POLITECNICO DI TORINO

Corso di Laurea Magistrale
in Sustainable Nuclear Energy

Nuclear Fission Plant

Exercise 1: Natural Circulation and Passive Decay Heat Removal System

Professor
Davide PETTINARI
Cristina BERTANI

Student
Paolo GUIDI s331322
Lara MASSA s331285
Federico PATI s328987

2023/2024

Chapter 1

Introduction

The energy released in a reactor is produced by exothermic nuclear reactions in which part of the nuclear mass is transformed to energy: most of the energy is released when nuclei of heavy atoms split as they absorb neutrons and the splitting of these nuclei is called fission. The fission products are lighter atoms and fast neutrons, so the energy released by the fission process appears as kinetic and decay energy of the fission fragments and kinetic energy of the newborn neutrons. These fission fragments can be radioactive because instable and they undergo to beta-decay and gamma ray emission. The amount and activity of individual fission products and the total fission product inventory in the reactor fuel during operation and after shut-down are important to evaluate the amount of radiation and to determine the decrease of the fission product radioactivity in the spent fuel elements after removal from the reactor. Right after the insertion of a large negative reactivity to the reactor core (for example, due to an injection of control rods), the neutron flux rapidly decreases according to the following equation for a LWR operating with U-235:

$$\frac{\Phi(t)}{\Phi_0} = 0.067e^{-0.075t} + 0.933e^{-96.5t} \tag{1.1}$$

From the equation above is easy to understand that the second term is negligible already after a period of time of 0.01s: the neutron flux, and so also the power, immediately jumps to 6.7% of its initial value and then it is reduced e-fold. Summing up, after a reactor is shut down and the neutron flux may be neglected, substantial amounts of heat continue to be generated due to the beta particles and the gamma rays emitted by the fission products and this heat is called "Decay Heat". The decay heat must be exhausted to avoid a temperature increase and possible damages to the core, for this reason the Decay Heat Removal System (DHRS) is present. It can be:

- Active DHRS: a pump drives the coolant which extracts thermal power from decay heat, the coolant is cooled down in a heat exchanger which transfer the heat to a thermal sink;
- Passive DHRS: this system is based on natural circulation and the thermal sink is usually an in-pool heat exchanger.

Currently most of the operating reactors use active safety system, however, the will to overcame also rare problems during severe accident, led to the development of the newest philosophy of reactor design based on passive system.

Natural circulation present in DHRS, allows the system to operate even if external power is not available thanks to the density changes due to temperature differences. enhancing safety and reliability of the system.

In this report it will be analysed the Passive DHRS increasing gradually the degree of complexity. Starting from a simple loop driven by natural circulation and its optimization, to a more realistic simulation of the system.

Chapter 2

Natural Circulation

2.1 Case Study

In this first part of the report it will be analysed a basic example of a rectangular loop in Figure 2.1 , in which a fluid flows driven by buoyancy effects, due to a density gradient. A heater brings the fluid from saturated liquid to saturated steam, whereas a cooler brings the saturated steam to saturated liquid. The section of the pipe where the fluid is in liquid state (i.e., the cold side) is called cold leg, while the section of the pipe where the fluid is saturated steam (i.e., the hot side) is called hot leg. Moreover, in the report we

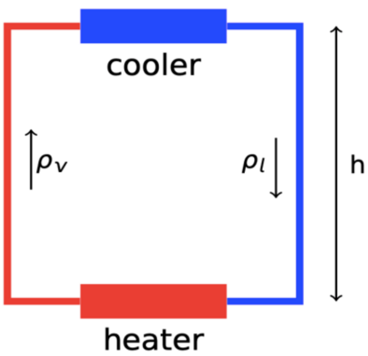

Figure 2.1: Scheme of the rectangular loop

be discussed three different situation:

1. With L fixed, find h such that h<L;
2. With L free parameter, find such that h = L;
3. Repeating the previous point considering also pipe roughness.

To solve these points, equations and iterative algorithms will be implemented in MATLAB. While the properties of the steam and liquid water will be evaluated with the help of XSteam.

2.2 Assumptions

The following simplifying assumptions are made:

- Both the heater and the cooler are assumed to be horizontal;
- Heat transfer in both the hot leg and cold leg is neglected;
- System is in steady state;
- Steam and liquid always at saturation;
- Constant pressure of $p_l = p_v = 70$ bar for liquid and steam phase .

2.3 Data

- Thermal power $P_{th} = 34.8 MW_{th}$;
- Length of each pipe L = 10 m;
- Constant pressure p = 70 bar;
- Pressure loss coefficient in the heater $k_v = 40$;
- Pressure loss coefficient in the cooler $k_l = 20$;
- Pipe diameters: will e considered diameters in Figure 3.2 contained in the STD column.

NPS (in)	Outside Diameter (in)	Schedule												
		10	20	30	STD	40	60	XS	80	100	120	140	160	XXS
		Wall Thickness (in)												
1/8"	0,405				0,068	0,068		0,095	0,095					
1/4"	0,540				0,088	0,088		0,119	0,119					
3/8"	0,675				0,091	0,091		0,126	0,126					
1/2"	0,840				0,109	0,109		0,147	0,147				0,187	0,294
3/4"	1,050				0,113	0,113		0,154	0,154				0,219	0,308
1"	1,315				0,133	0,133		0,179	0,179				0,250	0,358
1 1/4"	1,660				0,140	0,140		0,191	0,191				0,250	0,382
1 1/2"	1,900				0,145	0,145		0,200	0,200				0,281	0,400
2"	2,375				0,154	0,154		0,218	0,218				0,344	0,436
2 1/2"	2,875				0,203	0,203		0,276	0,276				0,375	0,552
3"	3,500				0,216	0,216		0,300	0,300				0,438	0,600
3 1/2"	4,000				0,226	0,226		0,318	0,318					
4"	4,500				0,237	0,237		0,337	0,337		0,438		0,531	0,674
5"	5,563				0,258	0,258		0,375	0,375		0,500		0,625	0,750
6"	6,625				0,280	0,280		0,432	0,432		0,562		0,719	0,864
8"	8,625		0,250	0,277	0,322	0,322	0,406	0,500	0,500	0,594	0,719	0,812	0,906	0,875
10"	10,750		0,250	0,307	0,365	0,365	0,500	0,500	0,594	0,719	0,844	1,000	1,125	1,000
12"	12,750		0,250	0,330	0,375	0,406	0,562	0,500	0,688	0,844	1,000	1,125	1,312	1,000
14"	14,000	0,250	0,312	0,375	0,375	0,438	0,594	0,500	0,750	0,938	1,094	1,250	1,406	
16"	16,000	0,250	0,312	0,375	0,375	0,500	0,656	0,500	0,844	1,031	1,219	1,438	1,594	
18"	18,000	0,250	0,312	0,438	0,375	0,562	0,750	0,500	0,938	1,156	1,375	1,562	1,781	
20"	20,000	0,250	0,375	0,500	0,375	0,594	0,812	0,500	1,031	1,281	1,500	1,750	1,969	
22"	22,000	0,250	0,375	0,500	0,375		0,875	0,500	1,125	1,375	1,625	1,875	2,125	
24"	24,000	0,250	0,375	0,562	0,375	0,688	0,969	0,500	1,219	1,531	1,812	2,062	2,344	
30"	30,000	0,312	0,500	0,625	0,375			0,500						
32"	32,000	0,312	0,500	0,625	0,375	0,688								
34"	34,000	0,312	0,500	0,625	0,375	0,688								
36"	36,000	0,312	0,500	0,625	0,375	0,750								
42"	42,000		0,500	0,625	0,375	0,750								

Figure 2.2: Pipe diameter from ANSI B36.10

2.4 Governing Equations

To model the loop the governing equations for the momentum balance fluid are the Navier-Stokes set of equations. After some simplifications, according to the physics phenomena, the equation becomes:

$$\Delta p_{buoyancy} = \Delta p_{loss}$$

$$(\rho_l - \rho_v)gh = f_l \frac{L_l}{2D} \rho v_l^2 + k_l \frac{\rho v_l^2}{2} + f_v \frac{L_v}{2D} \rho v_v^2 + k_v \frac{\rho v_v^2}{2} \tag{2.1}$$

While the equation for the energy balance is:

$$P_{th} = \dot{G} A (H_v - H_l) \tag{2.2}$$

2.5 Case 1

As anticipated, in the first part we will consider the length of each smooth pipe fixed. The aim is to find the value of the height h, fixing the diameter D, such that the thermal power is power is removed by natural circulation.

First of all, using XSteam at the given pressure, density, temperature, enthalpy and viscosity are evaluated at saturation condition, for both steam and liquid water. Using the energy equation, the mass flow rate could be computed as:

$$\dot{m} = \frac{P_{th}}{H_v - H_l} \tag{2.3}$$

Now, in order to have a better representation of the phenomena, it has been chosen to consider the diameter as a continuous parameter, not considering only the discrete value given by Figure 3.2, that remains our final reference.

From the value of the mass flow rate, and the area of the specific pipe, we can evaluate the velocities for steam and liquid water and the correspondent Reynolds number.

$$Re_l = \frac{\rho_l v_l D}{\mu_l} \qquad\qquad Re_v = \frac{\rho_v v_v D}{\mu_v} \tag{2.4}$$

Subsequently, the friction factor is computed with the following semi-empirical correlation, considering smooth pipe:

$$f = \frac{0{,}316}{Re^{0{,}25}} \tag{2.5}$$

This correlation is valid for Re > 3000, and it will be verified later.

