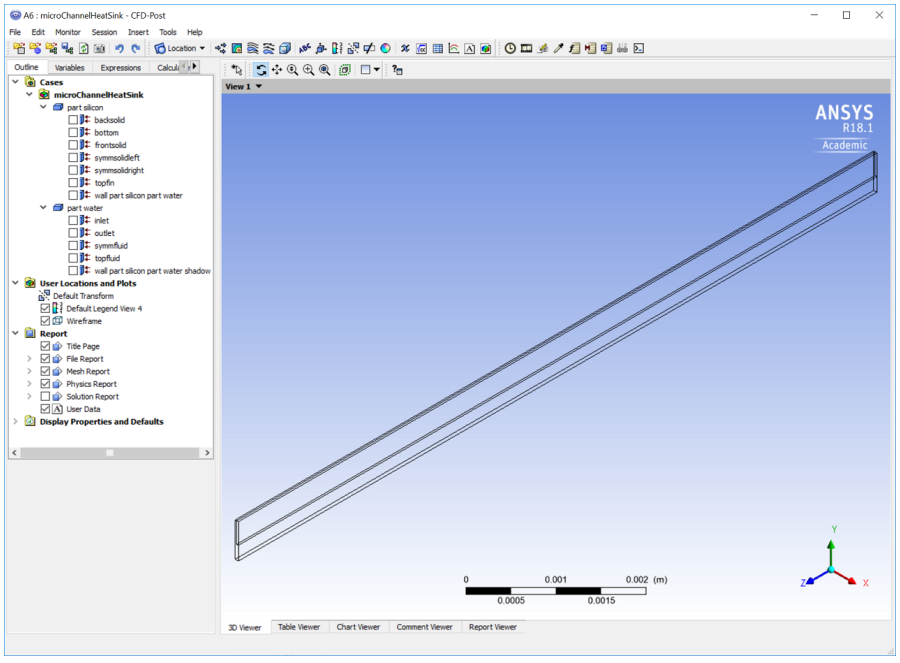

Figure 58: Start screen of ANSYS CFD-Post.

It would be interesting to visualize the temperature of the surface of the heat sink. To do this, click on the *Insert* menu item at the top of the CFD-Post window, and then *Contour*, as shown in figure 59. Alternatively, just click on the *contour* icon ( ).

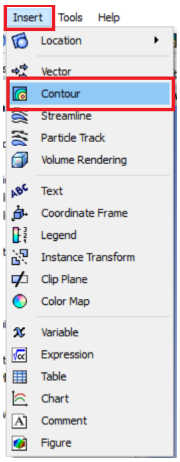

Figure 59: Insert contour in ANSYS CFD-Post.

This will display the *Insert Contour* dialog box and, as shown in figure 60, change its default name *Contour 1* to *Heat Sink surface temperature* and click *OK* to close the *Insert Contour* dialog box.

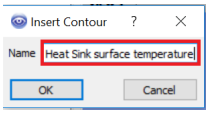

Figure 60: Insert contour name.

This will display the *Details of Heat Sink surface temperature* below the outline on the left. In this, within the *Geometry* tab:

(a) From the *Domains* drop-down menu, select *part silicon*.

(b) For the *Locations*, click on the *Location editor* icon (...).

(c) In the *Location Selector* dialog box that will appear, select all the solid surfaces, as indicated in figure 61.

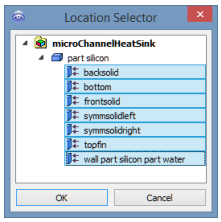

Figure 61: Select locations for contour plot.

(d) From the *Variable* drop-down list, select *Temperature* and as *Range* leave the *Global* default selection.

(e) In the *# of Contours* field, change the value from 11 (the default) to 21.

(f) Click *Apply*.

The result will be as in figure 62. It is clear from the figure that, as expected, the highest temperature is reached on the bottom of the substrate at the exit section.

