1D Heat Flux Code breakdown

|  |  |  |
| --- | --- | --- |
| Version log | Change log | Date |
| 1 | First issue of document | 07/04/2020 |

# Introduction

Parameters and functions for each file explained in the current version of the code, code for each file presented at the bottom of each section. Where possible the labels assigned to parameters in the code are used.

## Applicable documents

# Code breakdown

Each file is separated and handled independently to understand the flow of each section, below is a holistic view of how the code operates.

Coolant\_Geometry.py

Runner.py

CHICA 1.3.py

Setup.py

CHICA 1.3.py

Non-Dimensional.py

Non-Dimensional.py

## CHICA 1.3.py

### CoolProp

Table 1‑1 details the parameters used in the code and their physical definitions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **len(Parameter)** | **units** | **comments** |
| section\_0 | 10 | m | X coordinate of the copper panel |
| section\_1 | 10 | m | Y coordinate of the copper panel |
| input\_power | 9 | W/m2 | Input power per meter squared |
| h | 1 | m | Thickness of copper |
| epsi | 1 | m | Surface roughness |
| input\_pressure | 1 | Pa | Currently assumed 80 bar |
| input\_temperature | 1 | K | Currently assumed 100 oC |
| input\_rho | 1 | kg/m3 | Calculated used CoolProp and input\_pressure and input\_temperature |
| q\_r | 10 | kg/s | Mass flow rate, varies from 5 -50 in steps of 5 |
| v\_r\_input | 10 | m/s | Velocity, varies from 5 -50 in steps of 5 |

Table 1‑1 Parameter Definitions

## Setup.py

### Parameter Definition

The class *setup* contains two functions, *initial\_setup* and *looper\_setup,* used to initialise inputs for the solver functions in Runner.py. These solvers being an initial guess of flow conditions considering constant pressure, *initial*, and an iterative solver for calculated pressure, *looper*. The outputs of the setup functions are detailed in Table 2‑1, where Figure 2‑1 presents the flow of the code.

Setup.py output:

htc\_0 T\_metal

Re P\_secc

Pr hf\_tot

Nu Alist

h\_f deltaz

Ag T\_ref

tg

dh

v\_secc

input\_power

non\_dimensional output:

nu

re

pr

h\_1

v\_s

coolant\_geometry output:

Ag

tg

dh

deltaz

Alist

input\_power

Setup.py input:

section\_0

input\_temperature

input\_pressure

y

input\_power

h

MassFlow

input\_rho

epsi

section\_1

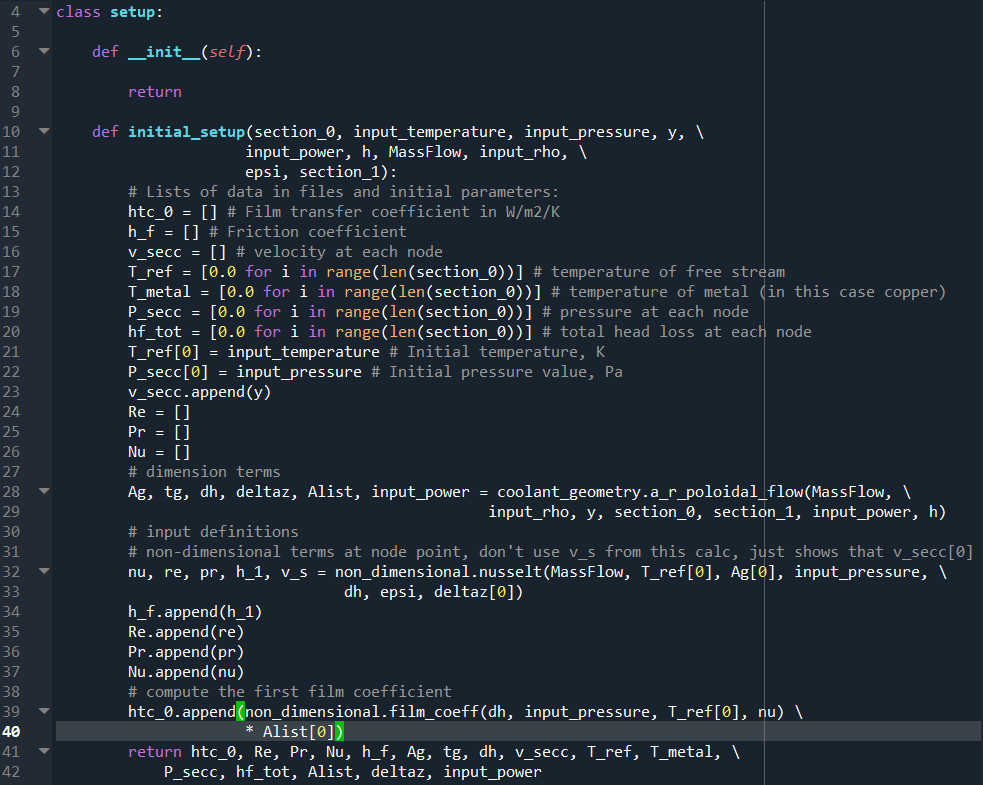
Figure ‑ Setup.py code flow

Setup.py

The outputs of *initial\_setup* that are derived from the *coolant\_geometry* and *non\_dimensional*  functions are described in 2.3 and 2.4 respectively. No calculations are performed in the body of this code outside those called from other classes.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Units** | **Type** | **Description** |
| htc\_0 | W/K | List of floats | Film coefficient |
| h\_f | m | List of floats | Head loss in pipe |
| v\_secc | m/s | List of floats | Velocity of bulk flow |
| T\_ref | K | List of floats | Temperature of coolant |
| T\_metal | K | List of floats | Temperature of copper |
| P\_secc | Pa | List of floats | Pressure of coolant |
| hf\_tot | m | List of floats | Total head loss of pipe, sum of h\_f |
| Re | - | List of floats | Reynolds number at each discretised point |
| Pr | - | List of floats | Prandtl number at each discretised point |
| Nu | - | List of floats | Nusselt number at each discretised point |