Computing all we need, from 2.1 we can evaluate values of the height as follows:

$$h = \frac{f_l \frac{L_L \rho_l v_l^2}{2D} + k_l \frac{\rho_l v_l^2}{2} + f_v \frac{L_v \rho_v v_v^2}{2D} + k_v \frac{\rho_v v_v^2}{2}}{(\rho_l - \rho_v) g} \tag{2.6}$$

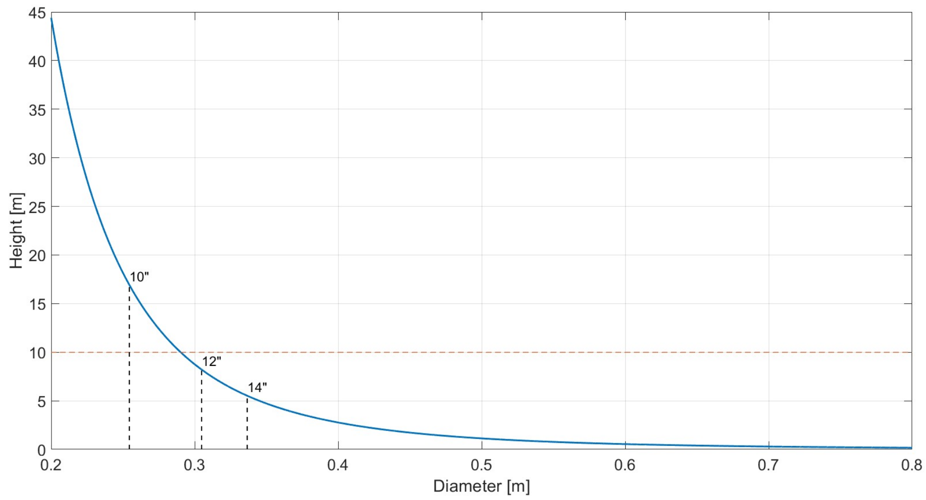

Figure 2.3: Corresponding height for different diameters.

can be computed starting and the result are represented in Figure 2.3

From the graph we can deduce that the value of diameter needed in order to have h = L, is given by the intersection of the black dashed line and the blue curve. Its value is around D = 0.29 m. However, diameter is constrained by industrial standards as already mention in Figure 3.2. For this reason in the plot, value of NPS diameters in inches are also reported in red. This value are deduced starting from the internal diameter and the STD wall thickness. For this case, an NPS diameter of 12" (Outside Diameter = 12.75 in , Inside Diameter = 12 in) would be the best choice to achieve our objective.

Finally, the value of the Reynolds number is checked and this is well above the correlation limit, so its use is legit.

2.6 Case 2 and 3

Second and third cases are different from the first one, since L is now a free parameter.

In the second case the pipe is still considered a smooth pipe, while in the third case the pipes are assumed to have different roughness: $\epsilon = 5e-4, 5e-5, 2.5e-5$.

For smooth pipe, except from the last equation, the procedure is the same. For rough pipe, Correlation 2.5 is not valid anymore. Now, the friction factor is computed using Colebrook correlation for steam and liquid:

$$\frac{1}{\sqrt{f}} = -2\log_{10}\left(\frac{2.51}{Re\sqrt{f}} + \frac{\frac{\epsilon}{D}}{3.71}\right) \tag{2.7}$$

Finally, starting from Equation 2.1, different values of h = L are computed as follows:

$$h = \frac{k_l \frac{\rho_l v_l^2}{2} + k_v \frac{\rho_v v_v^2}{2}}{(\rho_l - \rho_v) g - f_l \frac{\rho_l v_l^2}{2D} - f_v \frac{\rho_v v_v^2}{2D}} \tag{2.8}$$

The result are represented together in Figure 2.4

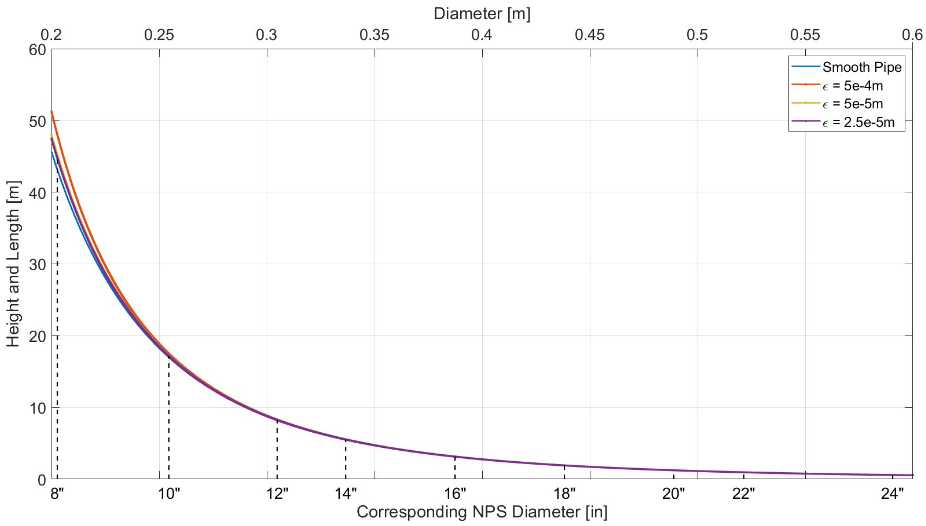

Figure 2.4: Corresponding height and length for different diameters and roughness.

While, in the first case, the size of the loop was fixed, here an economical consideration can help in the choice of the design. The first consideration that could be deduced is the existence of a knee in the curve. Before the knee a small reduction of the diameter causes a great increase in the length of the loop, since it would increase pressure losses. While pipes with smaller diameters would be cheaper, much longer pipes makes it not worthy. After the knee, the slope of the curve decreases sensibly. An increase in the diameter doesn't change that much the length of the system, so, again, it wouldn't be convenient, In the knee of the curve we have the best trade off between pipes and length, and this happens for the Normal Pipe Size of 12 inches and a length similar to the previous case of 10 or 9m, approaching the minimum height for this diameter might be dangerous due to all the simplification assumed, so a safety margin could be a reasonable choice.

The second observation is that different roughnesses of the pipe result in different lengths only for smaller diameters and longer pipes. After 10 inches, the curves are superimposed and the differences are almost imperceptible. For this reason, the roughest pipe is good enough to carry out the same result, saving money.

Chapter 3

Passive Decay Heat Remova System

3.1 Case Study

This report covers an example of DHRS operating in natural circulation, so a passive DHRS. The system is made of three circuits:

- The Primary System Circuit (PSC) which transfers the heat produced in the core to the next circuit via a liquid-liquid heat exchanger;
- The Intermediate System Circuit (ISC), which transfers the heat to a pool in which water boils at atmospheric pressure;
- The Tertiary System Circuit (TSC), which transfers heat to the final exchanger where the steam produced in the pool condenses inside an air-cooled exchanger at atmosphere conditions.

The DHRS is not active during nominal operations, in fact is present a check valve located in the cold leg blocks the circulation in the PSC. This valve works in accidental conditions if the primary coolant pump stops and the actuator is triggered by a pressure differential across the check valve. The density difference between the cold fluid in the vessel downcomer and the hot fluid in the core triggers natural circulation:

- The PSC hot leg is gradually heated up, and so does the primary side of HX1;
- This leads to circulation entering the ISC, which heats the pool side of the HX2 until saturation is reached;
- The steam produced in the pool is directed to the cooling tower through the TSC.

The DHRS is not active during nominal operations, in fact is present a check valve located in the cold leg blocks the circulation in the PSC. This valve works in accidental conditions if the primary coolant pump stops and the actuator is triggered by a pressure

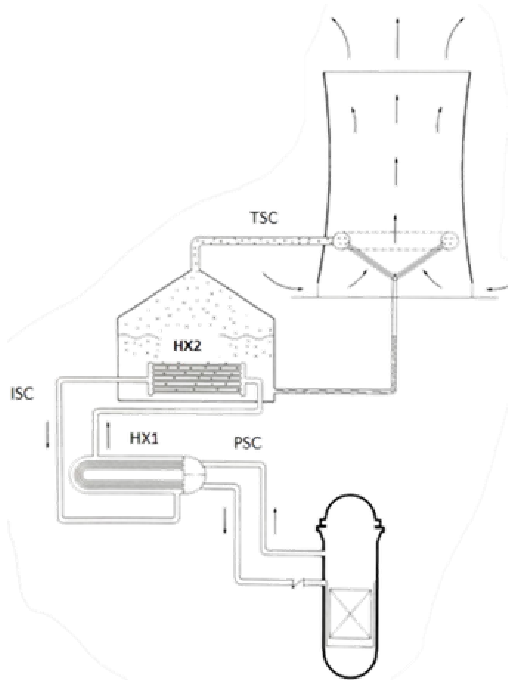

Figure 3.1: Representation of the system.

differential across the check valve. The density difference between the cold fluid in the vessel downcomer and the hot fluid in the core triggers natural circulation:

- The PSC hot leg is gradually heated up, and so does the primary side of HX1;
- This leads to circulation entering the ISC, which heats the pool side of the HX2 until saturation is reached;
- The steam produced in the pool is directed to the cooling tower through the TSC.

3.2 Data

3.2.1 General data

- Nominal thermal power Pnom = 600 MWth, when the steady condition is reached the power is reduced to the value of 1% of the nominal one: P=6 MW;
- Vessel pressure drop Δp_{vessel} = 1.2 bar for $\dot{m}$ = 3200kg/s with a constant loss coefficient with the flow rate.

3.2.2 PSC

Data about the Primary System Circuit are reported in Table 3.1:

Data	**Symbol**	**Value**	**Units**
Coolant pressure	p	75	bar
Outer pipe diameter	D_{out}	16 (schedule 100)	inches
Pipe length	L	16	m
Pipe relative roughness	ϵ/D	$2 \cdot 10^{-4}$	-
Pressure loss coefficient for 90° bends	k_{loss}	0.45	-
Number of 90° bends	N_{bends}	4	-
Valve pressure loss coefficient	$k_{\text{loss,valve}}$	0.12	-
Elevation between hot and cold leg	H_1	7	m
Elevation between top and bottom of the core	H_2	3	m

Table 3.1: PSC data

3.2.3 HX1

HX1 is a counter current, shell and tubes HX with U tubes, with a triangular lattice.

