

TECHNICAL REPORT:
THERMOALPHA – Heatblock - Input Manual

February, 2017

Jan Vermaak
Rev 1.0



Contents

	Page
1 The Input File	2
1.1 HEATBLOCK	3
1.2 HEATBLOCKK	3
1.3 HEATBLOCKC	4
1.4 HEATBLOCKRHO	4
1.5 HEATBLOCKT	4
1.6 HEATBLOCKL	4
1.7 HEATBLOCKR	5
1.8 HEATBLOCKESHAPE	5
1.9 HEATBLOCKESHAPE	5
1.10 Input file example	6
2 Scripting interface	7

List of Figures

Figure 1 Heat block nodalization.	2
Figure 2 Flow cycle of the simulation.	8
Figure 3 Interactive display.	9

1 The Input File

The initial simulation setup of a radial heat conduction model is performed using an input file. The input file specifies all the parameters needed to define a heatblock as shown in Figure 1 below. Physical regions are contained in **intervals** and temperatures are calculated at **nodes**. The simulation is one-dimensional (1D) and therefore does not account for azimuthal- or axial heat conduction.

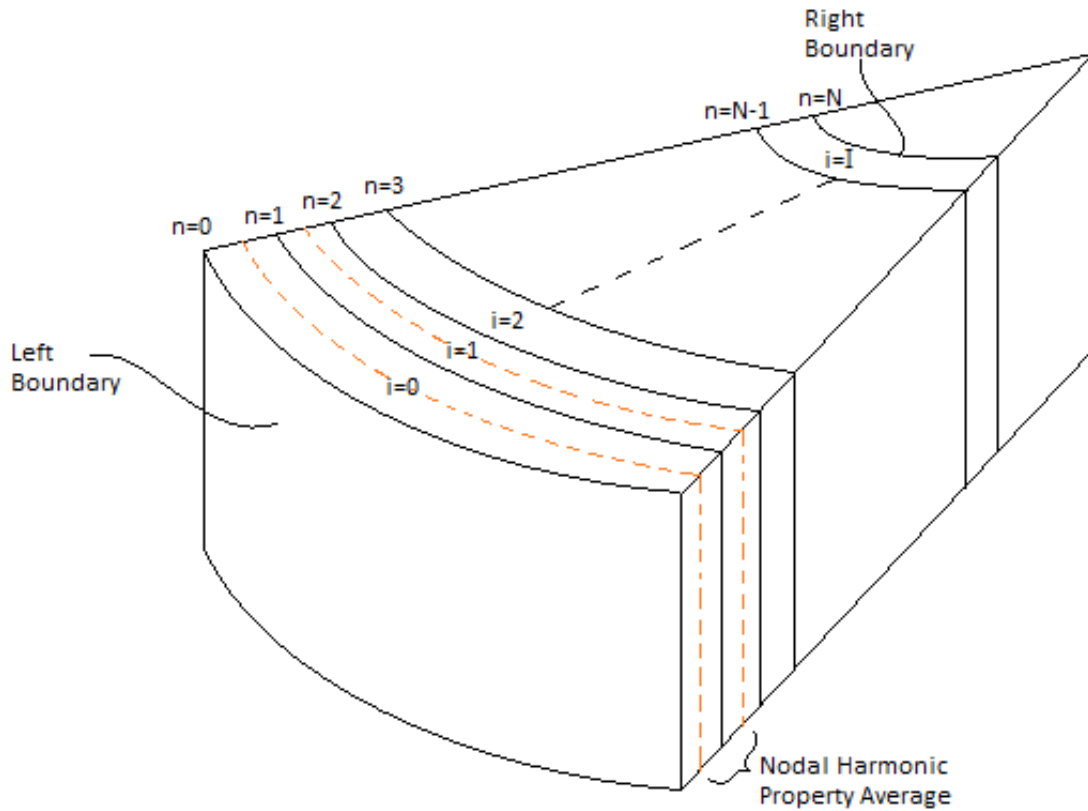


Figure 1: Heat block nodalization.