Table ‑ Setup.py Parameter Definition



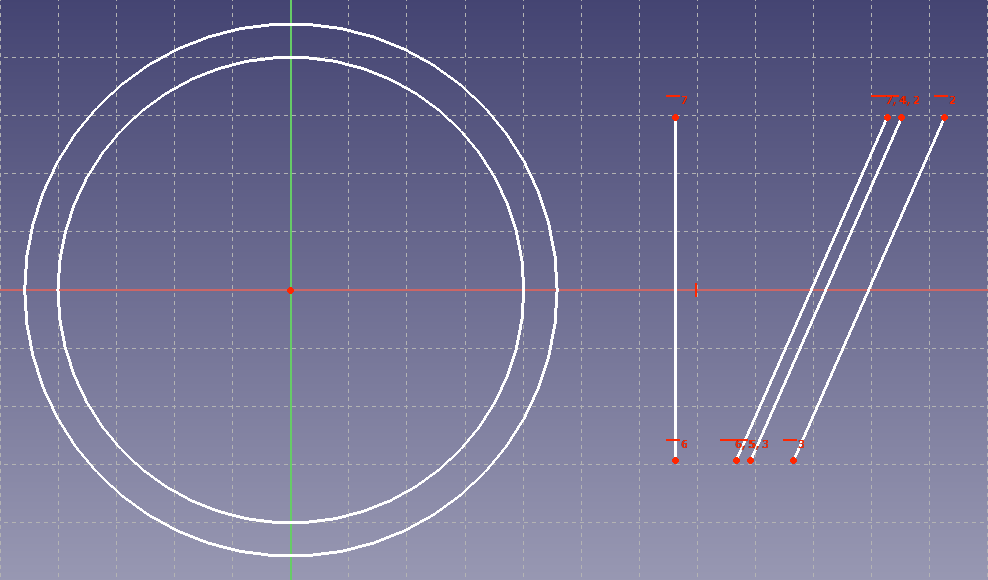
## Coolant\_Geometry.py

### Parameter definition

The class *coolant\_geometry* contains two functions, *a\_r\_poloidal* and *a\_r\_toroidal* for poloidal and toroidal flow cases. The outputs of these functions are detailed in Table 2‑2, where Figure 2‑2 presents the geometry in a cartesian coordinate system. Sections 2.3.2 to 2.3.7 explain the calculation for each parameter.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Units** | **Type** | **Description** |
| tg | m | Float | Thickness of the gap for the coolant, this is assumed constant for the entire channel. |
| Ag | m2 | List of floats | Area of the channel of the annulus at a given discretised point.  Flow out of plane with Ag. |
| deltaz | m | List of floats | Direct distance between two adjacent discretised points. |
| dh | m | Float | Hydraulic diameter at a given discretised points. |
| Alist | m2 | List of floats | Area of the internal wall between adjacent discretised points.  Flow in plane with Alist. |
| input\_power | W | List of floats | Updates the original input\_power list in [W/m2] to [W], accounting for area distribution as the flow progresses along the channel. |

Table ‑ Coolant\_Geometry.py Parameter definition



Y

X

X

Z

tg

h

Ag

deltaz

Centre of revolution

Figure ‑ Parameter definition

### tg

|  |  |  |
| --- | --- | --- |
|  |  | () |

= volumetric flow rate

= mass flow rate, q\_r

= velocity, v\_r\_input

= Area,

R = internal radius, section\_0[i]

As , , R and u are known, tg is then calculated via solution of quadratic equation.

### Ag

|  |  |  |
| --- | --- | --- |
|  |  | () |

R = internal radius, section\_0[i]

tg = coolant gap thickness, tg

### deltaz

|  |  |  |
| --- | --- | --- |
|  | deltaz = | () |

= section\_1[i+1], section\_1[i]

= section\_0[i+1], section\_0[i]

### dh

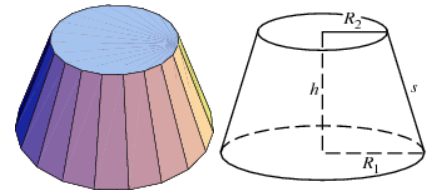
|  |  |  |
| --- | --- | --- |
|  | dh = 2 x tg | () |

For an annulus hydraulic diameter is double the gap thickness tg.

### Alist

Calculated using the equation for the surface area of a Frustrum, minus the area of the top and bottom circles.

|  |  |  |
| --- | --- | --- |
|  |  | () |



h = section\_1[i] – section\_1[i+1]

= section\_0[i], section\_0[i+1]

### input\_power

|  |  |  |
| --- | --- | --- |
|  | input\_power[i] [W] = input\_power[i] [W] \* Alist[i] [] \* R[i]/(R[i]+tg) [-] | () |

input\_power[i] [W] = input\_power updated to watts and scaled for copper thickness, to be used in the rest of the code

input\_power[i] [W/m2] = assigned in CHICA 1.3.py

Updating the input power loops over the original list, therefore “i” refers to the list index being updated. Units have been included to be explicit regarding changes to the input\_power list.