Data	**Symbol**	**Value**	**Units**
Outer tube diameter	$D_{\text{out,tube}}$	19.05	mm
Tube thickness	t	1.24	mm
Number of tubes	N_t	897	-
Pitch	d_p	28.5	mm
Shell inner diameter	D_{shell}	1.5	m
Number of baffles	N_b	2	-
Baffles spacing	l_{baffles}	1.6	m
Average tube length	L_{tubes}	9.314	m
Tube thermal conductivity	k_{th}	15	$\text{Wm}^{-1}\text{K}^{-1}$
Headers average flow area	A_h	0.883	m^2
Heat transfer area	$A_{\text{ht,HX1}}$	500	m^2
Tube relative roughness	ϵ/D	10^{-4}	-
Correction factor for ΔT_{ml}	F_T	0.7	-

Table 3.2: HX1 data

3.2.4 ISC

Data about the Intermediate System Circuit are reported in Table 3.3:

Data	**Symbol**	**Value**	**Units**
Coolant pressure	p	70	bar
Outer tube diameter	$D_{\text{out,tube}}$	16 (schedule 100)	inches
Total pipe length	L_{pipe}	40	m
Pressure loss coefficient for 90° bends	k_{loss}	0.45	
Number of 90° bends	N_{bends}	6	
Tube relative roughness	ϵ/D	$2 \cdot 10^{-4}$	
Net elevation	H_{ISC}	10	m

Table 3.3: ISC data

3.2.5 HX2

HX2 is an in-pool type heat exchanger.

Data	**Symbol**	**Value**	**Units**
Tube outer diameter	$D_{\text{out,tube}}$	25.4	mm
Tube thickness	t	1.24	mm
Number of tubes	N_t	770	
Average tube length	L_{tubes}	7	m
Manifold diameter	D_{manifold}	16	inches
Tube thermal conductivity	k_{th}	15	$\text{Wm}^{-1}\text{K}^{-1}$
Heat transfer area	$A_{\text{ht,HX2}}$	430	m^2
Tube relative roughness	ϵ/D	10^{-4}	

Table 3.4: HX2 data

3.3 Assumptions

The following simplifying assumptions are made:

- The system is in steady state, the initial transient following reactor shutdown is neglected;
- The cooling tower exhausts all the power necessary to keep the pool at saturation conditions;
- Constant coolant pressure in the three circuits to compute water and steam properties;
- No fouling resistance in the heat exchangers;
- Correction factor for the mean log temperature calculations (HX1 is not a pure counter current HX) equal to 0.7 in HX1.

3.4 Objectives

The goal is to analyse the DHRS at steady state conditions, after its initial activation and the reaching of saturation conditions in the pool side of HX2. The two quantities of interest are:

1. The mass flow rates in the PSC and ISC;

2. The temperatures of the hot and cold leg in the PSC and ISC.

3.5 Correlation

3.5.1 Pressure losses

Observing the behaviour of the flows and so evaluating the Reynolds numbers, the common result is that a turbulent flow is always present. For viscous flow, as in this case, head losses are expressed with two different components:

$$h_L = h_{L_{major}} + h_{L_{minor}} \tag{3.1}$$

1. Major or distributed losses: represent energy losses due to friction of the fluid in a conduit caused by effect of the fluid's viscosity a wall roughness. This head loss can be expressed as:
$$h_{L_{major}} = f \frac{\ell}{D} \frac{V^2}{2g} \tag{3.2}$$
Where the friction factor for rough pipe is computed with the Colebrook correlation 2.7 already used.

2. Minor or concentrated losses: represent the energy losses due to additional components as valves and bends:
$$h_{L_{minor}} = K_L \frac{V^2}{2g} \tag{3.3}$$
K_L coefficients are usually given as assigned value of the different components.

DHRS is based on the same physics principle of natural circulation, so as the previous exercise, although it was less complex, the governing equation is the still the simplified form of the Navier-Stokes equations 2.1.

3.5.2 Global heat transfer coefficient

For a heat exchanger the global heat transfer coefficient is given by three thermal resistances:

$$R_{tot} = R_{inner\ side} + R_{thermal} + R_{outer\ side} \tag{3.4}$$

$R_{inner\ side}$ and $R_{outer\ side}$ are related to convective heat transfer coefficient, while $R_{thermal}$ is due to the conduction within the tube wall. The resulting coefficient is:

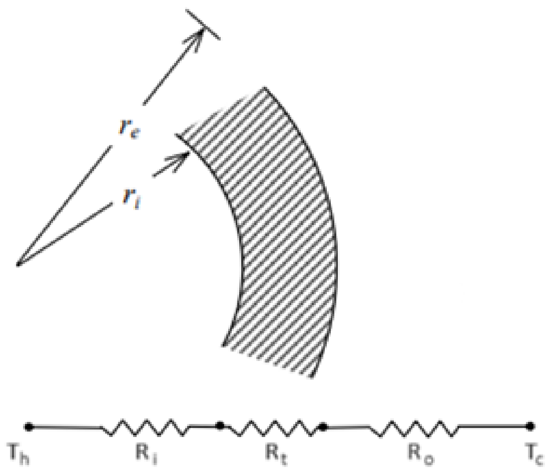

Figure 3.2: Total thermal resistance with a cylindrical wall.

$$U_{out} = \frac{1}{\left(\left(\frac{r_{out}}{h_{in} * r_{in}} \right) + \left(r_{out} * \frac{\ln\left(\frac{r_{out}}{r_{in}}\right)}{k} \right) + \left(\frac{1}{h_{out}} \right) \right)} \tag{3.5}$$

3.6 Simulation setup

In this analysis the components are studied one at time and the shared properties were then joined together again through the same iterative cycle. For what concern the properties of the water it is used the library XSteam for Matlab. The approach used in this analysis for the two heat exchanger follows these steps:

1. Hpothesize the temperature of the two legs;
2. Compute the differential temperature and the dimensionless numbers: Reynolds, Prandtl and Nusselt;
3. Compute the global heat transfer coefficient;
4. Compute the logarithmic mean temperature difference and evaluate the two temperature in the legs;
5. Compute the density difference between the two legs and recalculate the Reynolds numbers;
6. Compute the friction factor in hot leg, cold leg, HX1 and HX2;
7. Compute the pressure losses;
8. Compute the new mass flow rate.

3.7 ISC

To fully analyse the Intermediate System Circuit, is essential to define also heat tranfer in HX1 and HX2, recreating a loop analogous to the first exercise.

1. In this exercise the two temperatures for the ISC legs must be hypothesized: $T_{cold} = 100°C$ and $T_{hot} = 140°C$, with an average temperature of $T_{ave} = 120°C$ as suggested. The mass flow rate guessed is $\dot{m}_{ISC} = 100\frac{kg}{s}$.

2. The dimensionless numbers are evaluated for HX2:

   - Reynolds:
   $$Re = \frac{4\dot{m}_{ISC}}{N_{tubes}D\pi\mu} \tag{3.6}$$

   - Prandtl:
   $$Pr = \frac{c_p\mu}{k} \tag{3.7}$$

   - Nusselt: is used the Dittus-Boelter correlations:
   $$Nu = 0{,}023Re^{0{,}8}Pr^n \tag{3.8}$$

   valid for:

     - $0.7 \leq Pr \leq 160$
     - $Re_D \geq 10000$
     - $\frac{L}{D} \geq 10$

   With n=0.4 for heating.

   The values of μ, k and c_p are given by XSteam evaluated to the leg temperature and to the ISC pressure.

3. To compute the global heat transfer is important to evaluate the following values:

   - Convective heat transfer coefficient of the internal part of the tube:
   $$h_{int} = \frac{kNu}{D_{int}} \tag{3.9}$$

   - Convective heat transfer coefficient of the internal part of the tube:
   $$h_{out} = q''\Delta T_{sat} \tag{3.10}$$

   Where:
   $$\Delta T_{sat} = (\frac{P}{2.257A_{tot,HX2}})^{\frac{1}{3.86}} \tag{3.11}$$

   Now the global heat transfer can be evaluated with the thermal resistances with the equation 3.5.

With U_{out} is possible to calculate the logarithmic temperature with HX2 treated as a co-current:

$$\Delta T_{ml} = \frac{(T_{in_{HX2}} - T_{water}) - (T_{out_{HX2}} - T_{water})}{\ln \frac{(T_{in_{HX2}} - T_{water})}{(T_{out_{HX2}} - T_{water})}} = \frac{P}{U_{out} A_{tot_{HX2}}} \tag{3.12}$$

Where P can be computed as:

$$P = c_p \ \dot{m_D} \ \left(T_{in_{HX2}} - T_{out_{HX2}}\right) \tag{3.13}$$

The T_{water}, as said in the assumptions, is take equal to the saturation one at atmospheric pressure, so $T_{water} = 100řC$. The system is composed of two equations with only two unknowns that are the two temperatures of HX2, therefore it can be solved by making one of the two temperatures explicit.

4. The two new densities can be computed with XSteam using the two new temperatures $T_{in_{HX2}}$ and $T_{out_{HX2}}$, so also the new Reynolds.

5. For the ISC, the pressure losses of the coolant are composed by the following terms:

   a. Distributed losses evaluated with Colebrook correlation 2.7 for the cold and hot leg, and for the HX2, where the water flows into the pipes. Properties of the fluid are evaluated respectively at:

      i. $T_{out_{HX2}}$ for the cold leg;
      ii. $T_{in_{HX2}}$ for the hot leg ;
      iii. $T_{ave_{HX2}}$ the average temperature between the previous two for HX2.

   b. Concentrated losses:

      i. For HX1 the shell side give a local loss coefficient that can be computed as:

$$k_{shell} = 8 \frac{0.227}{Re_{shell}^{0.193}} \frac{D_{shell}}{D_{eq,\ shell}} \left(n_{baffle} + 1\right)] \tag{3.14}$$

$$D_{eq,\ shell} = \frac{2\sqrt{3} d_p^2}{\pi D_{tube}} - D_{tube} \tag{3.15}$$

$$Re_{shell} = \frac{\rho \ v_{shell} \ D_{eq,shell}}{\mu} \tag{3.16}$$

$$v_{shell} = \frac{m_{\dot{ISC}}}{\rho \ A_{shell}} \tag{3.17}$$

$$A_{shell} = \frac{D_{shell}}{d_p} \left(d_p - D_{tube}\right) l_{baffles} \tag{3.18}$$

Where density and viscosity are evaluated with XSteam at the average temperature between hot and cold legs.

ii. For HX2 there are the pressure losses given by:

1. Expansion from the pipe to the collector:

$$K_{Lenl_{hot_{side}}} = \left(1 - \frac{A_{pipe_{ISC}}}{A_{collector}}\right)^2 \tag{3.19}$$

2. Contraction from the collector to the tubes:

$$K_{Lcontr_{hot_{side}}} = \frac{1}{2}\left(1 - \frac{A_{tube_{HX2}}}{A_{collector}}\right) \tag{3.20}$$

3. Expansion from the tubes to the collector:

$$K_{Lenl_{cold_{side}}} = \left(1 - \frac{A_{tube_{HX2}}}{A_{collector}}\right)^2 \tag{3.21}$$

4. Contraction from the collector to the pipe:

$$K_{Lcontr_{cold_{side}}} = \frac{1}{2}\left(1 - \frac{A_{pipe_{ISC}}}{A_{collector}}\right) \tag{3.22}$$

iii. ISC pressure losses for 90° bends where 3 are considered with pressure in hot leg and the other 3 in the cold one:

$$K_{L_{each\ bend}} = 0.45 \tag{3.23}$$

7. Now it is used the modified Bernoulli's equation 2.1, implemented in the *fzero* function of MATLAB to compute the new mass flow rate.