The *Heatblock* input cards are entered on the same file used for the input to *Thermoalpha*. The input cards include:

- **HEATBLOCK**, defines the mesh.
- **HEATBLOCKK**, defines interval thermal conductivity k .
- **HEATBLOCKC**, defines interval heat capacity C_p .
- **HEATBLOCKRHO**, defines interval density ρ .
- **HEATBLOCKT**, defines initial nodal temperature T_n .
- **HEATBLOCKL**, defines left boundary condition.
- **HEATBLOCKR**, defines right boundary condition.
- **HEATBLOCKSHAPE**, defines radial heat generation profile.
- **HEATBLOCKE**, defines block average heat generation rate.

1.1 HEATBLOCK

Defines the amount of nodes and radial mesh spacing.

- **Word 1**. [String] Heatblock name. Maximum 200 characters.
- **Word 2**. [Integer] Number of nodes in the spatial mesh N .
- **Word 3 to $(3 + N)$** . [Float] Nodal radius value. This value cannot be negative. The values of each nodal radius must be monotonically decreasing and will result in a crash if specified otherwise.

1.2 HEATBLOCKK

Defines interval thermal conductivity k .

- **Word 1**. [String] Heatblock name. Maximum 200 characters.
- **Word 2 to $(2 + I)$** . [Float] Interval thermal conductivity k .

1.3 HEATBLOCKC

Defines interval heat capacity C_p .

- **Word 1.** [String] Heatblock name. Maximum 200 characters.
- **Word 2 to $(2 + I)$.** [Float] Interval heat capacity C_p .

1.4 HEATBLOCKRHO

Defines interval density ρ .

- **Word 1.** [String] Heatblock name. Maximum 200 characters.
- **Word 2 to $(2 + I)$.** [Float] Interval density ρ .

1.5 HEATBLOCKT

Defines initial nodal temperature T_n .

- **Word 1.** [String] Heatblock name. Maximum 200 characters.
- **Word 2 to $(2 + N)$.** [Float] Initial nodal temperature T_n .

1.6 HEATBLOCKL

Defines the left boundary condition.

- **Word 1.** [String] Heatblock name. Maximum 200 characters.
- **Word 2.** [Integer] Boundary condition type. 1-convection boundary condition, 2-symmetry/insulated boundary condition, 3-heat flux. If this word is 1 then words 3 and 4 are convection coefficient h and bulk coolant temperature T_b . If this word is 2 then words 3 and 4 are ignored. If this word is 3 then word 3 is the specified heat flux and word 4 is ignored.
- **Word 3.** [Float] Required if word 2 is 1 or 3. If word 2 is 1 then this word is the convection coefficient h [$W.m^{-2}.K^{-1}$]. If word 2 is 3 then this word is the specified heat flux q'' [$W.m^{-2}$].
- **Word 4.** [Float] Required if word 2 is 1. This word is the bulk coolant temperature T_b [C] associated with the convection coefficient in word 3.

1.7 HEATBLOCKR

Defines the right boundary condition.

- **Word 1.** [String] Heatblock name. Maximum 200 characters.
- **Word 2.** [Integer] Boundary condition type. 1-convection boundary condition, 2-symmetry/insulated boundary condition, 3-heat flux. If this word is 1 then words 3 and 4 are convection coefficient h and bulk coolant temperature T_b . If this word is 2 then words 3 and 4 are ignored. If this word is 3 then word 3 is the specified heat flux and word 4 is ignored.
- **Word 3.** [Float] Required if word 2 is 1 or 3. If word 2 is 1 then this word is the convection coefficient h [$W.m^{-2}.K^{-1}$]. If word 2 is 3 then this word is the specified heat flux q'' [$W.m^{-2}$].
- **Word 4.** [Float] Required if word 2 is 1. This word is the bulk coolant temperature T_b [C] associated with the convection coefficient in word 3.

1.8 HEATBLOCKESHAPE

Defines radial interval heat generation profile.

- **Word 1.** [String] Heatblock name. Maximum 200 characters.
- **Word 2 to $(2 + I)$.** [Float] Interval relative heat generation factor. This value gets multiplied by the block average heat generation rate \dot{e}_{gen} to specify the heat generation rate in this interval. A factor of 0.0 indicates that no heat will be generated.

1.9 HEATBLOCKESHAPE

Defines block average heat generation rate.

- **Word 1.** [String] Heatblock name. Maximum 200 characters.
- **Word 2.** [Float] Block average heat generation rate \dot{e}_{gen} [$W.m^{-3}$].