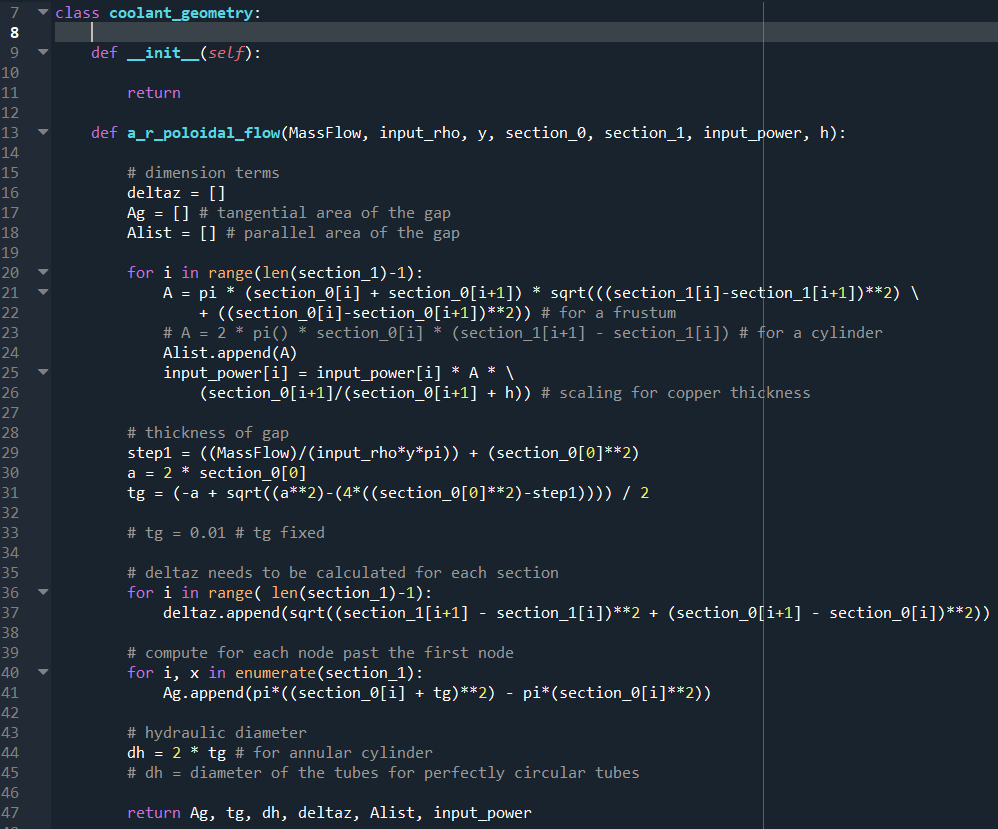
Alist[i] = defined in 2.3.6

R[i]/(R[i] + tg) = term to scale the power considering the copper thickness

tg = thickness of the copper

R[i] = section\_0[i], currently scales by the radius at the inlet of the Alist[i] section, may be updated to the average of the inlet and outlet radius

### coolant\_geometry code



## Non-Dimensional.py

### Parameter Definition

The class *non\_dimensional* contains two functions, *nusselt* and *film\_coeff* which compute the Nusselt number and film coefficient at a given discretised point. The outputs of these functions are detailed in Table 2‑3, where Figure 2‑2 presents the geometry in a cartesian coordinate system.

|  |  |  |  |
| --- | --- | --- | --- |
| **Parameter** | **Units** | **Type** | **Description** |
| film | W/(Km2) | Float | Film coefficient |
| Nu | - | Float | Nusselt number at each discretised point |
| re | - | Float | Reynolds number at each discretised point |
| pr | - | Float | Prandtl number at each discretised point |
| h\_1 | m | Float | Head loss at each discretised point |
| v\_s | m/s | Float | Bulk velocity at each discretised point |
| fricc | - | Float | Moody friction factor |

Table ‑ Non\_Dimensional.py Parameter definition

### Nusselt

### Bulk velocity,

|  |  |  |
| --- | --- | --- |
|  |  | (7) |

= bulk velocity

= mass flow rate, *MassFlow*

= Density, calculated with *CoolProp*

= Area, calculation in Section 2.3.3

= internal radius, section\_0[i]

### Reynolds number,

|  |  |  |
| --- | --- | --- |
|  |  | (8) |

= Reynolds number at each discretised point

= Density, calculated with *CoolProp* at constant pressure and temperature at a given discretised point

= bulk velocity,

= dh, Reynolds number calculation for a pipe uses hydraulic diameter

= kinematic viscosity, calculated with *CoolProp* at constant pressure and temperature at a given discretised point

### Pipe head loss,

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

= moody friction factor,

= considered length,

= bulk velocity,

= gravity,

= hydraulic diameter,

### Prandtl number,

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

= kinematic viscosity, calculated with *CoolProp* at constant pressure and temperature at a given discretised point

= specific heat capacity, calculated with *CoolProp* at constant pressure and temperature at a given discretised point

= thermal conductivity, calculated with *CoolProp* at constant pressure and temperature at a given discretised point

### Moody friction factor,

Moody friction factor

|  |  |  |
| --- | --- | --- |
|  |  | (11) |

### Nusselt number,

### film\_coef

|  |  |  |
| --- | --- | --- |
|  | input\_power[i] [W] = input\_power[i] [W] \* Alist[i] [] \* R[i]/(R[i]+tg) [-] | (12) |

## Runner.py

### initial