With the newest value of mass flow rate, this iterative procedure starts again and goes through this points until an error condition is satisfied:

$$\epsilon_{rel} = \frac{\dot{m}_{new} - \dot{m}_{old}}{\dot{m}_{old}} < 1e - 8 \tag{3.24}$$

3.8 PSC

For the definition of thermo-fluid dynamic properties of the water in the Primary System Circuit, it will be analyzed also the heat transfer in HX1 and the reactor core. Also for this system an analogous iterative process is required:

1. First of all, temperatures are hypothesized for the PSC legs: $T_{cold} = 130\ °C$ and $T_{hot} = 150\ °C$, with an average temperature of $T_{ave} = 140\ °C$. The mass flow rate guessed here is $\dot{m}_{PSC} = 100\frac{kg}{s}$.

2. The dimensionless numbers are evaluated for the fluid in the heat exchanger HX1:

- Reynolds:

$$Re = \frac{4m\dot{_{PSC}}}{D_{in,\ HX1}\ \pi\mu} \tag{3.25}$$

- Prandtl: with equation 3.7
- Nusselt: is used the Dittus-Boelter correlation 3.8 with $n = 0.4$.

The values of $\mu,\ k\ and\ c_p$ are given by XSteam evaluated to the leg temperatures and to the ISC pressure.

3. To compute the global heat transfer is important to evaluate the following values:

   - Convective heat transfer coefficient of the inner side of the tube: find h_{in} with 3.9.
   - Convective heat transfer coefficient of the outer side of the tube, i.e. the shell side:

$$h_{out} = 0.351 * Re_{shell}^{0.55} * \frac{k_{th}}{D_{eq,shell}} * \mathrm{Pr}^{\frac{1}{3}} \tag{3.26}$$

   Where Re_{shell} is calculated from 3.16.

   Now the global heat transfer can be evaluated with the thermal resistances with the equation 3.5.

4. Evaluated U_{out}, now is possible to calculate the logarithmic temperature, where HX1 is considered as a counter-current exchanger:

$$\Delta T_{ml} = \frac{(T_{in_{HX1}} - T_{in_{HX2}}) - (T_{out_{HX1}} - T_{out_{HX2}})}{\ln\frac{(T_{in_{HX1}} - T_{in_{HX2}})}{(T_{out_{HX1}} - T_{out_{HX2}})}} = \frac{P}{U_{out} * A_{tot_{HX1}} * F_T} \tag{3.27}$$

$$P = c_p \; \dot{m}_{PSC} \; (T_{in_{HX1}} - T_{out_{HX1}}) \tag{3.28}$$

The two temperatures, in and out of HX2, are taken from the point 4 of the previous analysis on ISC circuit. The coefficient F_T is used to correct the mean log temperature since HX1 is not a pure counter current. In these two equations, the unknowns are the two temperatures of HX1. To solve this non-linear system the function *fsolve* is implemented.

5. The two new densities can be computed with XSteam using the new temperatures $T_{in_{HX1}}$ *and* $T_{out_{HX1}}$, so also the new Reynolds.

6. The pressure losses felt by the coolant in the PSC are composed by the following terms:

   a. Distributed losses evaluated with Colebrook correlation 2.7 for the cold and hot leg, and for the HX1, where the water now flows into the pipes. Properties of the fluid are evaluated respectively at:

      i. $T_{out_{HX1}}$ for the cold leg;
      ii. $T_{in_{HX1}}$ for the hot leg ;
      iii. $T_{ave_{HX1}}$ the average temperature between the previous two for HX1.

   b. Concentrated losses:

      i. For HX1 there are the pressure losses given by:

         1. Expansion from the pipe to the header:

$$K_{Lenl_{hotside}} = \left(1 - \frac{A_{pipe_{PSC}}}{A_{header}}\right)^2 \tag{3.29}$$

         2. Contraction from the header to the tubes:

$$K_{Lcontr_{hotside}} = \frac{1}{2}\left(1 - \frac{A_{tube_{HX1}}}{A_{header}}\right) \tag{3.30}$$

         3. Expansion from the tubes to the header:

$$K_{Lenl_{coldside}} = (1 - \frac{A_{tube_{HX1}}}{A_{header}})^2 \tag{3.31}$$

         4. Contraction from the header to the pipe:

$$K_{Lcontr_{coldside}} = \frac{1}{2}(1 - \frac{A_{pipe_{PSC}}}{A_{header}} \tag{3.32}$$

      ii. Two PSC pressure losses for 90° bends in the hot leg and two in the cold one:

$$K_{L_{each\ bend}} = 0.45 \tag{3.33}$$

      iii. Due to the check valve located in the cold leg there is a pressure loss with:

$$K_{L_{valve}} = 0.12 \tag{3.34}$$

7. Now it is used again the modified Bernoulli's equation 2.1, but now the density in the core varies from the bottom to the top. Hence, it is needed to compute the contribution of the driving force from the density difference in the core. If it is assumed that the density changes linearly with the height it is possible to write:

$$rho(z) = \rho_{cold} + (\rho_{hot} - \rho_{cold}) * \frac{z}{H_2} \tag{3.35}$$

   So, the left term of 2.1 become:

$$\int_0^{H_2} \rho_{(z)}\ g\ dz + (\rho_{cold} - \rho_{hot})\ g\ (H_1 - H_2) = g\ H_2 \left(\frac{3}{2}\rho_{hot} - \frac{1}{2}\rho_{cold}\right) - g\ H_1(\rho_{hot} - \rho_{cold}) \tag{3.36}$$

This equation can be solve with the function *fzero* to evaluate the new mass flow rate and an iterative process starts again. Similarly to the ISC, the stopping criteria for the iteration is the relative error on the mass flow as reported in 3.24.

3.9 Energy conservation

Finally, to validate further the result, could be a matter of interest the energy conservation between the two systems. Equation 3.13 and equation 3.28, will be compared both with the powers exchanged with the two heat exchaner as it follows:

$$P = U_{out}\ A_{tot}\ F_T\ \Delta T_{ml} \tag{3.37}$$

This two power must be the same, since it is a necessary (but not sufficient) condition.

3.10 Results

3.10.1 Computed value

The quantities of interest, with the conditions stated as before reported, are:

	Mass flow rate [kg/s]	**Temperature in the hot leg [°C]**	**Temperature in the cold leg [°C]**
ISC	64.59	131.25	109.29
PSC	58.85	151.40	127.50

Table 3.5: Quantities of interest of the simulation.

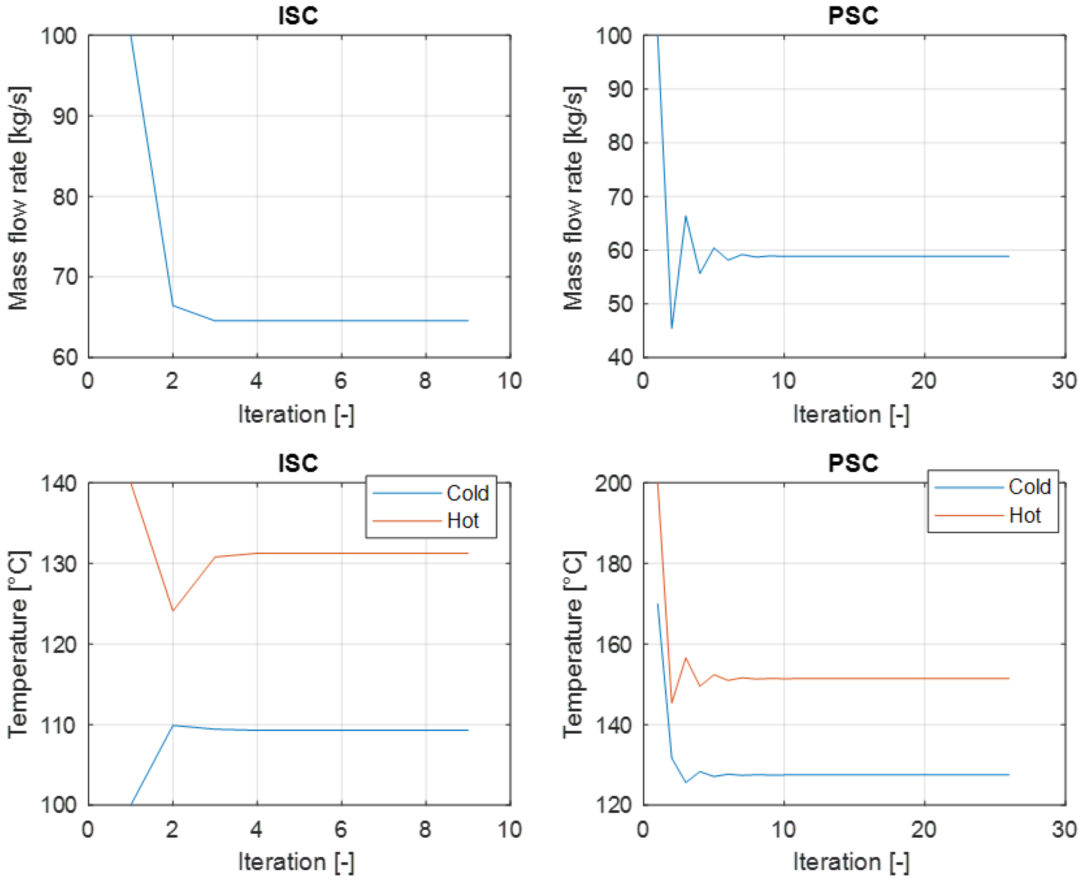

Figure 3.3: Quantities of interest as function of the number of iterations.