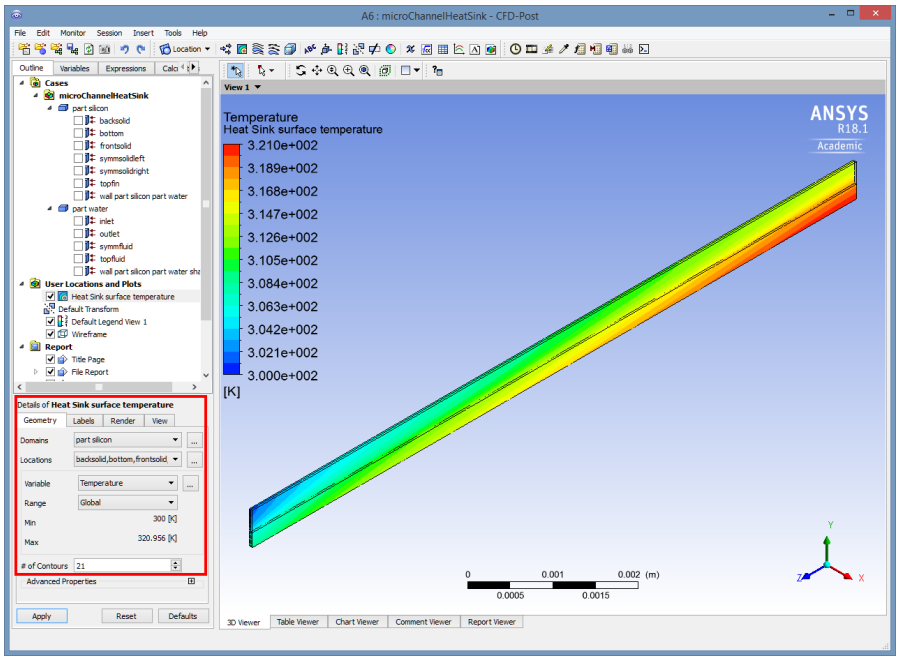

Figure 62: Temperature on the surface of the microchannel heat sink.

We want to confirm that the highest temperature on the surface of the heat sink is 321 [K], as already found during the calculations. To do so, click on the *Function Calculator* icon ( ). This will display the *Function Calculator* dialog box on the left:

(a) From the *Function* drop-down menu, select *maxVal*.

(b) For the *Location*, select *bottom*.

(c) Select *Temperature* as variable.

(d) Verify that the *Clear previous results on calculate* check box is selected.

(e) Click *Calculate*

The result is shown in figure 63.

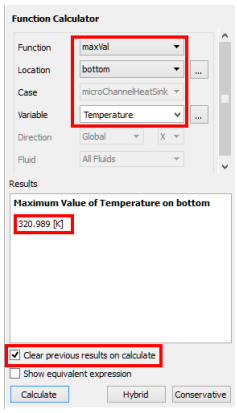

Figure 63: Max temperature for the heat sink.

The result confirm that the maximum temperature of the heat sink is 321 [K].
Since, as indicated in [1], the overall thermal resistance of the heat sink is defined as:

$$R = \frac{\Delta T_{Max}}{q'' \, A_s} \tag{18}$$

where $\Delta T_{Max} = (T_{w,o} - T_{f,i})$ is the maximum temperature rise in the heat sink, e.g. the temperature difference between the highest temperature in the heat sink at the outlet ($T_{w,o} = 321$ [K]) and the fluid inlet temperature ($T_{f,i} = 300$ [K]), $q'' = 1.81 \times 10^6$ W/m^2 is the heat flux at the bottom and $A_s = 1 \times 10^{-4}$ m^2 is the projected area of the heat sink.
It follows that, from the results of the calculation, the overall thermal resistance of the heat sink is $R = 0.116$ [K/W], which compares very well with the numerical result of Liu and Garimella [1], who found $R = 0.115$ [K/W], and also agrees reasonably well with the experimental result of Tuckerman and Pease [3], who found $R = 0.110$ [K/W].

Several other activities and variations can be done, like:

- Perform a grid-independence study.
- Check the other two cases considered in [1]:
  - for *Case 3*, for example, the numerical results, obtained with the same number of cells of *Case 1*, gives a value of thermal resistance of $R = 0.094$ [K/W], which again agrees well with both the numerical result reported in [1], $R = 0.093$ [K/W], and the experimental value of $R = 0.090$ [K/W] from [3].
- Verify the effects of temperature-dependency of the properties for both water and silicon.
- Verify the effects of radiation heat transfer.

This tutorial constitutes an example for conjugate heat transfer problems, which may be helpful for other problems of heat transfer in electronics and microelectronics.

References

[1] D. Liu and S. V. Garimella, Analysis and optimization of the thermal performance of microchannel heat sinks, *Int. J. Numerical Meth. Heat & Fluid Flow*, Vol.15, No. 1, pp. 7-26, 2005.

[2] F. P. Incropera, D. P. DeWitt, T. L. Bergman, A. S. Lavine, Fundamentals of Heat and Mass Transfer, John Wiley & Sons; 6th edition, 2006.

[3] D.B. Tuckerman, R.F.W Pease, High Performance Heat Sinking for VLSI, *IEEE Elect. Dev. Lett.*, vol. 2, 1981, pp 126-129.

[4] S. Zimmermann, I. Meijer, M. K. Tiwari, S. Paredes, B. Michel, D. Poulikakos, Aquasar: A hot water cooled data center with direct energy reuse, *Energy*, vol. 43. pp. 237-245, 2012.