1.10 Input file example

```
// Heatblock fuel rod input model
// Inner hole(OD)      = 3.175   (6.35   mm) = 0.0031750 m (normally filled with zirconium rod)
// Meat radius(OD)     = 17.4115 (34.823 mm) = 0.0174115 m
// Clad radius(OD)     = 17.9195 (35.839 mm) = 0.0179195 m
// Logic 9 meat intervals, 1 cladding interval
// ##### MESH #####
HEATBLOCK h101 11 0.0179195 0.0174115 0.015829667 0.014247833 0.012666 0.011084167
                _0.009502333 0.0079205 0.006338667 0.004756833 0.003175

// ##### THERMAL CONDUCTIVITY #####
HEATBLOCKK h101 17.360 18.000 18.000 18.000 18.000 18.000 18.000 18.000 18.000 18.000
// ##### HEAT CAPACITY #####
HEATBLOCKC h101 500.0 321.0 321.0 321.0 321.0 321.0 321.0 321.0 321.0 321.0
// ##### DENSITY #####
HEATBLOCKRHO h101 8000.0 7133.0 7133.0 7133.0 7133.0 7133.0 7133.0 7133.0 7133.0 7133.0

// ##### INITIAL TEMPERATURE #####
HEATBLOCKT h101 30.00 30.00 30.00 30.00 30.00 30.00 30.00 30.00 30.00 30.00
// ##### LEFT BOUNDARY CONDITION #####
HEATBLOCKL h101 1 2000.0 30.0
// ##### RIGHT BOUNDARY CONDITION #####
HEATBLOCKR h101 2

// ##### RADIAL HEAT GENERATION PROFILE #####
HEATBLOCKESHAPE h101 0.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0
// ##### AVERAGE HEAT RATE #####
HEATBLOCKE h101 1.0
```

2 Scripting interface

ChiTech executes a physics loop at a frequency of 60 Hz, corresponding to once every 16.66667 milliseconds. This physics routine uploads simulation values like the total program time [s], *chi_programTime*, to the scripting interface and executes a function called *main*. At program initialization the *main* is merely a blank function:

```
function main()  
end
```

The user has the option to run *ChiTech* with a specified input script. This script will execute after the entire simulation is initialized and therefore the user can either choose to run the code before the first call to *main* or within the *main* function itself. See Figure 2. In order to run the code before the first call to *main* the following lua script (running for 1000 time steps) can be called:

```
chiThermoflowProcessInput("CHI_RESOURCES/Worlds/Tut02Heatblock/Input.txt");  
chiThermoflowInitialize();  
  
for k=1,1000 do  
    chiThermoflowHBPerformTimeStep(0.0166667);  
end  
  
chiThermoflowPrintOutput("CHI_RESOURCES/Worlds/Tut02Heatblock/Output.txt");
```

For interactive use the time step can be issued withing the *main* function as shown below. One aspect that gets lost in this way is that there is no means to write the output file reliably. Also, some code needs to be added to receive feedback, i.e. calls to get a temperature value and store it within lua.

```
chiThermoflowProcessInput("CHI_RESOURCES/Worlds/Tut02Heatblock/Input.txt");  
chiThermoflowInitialize();  
  
function main()  
    chiThermoflowHBPerformTimeStep(0.0166667);  
end
```