3.10.2 Iterations

It is interesting to know also how the above quantities vary with the number of iterations. The convergence of the solution is reported in Figure 3.3:

It is possible to understand quickly that ISC's results arrive to the convergence with a smaller number of iterations, instead PSC requires more iterations. This could be related to different factors, such as the initial value guessed or the more complex equation solved.

3.10.3 Energy conservation

As previously said, energy conservation is mandatory.The value of energy transported by the coolant in the PSC, exchanged into HX1, transferred to ISC and so to the HX2 must be the same. This value of energy per unit of time need to be equal to 6 MW extracted from the core.

Figure 3.4 is useful to understand two different meanings:

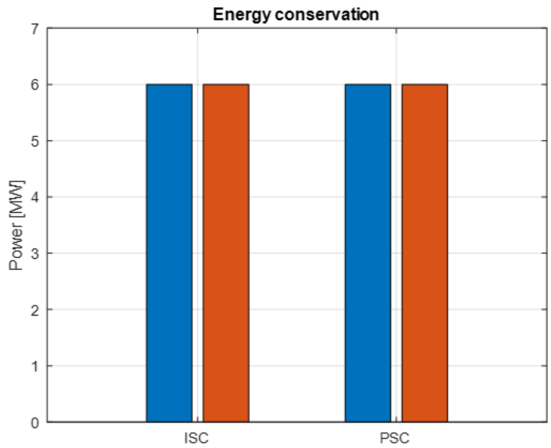

Figure 3.4: Energy conservation of the two circuits.

1. Mathematical meaning: the mean logarithmic temperature and the two leg temperatures for each leg are computed rightly;

2. Physical meaning: the energy is conserved between the two circuits.

If the energy is conserved the simulation works good and can reproduce reliable results.

3.10.4 Pressure losses distribution

A resume of the different pressure losses is reported in Table 3.6:

	Distributed	**Concentrated**
ISC	- HX2 tubes - Hot and cold legs	- Shell - Expansion and contraction - 3+3 90° bends
PSC	- HX1 tubes - Hot and cold legs	- Check valve - Expansion and contraction - 3+3 90° bends

Table 3.6: Distributed and Concentrated Components for ISC and PSC

The contributes of this different types of pressure losses are reported in Figure 3.5 In the ISC the Buoyancy term (represented in blue even if it is not a distributed losses, but a pressure "gain") is balanced from concentrated losses in the shell of HX1, the distributed losses in HX2 and both losses in the two legs. The dissimilarities between pressure losses in cold and hot leg are not mainly a consequence of different density or velocity. The geometry of the system, hence contractions and enlargements, are more influential losses. In the PSC, instead of HX2, are present vessel pressure drop. This loss is indicated as

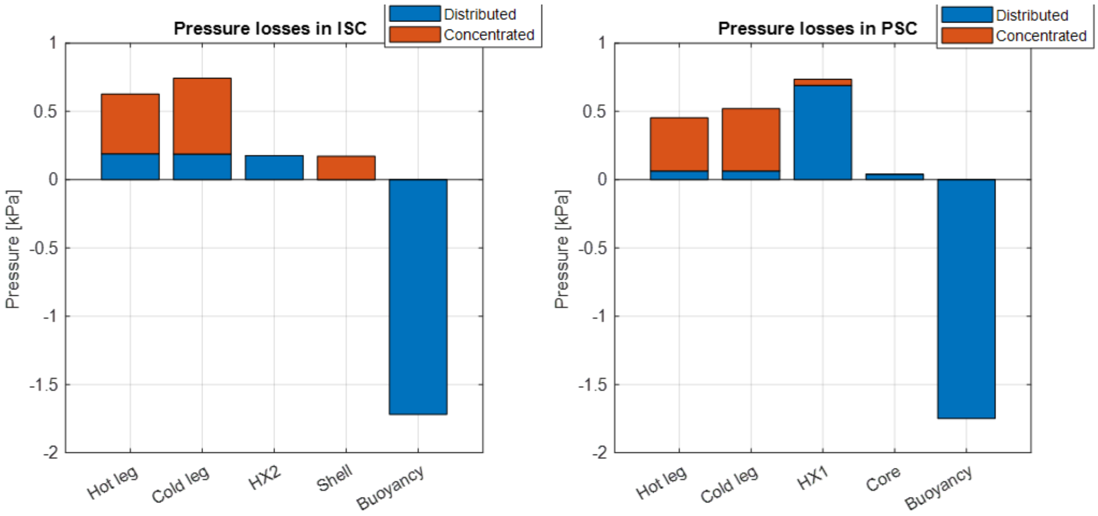

Figure 3.5: Pressure losses distribution in the two circuits.

distributed, but it is not properly a distributed losses because it's a comprehensive term which is assumed to be dependent on the mass flow rate. Moreover we can notice that the distributed losses in PSC given by the HX1 tubes are very significant. This is probably due to the large number of tubes and their small diameter.

Chapter 4

Conclusions

In conclusion in this two exercise it has been analyzed, at first, a quite simple application of natural circulation, then a more realistic implementation in a Passive Decay Heat Removal System, with coupled circuit.

- From the first exercise, the simpler physics of the problem, allows us to bring the focus of the analysis primarily on economical aspect. From the implementation of the conservation equations in an iterative process, interesting results were carried out. The knee of the curve is the most attractive region for the economical trade-off between larger pipes and longer ones.
- In the second exercise, since the design of the system was defined, the focus of the analysis was mainly the mathematical and physical discussion to solve the problem. With an iterative algorithm we reached the aim of evaluating the mass flow rate and the temperature of the coolant in both the systems. With this flow rate, the system is able to remove safely, and without the use of auxiliary systems, all the heat produced some hours after the shutdown, when it reaches 1% of the nominal value. However, should be mention that an economical analysis could be discussed. Lower value of concentrated losses for the legs or smaller distributed losses in HX1 could allows to save money reducing the height or the diameter or the pipes. A lot of parameters contributes and could also influence simultaneously in two different ways the behaviour of the system.

Appendix A

A.1 Exercise 1.1

```matlab

clear all;close all; clc;

P_th = 34.8e6; %Wth
% steady state, no heat loss
p =70; %bar
LL = 10; %m
k_h = 40;
k_c = 20;
delta_T = 0.001; %  C

%% STEP 1:Fluid properties
rhoL = XSteam('rhoL_p',p);
rhoV = XSteam('rhoV_p',p);

Tsat = XSteam('Tsat_p',p);

hV = XSteam('hV_p',p)*1000;
hL = XSteam('hL_p',p)*1000;

mycold = XSteam('my_pT',p,Tsat-delta_T);
myhot = XSteam('my_pT',p,Tsat+delta_T);

%% STEP 2 and 3: Flow rate evaluation and Diameter


conversion = 0.0254;
DD_out=[8.625 10.750 12.750 14 16 18 20 22 24 1000 ]*conversion;
STD=[0.322 0.365 0.375 0.375 0.375 0.375 .375 0.735 0.375 1]*
    conversion;
DD_int = DD_out-2*STD;
DD_int = round(DD_int,4);

```

```
index = 1;
ii_saved= [];
DD_vec = 0.2:1e-4:0.8;

for ii = 1: length(DD_vec)
    DD = DD_vec(ii);
    Area = pi/4*DD^2;

    if round(DD,4) == DD_int(index) && index<5;
            ii_saved(index) = ii;
            index = index +1;
    end

    GG = P_th/Area/(hV-hL); %rho*vel
    mm =  P_th/(hV-hL);

    %% STEP 4: obtain steam and dynamic viscosity
    vL = GG/rhoL;
    vV = GG/rhoV;

    %% STEP 5: Find Re and f so a new D
    ReV = rhoV*vV*DD/myhot;
    ReL = rhoL*vL*DD/mycold;

    % form correlations
    fv = 0.316/ReV^0.25;
    fl = 0.316/ReL^0.25;

    newh = (fl*LL*0.5/DD*rhoL*vL^2 + k_c*0.5*rhoL*vL^2+fv*LL/DD
   *0.5*rhoV*vV^2+k_h*0.5*rhoV*vV^2)/((rhoL-rhoV)*9.81);
    newh_vec(ii) = newh;
end
plot(DD_vec,newh_vec,'LineWidth',2);
hold on
grid on
xlim([0.2,0.8])
plot(DD_vec(1:ii),LL*ones(ii,1),'--','Color',   "#D95319")
xlabel('Diameter [m]');
ylabel('Height [m]')

x_points = DD_int(2:4);
y_points = newh_vec(ii_saved(2:4));
point_text= {'8"','10"','12"','14"','16"','18"','20"','22"','24"'
   };

for k = 1:length(x_points)

   plot([x_points(k) x_points(k)], [0 y_points(k)], 'k--', '
   LineWidth',1.2);
```

```
    grid on
    text(x_points(k), y_points(k), sprintf('%s', point_text{k+1}), 'VerticalAlignment','bottom', 'HorizontalAlignment','left', 'FontSize',14);
end
ax = gca;
ax.FontSize = 16;

%% GOAL 2
DD_vec = 0.2:1e-6:0.7;
for ii = 1: length(DD_vec)

    DD = DD_vec(ii);
    Area = pi/4*DD^2;

    GG = P_th/Area/(hV-hL); %rho*vel
    mm =  P_th/(hV-hL);

    %% STEP 4: obtain steam and dynamic viscosity
    vL = GG/rhoL;
    vV = GG/rhoV;

    mycold = XSteam('my_pT',p,Tsat-delta_T);
    myhot = XSteam('my_pT',p,Tsat+delta_T);

    %% STEP 5: Find Re and f so a new D
    ReV = rhoV*vV*DD/myhot;
    ReL = rhoL*vL*DD/mycold;

    % form correlations
    fv = 0.316/ReV^0.25;
    fl = 0.316/ReL^0.25;

    newh = (k_c*rhoL/2*vL^2+k_h*rhoV/2*vV^2)/((rhoL-rhoV)*9.81-fl*rhoL*vL^2/2/DD - fv*rhoV*vV^2/2/DD);
    %newh = (fl*L*0.5/DD*rhoL*vL^2 + k_c*0.5*rhoL*vL^2+fv*L/DD*0.5*rhoV*vV^2+k_h*0.5*rhoV*vV^2)/((rhoL-rhoV)*9.81);
    newh_vec(ii) = newh;
end
figure(2)
plot(DD_vec,newh_vec,'LineWidth',1.5);
hold on


%% GOAL 3
roughness_vec = [5e-4, 5e-5, 2.5e-5];
index = 1;
ii_saved = [];
for jj = 1:3
```

```
    roughness = roughness_vec(jj);
    newh_vec = [];
    DD_vec = 0.2:1e-4:0.6;
    for ii = 1: length(DD_vec)

        DD = DD_vec(ii);
        Area = pi/4*DD^2;

        if round(DD,4) == DD_int(index) && index<10;
            ii_saved(index) = ii;
            index = index +1;
        end

        GG = P_th/Area/(hV-hL); %rho*vel
        mm =  P_th/(hV-hL);

        %% STEP 4: obtain steam and dynamic viscosity
        vL = GG/rhoL;
        vV = GG/rhoV;

        mycold = XSteam('my_pT',p,Tsat-delta_T);
        myhot = XSteam('my_pT',p,Tsat+delta_T);

        %% STEP 5: Find Re and f so a new D
        ReV = rhoV*vV*DD/myhot;
        ReL = rhoL*vL*DD/mycold;

        % form correlations
        Colebrook_eq=@(ff)(1/sqrt(ff)+2*log10(2.51/(ReL*sqrt(ff))+roughness/3.71/DD));
        fl=fzero(Colebrook_eq,0.017);

        Colebrook_eq=@(ff)(1/sqrt(ff)+2*log10(2.51/(ReV*sqrt(ff))+roughness/3.71/DD));
        fv=fzero(Colebrook_eq,0.017);

        newh = (k_c*rhoL/2*vL^2+k_h*rhoV/2*vV^2)/((rhoL-rhoV)*9.81-fl*rhoL*vL^2/2/DD - fv*rhoV*vV^2/2/DD);
        newh_vec(ii) = newh;
    end
    figure(2)
    plot(DD_vec,newh_vec,'.-','LineWidth',1.5);
    ax1 = gca;
    ax1.XAxisLocation = 'top';
    hold on; grid on; box on;
end
legend('Smooth Pipe','\epsilon = 5e-4m', '\epsilon = 5e-5m','\epsilon = 2.5e-5m')

```

```

x_points = DD_int(1:end-1);
y_points = newh_vec(ii_saved);
point_text= {'8"','10"','12"','14"','16"','18"','20"','22"','24"'};

for k = 1:length(x_points)
    h2 =plot([x_points(k) x_points(k)], [0 y_points(k)], 'k--','LineWidth',1.3);
    grid on
    text(x_points(k), 0, sprintf('%s', point_text{k}), 'VerticalAlignment','cap', 'HorizontalAlignment','center', 'FontSize',16);
    hAnnotation = get(h2, 'Annotation');
    hLegendEntry = get(hAnnotation, 'LegendInformation');
    set(hLegendEntry, 'IconDisplayStyle', 'off');
end

ax2 = axes('Position', ax1.Position, 'Color', 'none');
ax2.XAxisLocation = 'bottom';
ax2.YAxisLocation = 'right';
ax2.XTick = [];
ax2.YTick = [];
ax2.XLabel.String = 'Corresponding NPS Diameter [in]';
linkaxes([ax1, ax2], 'x');
ax1.XLabel.String = 'Diameter [m]';
xlim([DD_vec(1), 0.6]);
ax1.YLabel.String ='Height and Length [m]';
ax1.FontSize = 16;
ax2.FontSize = 16;
```

A.2 Exercise 1.2

```
    close all
clear all
clc
inches=0.0254;

power=6e6;  %W/m^2
portata0=100;       %kg/s
T0_cold=100; %\ C
T0_hot=140; % C
T0_ave=120; % C

%% ISC
pressure_ISC=70;    %bar
dd_out_ISC=16*inches;
```

```
thickness_ISC=1.031*inches;
dd_in_ISC=dd_out_ISC-2*thickness_ISC;
roughness_ISC=2e-4; % -
hh_ISC=10;
k_90bends_ISC=0.45;
number_bends_ISC=6;
pipe_length_ISC=40;
area_ISC=dd_in_ISC^2*pi/4;

err=1;
toll=1e-8;
i=1;
mm_ISC(1)=portata0;
T_H_ISC(1)=T0_hot;
T_C_ISC(1)=T0_cold;
while err>toll
    viscosity_ISC_cold=XSteam('my_pT',pressure_ISC,T0_cold);
    density_ISC_cold=XSteam('rho_pT',pressure_ISC,T0_cold);
    k_fluid_ISC_cold=XSteam('tc_pT',pressure_ISC,T0_cold);
    cp_ISC_cold=XSteam('Cp_pT',pressure_ISC,T0_cold)*1e3;


    viscosity_ISC_hot=XSteam('my_pT',pressure_ISC,T0_hot);
    density_ISC_hot=XSteam('rho_pT',pressure_ISC,T0_hot);
    k_fluid_ISC_hot=XSteam('tc_pT',pressure_ISC,T0_hot);
    cp_ISC_hot=XSteam('Cp_pT',pressure_ISC,T0_hot)*1e3;

    Re_ISC_cold=(4*portata0)/(dd_in_ISC*pi*viscosity_ISC_cold);
    Re_ISC_hot=(4*portata0)/(dd_in_ISC*pi*viscosity_ISC_hot);

    Pr_ISC_cold=(cp_ISC_cold*viscosity_ISC_cold)/k_fluid_ISC_cold;
    Pr_ISC_hot=(cp_ISC_hot*viscosity_ISC_hot)/k_fluid_ISC_hot;
   %% HX1
    dd_out_HX1=19.05e-3;
    thickness_HX1=1.24e-3;
    dd_in_HX1=dd_out_HX1-2*thickness_HX1;
    nuber_tubes_HX1=897;
    pitch_HX1=28.5e-3;
    dd_shell_HX1=1.5;
    number_baffle_HX1=2;
    baffle_spacing_HX1=1.6;
    tube_length_HX1=9.314;
    k_tube_HX1=15;  %W/m/K
    headers_area_HX1=0.883;
    heattransferarea_HX1=500;
    roughness_HX1=1e-4;
    correctionfactor_HX1=0.7;


```

```
%% HX2
dd_out_HX2=25.4e-3;
thickness_HX2=1.24e-3;
dd_in_HX2=dd_out_HX2-2*thickness_HX2;
nuber_tubes_HX2=770;
tube_length_HX2=7;
dd_manifold=16*inches;
k_tube_HX2=15;  %W/m/K
heattransferarea_HX2=430;
roughness_HX2=1e-4; %-
area_HX2=dd_in_HX2^2*pi/4;


viscosity_HX=XSteam('my_pT',pressure_ISC,T0_ave);
density_HX=XSteam('rho_pT',pressure_ISC,T0_ave);
k_fluid_HX=XSteam('tc_pT',pressure_ISC,T0_ave);
cp_HX=XSteam('Cp_pT',pressure_ISC,T0_ave)*1e3;

Re_HX2=(4*portata0/nuber_tubes_HX2)/(dd_in_HX2*pi*viscosity_HX);
%
Pr_HX2=(cp_HX*viscosity_HX)/k_fluid_HX;

Nu_HX_tube=0.023*Re_HX2^0.8*Pr_HX2^0.3; %(ff_HX/8)*((Re_HX-1000)*Pr_HX)/(1+12.7*(ff_HX/8)^0.5*(Pr_HX^2/3-1));
hh_int_HX2=k_fluid_HX*Nu_HX_tube/dd_in_HX2;


%% RESISTENZE TERMICHE
total_surface_HX2=dd_out_HX2*pi*tube_length_HX2*nuber_tubes_HX2;

delta_t_sat_HX2=(power/(2.257*total_surface_HX2))^(1/3.86);

hh_out_HX2=power/(delta_t_sat_HX2*total_surface_HX2);

R_int_HX2=dd_out_HX2/(hh_int_HX2*dd_in_HX2);
R_out_HX2=1/(hh_out_HX2);
R_cond_HX2=log(dd_out_HX2/dd_in_HX2)*dd_out_HX2/(k_tube_HX2*2);

U_HX2=1/(R_cond_HX2+R_int_HX2+R_out_HX2);

alpha_HX2=(U_HX2*total_surface_HX2)/(cp_HX*portata0);

pt1=100/(1-exp(alpha_HX2));
pt2=((power/(cp_HX*portata0))+100)*exp(alpha_HX2);
T_in_HX2=pt1-(pt2/(1-exp(alpha_HX2)));
T_out_HX2=T_in_HX2-power/(cp_HX*portata0);
```

```matlab

T_ave=(T_in_HX2+T_out_HX2)/2;

viscosity_HX=XSteam('my_pT',pressure_ISC,T_ave);
density_HX=XSteam('rho_pT',pressure_ISC,T_ave);
k_fluid_HX=XSteam('tc_pT',pressure_ISC,T_ave);
cp_HX=XSteam('Cp_pT',pressure_ISC,T_ave)*1e3;

density_ISC_cold=XSteam('rho_pT',pressure_ISC,T_out_HX2);
density_ISC_hot=XSteam('rho_pT',pressure_ISC,T_in_HX2);

viscosity_ISC_cold=XSteam('my_pT',pressure_ISC,T_out_HX2);
viscosity_ISC_hot=XSteam('my_pT',pressure_ISC,T_in_HX2);

Re_ISC_hot=(4*portata0)/(pi*dd_in_ISC*viscosity_ISC_hot);
Re_ISC_cold=(4*portata0)/(pi*dd_in_ISC*viscosity_ISC_cold);

Re_HX2=(4*portata0/nuber_tubes_HX2)/(pi*dd_in_HX2*viscosity_HX
);

area_shell_HX1=dd_shell_HX1*(pitch_HX1-dd_out_HX1)*
baffle_spacing_HX1/pitch_HX1;
dd_eq_shell=(2*sqrt(3)*pitch_HX1^2)/(pi*dd_out_HX1)-dd_out_HX1
;

v_shell_HX1=portata0/(density_HX*area_shell_HX1);
Re_shell_HX1=density_HX*v_shell_HX1*dd_eq_shell/viscosity_HX;


%% colebrook
eq=@(ff)(1/sqrt(ff)+2*log10(2.51/(Re_ISC_hot*sqrt(ff))+
roughness_ISC/3.71));
ff_ISC_hot_colebrook=fzero(eq,0.02);

eq=@(ff)(1/sqrt(ff)+2*log10(2.51/(Re_ISC_cold*sqrt(ff))+
roughness_ISC/3.71));
ff_ISC_cold_colebrook=fzero(eq,0.02);

eq=@(ff)(1/sqrt(ff)+2*log10(2.51/(Re_HX2*sqrt(ff))+
roughness_ISC/3.71));
ff_HX2_colebrook=fzero(eq,0.02);

%% PRESSURE LOSSES
area_shell_HX1=dd_shell_HX1*(pitch_HX1-dd_out_HX1)*
baffle_spacing_HX1/pitch_HX1;

k_shell_HX1=8*0.227*dd_shell_HX1*(number_baffle_HX1+1)/(
Re_shell_HX1^0.193*dd_eq_shell);