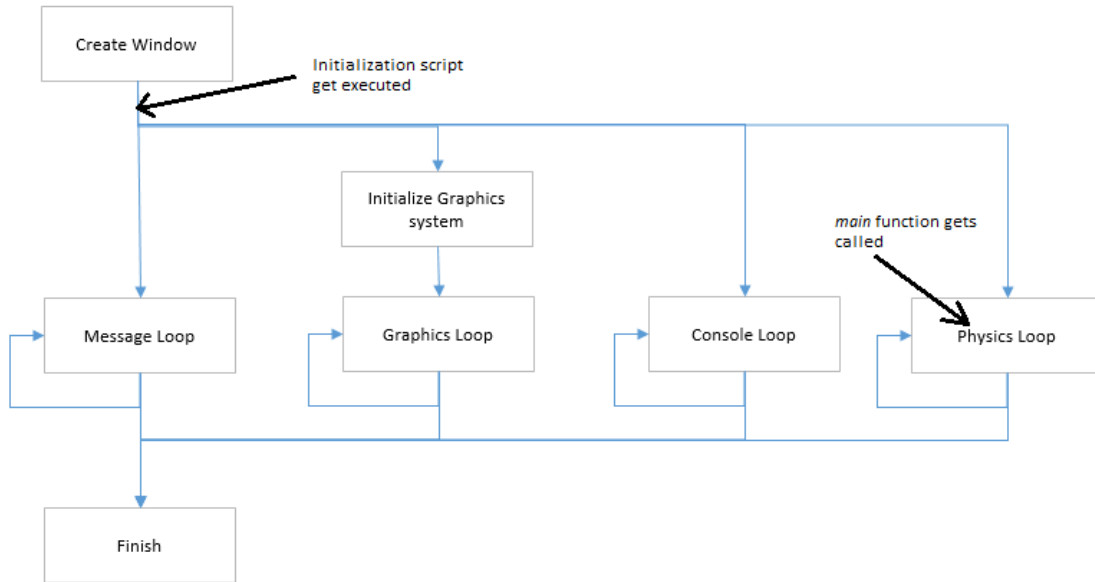



Figure 2: Flow cycle of the simulation.

The following is an example of a main function that prints a temperature to the interactive window and allows the heat input to be increased using the *A* and *Z* key presses:

```

chiThermoflowProcessInput("CHI_RESOURCES/Worlds/Tut02Heatblock/Input.txt");
chiThermoflowInitialize();

power=31.673e6;
AkeyDn=false;
ZkeyDn=false;
function main()
    chiThermoflowHBPerformTimeStep(0.0166667);
    cladT =chiThermoflowGetHBProperty("Node_Temperature","h101",0);
    centerT =chiThermoflowGetHBProperty("Node_Temperature","h101",10);

    if (WM_KEYDN.occured) then
        if (WM_KEYDN.iPar0 == 65) then AkeyDn = true; end
        if (WM_KEYDN.iPar0 == 90) then ZkeyDn = true; end
    end

    if (WM_KEYUP.occured or WM_MOUSELEAVE.occured) then
        if (WM_KEYUP.iPar0 == 65) then AkeyDn = false; end
        if (WM_KEYUP.iPar0 == 90) then ZkeyDn = false; end
    end

    if (AkeyDn) then power=power+1.0e6; end
    if (ZkeyDn) then power=power-1.0e6; end

    chiThermoflowSetHBProperty("HeatGeneration","h101",power);
    chiSetLabel("Tclad","Tclad_="..string.format("%5.3f", cladT),5,100);
    chiSetLabel("Tcent","Tcent_="..string.format("%5.3f", centerT),5,120);
    chiSetLabel("Heat","Heat_="..string.format("%5.3e", power),5,140);
end

```

The output of the above script will resemble the following:

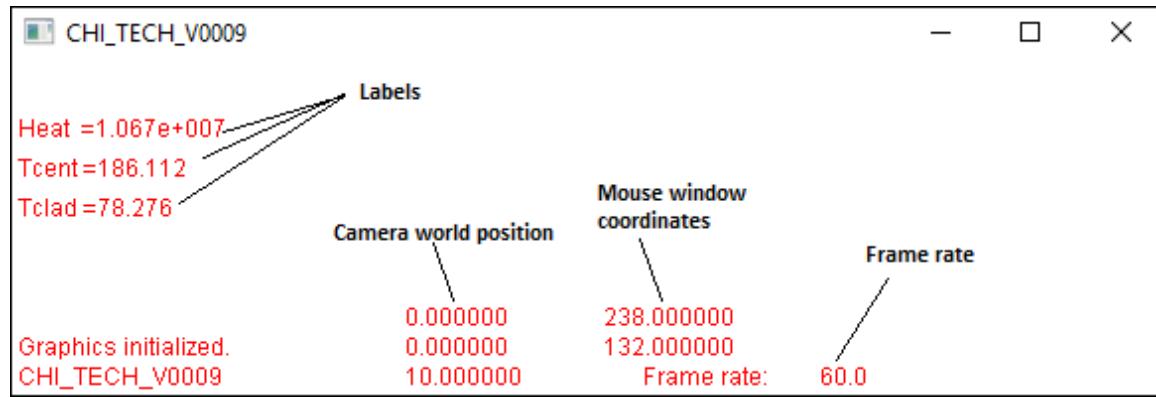


Figure 3: Interactive display.