```

```
    area_collettore=dd_manifold^2*pi/4;

    k_disloss_ISC_cold=ff_ISC_cold_colebrook*pipe_length_ISC/(dd_in_ISC*2);
    k_disloss_ISC_hot=ff_ISC_hot_colebrook*pipe_length_ISC/(dd_in_ISC*2);

    k_disloss_HX2=ff_HX2_colebrook*tube_length_HX2/dd_in_HX2;

    k_contrazione_hot=0.5*(1-(area_HX2/area_collettore));
    k_enl_hot=(1-(area_ISC/area_collettore))^2;

    k_contrazione_cold=0.5*(1-area_ISC/area_collettore);
    k_enl_cold=(1-(area_HX2/area_collettore))^2;

    k_conloss_ISC_hot=(k_contrazione_hot+k_enl_hot)+number_bends_ISC*k_90bends_ISC/2;
    k_conloss_ISC_cold=(k_contrazione_cold+k_enl_cold)+number_bends_ISC*k_90bends_ISC/2;

    termine_bounty=(density_ISC_cold-density_ISC_hot)*hh_ISC*9.81;
    termine_cold=(k_disloss_ISC_cold+k_conloss_ISC_cold)*(density_ISC_cold)/2;
    termine_hot=(k_disloss_ISC_hot+k_conloss_ISC_hot)*density_ISC_hot/2;
    termine_HX2=(k_disloss_HX2)*density_HX/2;
    termine_HX1=(k_shell_HX1)*density_HX/2;

    equazione=@(portata)(termine_hot)*(portata/(density_ISC_hot*area_ISC))^2 ...
          +(termine_cold)*(portata/(density_ISC_cold*area_ISC))^2 ...
          +(termine_HX2)*(portata/(density_HX*area_HX2*nuber_tubes_HX2))^2 ...
          +(termine_HX1)*(portata/(density_HX*area_shell_HX1))^2-termine_bounty;

    portata_new=fzero(equazione,portata0);

    err=abs(portata_new-portata0)/portata0;

    portata0=portata_new*0.7+portata0*0.3;
    T0_ave=T_ave;
    T0_hot=T_in_HX2;
    T0_cold=T_out_HX2;


    i=i+1;
    mm_ISC(i)=portata0;
```

```
    T_H_ISC(i)=T0_hot;
    T_C_ISC(i)=T0_cold;
end

figure(1)
subplot(2,2,1)
plot(1:i, mm_ISC)
hold on
grid on
xlabel('Iteration [-]')
ylabel('Mass flow rate [kg/s]')
title('ISC')

subplot(2,2,3)
plot(1:i, T_C_ISC)
hold on
grid on
plot(1:i, T_H_ISC)
hold on
xlabel('Iteration [-]')
ylabel('Temperature [ C ]')
legend('Cold', 'Hot')
title('ISC')


T0_ave=T_ave;
T0_hot=T_in_HX2;
T0_cold=T_out_HX2;
portata0=portata_new;


figure(2)
subplot(1,2,1)

% hot
ISC_termine_hot_Pa_dis=(k_disloss_ISC_hot*density_ISC_hot/2)*
    portata_new^2/(density_ISC_hot*area_ISC)^2;
ISC_termine_hot_Pa_conc=(k_conloss_ISC_hot*density_ISC_hot/2)*
    portata_new^2/(density_ISC_hot*area_ISC)^2;
% cold
ISC_termine_cold_Pa_dis=(k_disloss_ISC_cold*(density_ISC_cold)/2)*
    portata_new^2/(density_ISC_cold*area_ISC)^2;
ISC_termine_cold_Pa_conc=(k_conloss_ISC_cold*(density_ISC_cold)/2)
    *portata_new^2/(density_ISC_cold*area_ISC)^2;

ISC_termine_HX2_Pa_dis=termine_HX2*portata_new^2/(density_HX*
    area_HX2*nuber_tubes_HX2)^2;
ISC_termine_shell_Pa_conc=(termine_HX1)*(portata_new/(density_HX*
    area_shell_HX1))^2;
```

```
X=categorical({'Hot leg', 'Cold leg', 'HX2', 'Shell', 'Buoyancy'});
X=reordercats(X,{'Hot leg', 'Cold leg', 'HX2', 'Shell', 'Buoyancy'});
Y=[ISC_termine_hot_Pa_dis ISC_termine_hot_Pa_conc; ISC_termine_cold_Pa_dis ISC_termine_cold_Pa_conc; ISC_termine_HX2_Pa_dis 0; 0 ISC_termine_shell_Pa_conc; -termine_bounty 0]/1e3;
b=bar(X,Y, 'stacked',FaceColor='flat');
colors = [0 0.4470 0.7410;  0.8500 0.3250 0.0980];
for k=1:size(Y,2)

    b(k).CData=colors(k,:);
end
hold on
grid on
patch(NaN, NaN, colors(1,:), 'DisplayName', 'Distributed');
patch(NaN, NaN, colors(2,:), 'DisplayName', 'Concentrated');
hold off;


legend('Location', 'best');
ylabel('Pressure [kPa]')
title('Pressure losses in ISC')


%% conservazione energia

cp_HX2=XSteam('Cp_pT',pressure_ISC,T0_ave)*1e3;
energia1=portata0*cp_HX2*(T0_hot-T0_cold);
delta_T_log=(T0_hot-T0_cold)/log((T0_hot-100)/(T0_cold-100));
energia2=U_HX2*total_surface_HX2*delta_T_log;
energia1-energia2;




%% PSC
pressure_PSC=75; %bar
dd_in_PSC=dd_in_ISC;
pipe_length_PSC=16;
roughness_PSC=2e-4;
k_90bend_PSC=k_90bends_ISC;
number_bends_PSC=4;
k_lossvalve_PSC=0.12;
height1_PSC=7;
height2_PSC=3;
total_surface_HX1=dd_out_HX1*pi*tube_length_HX1*nuber_tubes_HX1;
```

```
T0_cold_PSC=170;
T0_hot_PSC=200;
portata0_PSC=100;
T0_ave_PSC=(T0_hot_PSC+T0_cold_PSC)*0.5;
err=1;
toll=1e-8;
i=1;
mm_PSC(1)=portata0_PSC;
T_H_PSC(1)=T0_hot_PSC;
T_C_PSC(1)=T0_cold_PSC;
while err>toll
    viscosity_HX1=XSteam('my_pT',pressure_PSC,T0_ave_PSC);
    density_HX1=XSteam('rho_pT',pressure_PSC,T0_ave_PSC);
    k_fluid_HX1=XSteam('tc_pT',pressure_PSC,T0_ave_PSC);
    cp_HX1=XSteam('Cp_pT',pressure_PSC,T0_ave_PSC)*1e3;

    viscosity_PSC_cold=XSteam('my_pT',pressure_PSC,T0_cold_PSC);
    density_PSC_cold=XSteam('rho_pT',pressure_PSC,T0_cold_PSC);
    k_fluid_PSC_cold=XSteam('tc_pT',pressure_PSC,T0_cold_PSC);
    cp_PSC_cold=XSteam('Cp_pT',pressure_PSC,T0_cold_PSC)*1e3;

    viscosity_PSC_hot=XSteam('my_pT',pressure_PSC,T0_hot_PSC);
    density_PSC_hot=XSteam('rho_pT',pressure_PSC,T0_hot_PSC);
    k_fluid_PSC_hot=XSteam('tc_pT',pressure_PSC,T0_hot_PSC);
    cp_PSC_hot=XSteam('Cp_pT',pressure_PSC,T0_hot_PSC)*1e3;

    Re_PSC_cold=(4*portata0)/(dd_in_PSC*pi*viscosity_PSC_cold);
    Re_PSC_hot=(4*portata0)/(dd_in_PSC*pi*viscosity_PSC_hot);

    Re_HX1=(4*portata0_PSC)/(pi*nuber_tubes_HX1*dd_in_HX1*
   viscosity_HX1);


    Pr_HX1=cp_HX1*viscosity_HX1/k_fluid_HX1;


    f_bad_PSC_hot=-1.8*log10((roughness_PSC/3.7)^1.11+(6.9/
   Re_PSC_hot));
    ff_PSC_hot=f_bad_PSC_hot^-2;

    f_bad_PSC_cold=-1.8*log10((roughness_PSC/3.7)^1.11+(6.9/
   Re_PSC_cold));
    ff_PSC_cold=f_bad_PSC_cold^-2;

    Nu_HX1=0.023*Re_HX1^0.8*Pr_HX1^0.3;
    h_HX1=Nu_HX1*k_fluid_HX1/dd_in_HX1;

```

```
    dd_eq_shell=(2*sqrt(3)*pitch_HX1^2)/(pi*dd_out_HX1)-dd_out_HX1;

    h_out_HX1=0.351*(Re_shell_HX1^0.55)*k_fluid_HX1*Pr_HX1^(1/3)/dd_eq_shell;

    R_int_HX1 = dd_out_HX1/(h_HX1*dd_in_HX1);
    R_out_HX1 = 1/h_out_HX1;
    R_cond_HX1 = dd_out_HX1/2/k_tube_HX1*log(dd_out_HX1/dd_in_HX1);

    U_HX1=1/(R_cond_HX1+R_out_HX1+R_int_HX1);

    alpha_HX1_prova=(U_HX1*total_surface_HX1*0.7)/(cp_HX1*portata0_PSC)+((T_out_HX2-T_in_HX2)*(U_HX1*total_surface_HX1*0.7)/(power));
    alpha_HX1=((power/(cp_HX1*portata0_PSC))+T_out_HX2-T_in_HX2)*(U_HX1*total_surface_HX1*0.7)/power;
    %% fsolve
    funzione = @(Tcold) (((Tcold+(P_th/(mguessPSC*Cp_intHX1))) - Thot_ISC)-(Tcold-Tcold_ISC))/log((((Tcold+P_th/(mguessPSC*Cp_intHX1)) - Thot_ISC))/(Tcold-Tcold_ISC)) - P_th/(U_HXT1*Sup*Ft);
    equazione_T_out_HX1=@(T_out)(-power+(U_HX1*total_surface_HX1*0.7*(power/(portata0_PSC*cp_HX1)-T_in_HX2+T_out_HX2))/log((power/(portata0_PSC*cp_HX1)+T_out-T_in_HX2)/(T_out-T_out_HX2)));


    T_out_HX1=fzero(equazione_T_out_HX1,T0_cold_PSC);
    T_in_HX1=T_out_HX1+power/(cp_HX1*portata0_PSC);


    T0_cold_PSC=T_out_HX1;
    T0_hot_PSC=T_in_HX1;
    T0_ave_PSC=(T0_hot_PSC+T0_cold_PSC)/2;

    viscosity_HX1=XSteam('my_pT',pressure_PSC,T0_ave_PSC);
    density_HX1=XSteam('rho_pT',pressure_PSC,T0_ave_PSC);
    k_fluid_HX1=XSteam('tc_pT',pressure_PSC,T0_ave_PSC);
    cp_HX1=XSteam('Cp_pT',pressure_PSC,T0_ave_PSC)*1e3;

    viscosity_PSC_cold=XSteam('my_pT',pressure_PSC,T0_cold_PSC);
    density_PSC_cold=XSteam('rho_pT',pressure_PSC,T0_cold_PSC);
    k_fluid_PSC_cold=XSteam('tc_pT',pressure_PSC,T0_cold_PSC);
    cp_PSC_cold=XSteam('Cp_pT',pressure_PSC,T0_cold_PSC)*1e3;

    viscosity_PSC_hot=XSteam('my_pT',pressure_PSC,T0_hot_PSC);
    density_PSC_hot=XSteam('rho_pT',pressure_PSC,T0_hot_PSC);
```

```
k_fluid_PSC_hot=XSteam('tc_pT',pressure_PSC,T0_hot_PSC);
cp_PSC_hot=XSteam('Cp_pT',pressure_PSC,T0_hot_PSC)*1e3;

Re_PSC_hot=(4*portata0_PSC)/(pi*dd_in_PSC*viscosity_PSC_hot);
Re_PSC_cold=(4*portata0_PSC)/(pi*dd_in_PSC*viscosity_PSC_cold)
;
Re_HX1=(4*portata0_PSC/nuber_tubes_HX1)/(pi*dd_in_HX1*
viscosity_HX1);

f_bad_cold=-1.8*log10((roughness_PSC/3.7)^1.11+(6.9/
Re_PSC_cold));
f_bad_hot=-1.8*log10((roughness_PSC/3.7)^1.11+(6.9/Re_PSC_hot)
);
f_bad_HX1=-1.8*log10((roughness_HX1/3.7)^1.11+(6.9/Re_HX1));

ff_PSC_cold=f_bad_cold^-2;
ff_PSC_hot=f_bad_hot^-2;
ff_HX1=f_bad_HX1^-2;

%% COLEBROK
eq=@(ff)(1/sqrt(ff)+2*log10(2.51/(Re_HX1*sqrt(ff))+
roughness_HX1/3.71));
ff_HX1_colebrook=fzero(eq,0.02);

eq=@(ff)(1/sqrt(ff)+2*log10(2.51/(Re_PSC_hot*sqrt(ff))+
roughness_PSC/3.71));
ff_PSC_hot_colebrook=fzero(eq,0.02);

eq=@(ff)(1/sqrt(ff)+2*log10(2.51/(Re_PSC_cold*sqrt(ff))+
roughness_PSC/3.71));
ff_PSC_cold_colebrook=fzero(eq,0.02);

%%

k_disloss_PSC_hot=ff_PSC_hot_colebrook*(pipe_length_PSC/2)*
density_PSC_hot/(2*dd_in_PSC);
k_disloss_PSC_cold=ff_PSC_cold_colebrook*(pipe_length_PSC/2)*
density_PSC_cold/(2*dd_in_PSC);
k_disloss_HX1=ff_HX1_colebrook*tube_length_HX1*density_HX1/(
dd_in_HX1*2);

area_PSC=dd_in_PSC^2*pi/4;
area_HX1=dd_in_HX1^2*pi/4;
area_headers=0.883/2;

k_enl_PSC_hot=(1-area_PSC/area_headers)^2;
k_contrazione_PSC_hot=0.5*(1-area_HX1/area_headers);


```

```
    k_enl_PSC_cold=(1-area_HX1/area_headers)^2;
    k_contrazione_PSC_cold=0.5*(1-area_PSC/area_headers);


    k_conloss_PSC_hot=(k_90bend_PSC*number_bends_PSC*0.5+k_contrazione_PSC_hot+k_enl_PSC_hot)*density_PSC_hot/2;
    k_conloss_PSC_cold=(k_lossvalve_PSC+k_90bend_PSC*number_bends_PSC*0.5+k_contrazione_PSC_cold+k_enl_PSC_cold)*density_PSC_cold/2;
    k_conloss_HX1=2*k_90bend_PSC*density_HX1/2;

    termine_hot=k_conloss_PSC_hot+k_disloss_PSC_hot;
    termine_cold=k_conloss_PSC_cold+k_disloss_PSC_cold;
    termine_HX1=k_disloss_HX1+k_conloss_HX1;

    k_loss_core_asterisco=2*1.2e5*density_HX1/(3200^2);
    termine_core=k_loss_core_asterisco/(2*density_HX1);


    termine_bounty_PSC_1=-9.81*height2_PSC/2*(density_PSC_hot+density_PSC_cold) + 9.81*height1_PSC*(density_PSC_cold-density_PSC_hot) + 9.81*density_PSC_cold*height2_PSC;
    equazione=@(portata)(termine_hot)*(portata/(density_PSC_hot*area_PSC))^2 ...
          +(termine_cold)*(portata/(density_PSC_cold*area_PSC))^2 ...
          +(termine_HX1)*(portata/(density_HX1*area_HX1*nuber_tubes_HX1))^2 ...
          +termine_core*(portata)^2-termine_bounty_PSC_1;

    portata_new_PSC=fzero(equazione,portata0_PSC);

    err=abs(portata_new_PSC-portata0_PSC)/portata_new_PSC;
    portata0_PSC=portata_new_PSC;
    i=i+1;
    mm_PSC(i)=portata0_PSC;
    T_C_PSC(i)=T0_cold_PSC;
    T_H_PSC(i)=T0_hot_PSC;
end

T_ave=(T_in_HX1+T_out_HX1)*0.5;
cp_core=XSteam('Cp_pT',pressure_PSC,T_ave)*1e3;
energia3=portata0_PSC*(T_in_HX1-T_out_HX1)*cp_core;
delta_T_log= ( (T_in_HX1-T_in_HX2) - (T_out_HX1 - T_out_HX2) )/log((T_in_HX1-T_in_HX2)/(T_out_HX1-T_out_HX2));
energia4=U_HX1*total_surface_HX1*delta_T_log*0.7;
%energia3-energia4


```

```
figure(1)
subplot(2,2,2)
plot(1:i, mm_PSC)
hold on
grid on
xlabel('Iteration [-]')
ylabel('Mass flow rate [kg/s]')
title('PSC')

subplot(2,2,4)
plot(1:i, T_C_PSC)
hold on
plot(1:i, T_H_PSC)
grid on
xlabel('Iteration [-]')
ylabel('Temperature [ C ]')
legend('Cold', 'Hot')
title('PSC')


figure(3)
X=categorical({'ISC', 'PSC'});
X=reordercats(X,{'ISC', 'PSC'});
bar(X,[energia1 energia2; energia3 energia4]/1e6)
hold on
grid on
ylabel('Power [MW]')
title('Energy conservation')

%% plotto i contributi delle cadute di pressione
% distribuzione in hot e cold

figure(2)
subplot(1,2,2)
% hot
termine_hot_Pa_dis=k_disloss_PSC_hot*portata0_PSC^2/(
    density_PSC_hot*area_PSC)^2;
termine_hot_Pa_conc=k_conloss_PSC_hot*portata0_PSC^2/(
    density_PSC_hot*area_PSC)^2;
%cold
termine_cold_Pa_dis=k_disloss_PSC_cold*portata0_PSC^2/(
    density_PSC_cold*area_PSC)^2;
termine_cold_Pa_conc=k_conloss_PSC_cold*portata0_PSC^2/(
    density_PSC_cold*area_PSC)^2;

termine_HX1_Pa_dis=k_disloss_HX1*portata0_PSC^2/(density_HX1*
    area_HX1*nuber_tubes_HX1)^2;
```

```
termine_HX1_Pa_conc=k_conloss_HX1*portata0_PSC^2/(density_HX1*
    area_HX1*nuber_tubes_HX1)^2;

termine_core_Pa_dis=termine_core*(portata0_PSC)^2;

X=categorical({'Hot leg', 'Cold leg', 'HX1', 'Core', 'Buoyancy'});
X=reordercats(X,{'Hot leg', 'Cold leg', 'HX1', 'Core', 'Buoyancy'
    });
Y=[termine_hot_Pa_dis termine_hot_Pa_conc; termine_cold_Pa_dis
    termine_cold_Pa_conc; termine_HX1_Pa_dis termine_HX1_Pa_conc;
    termine_core_Pa_dis 0; -termine_bounty_PSC_1 0]/1e3;

b=bar(X,Y, 'stacked',FaceColor='flat');
colors = [0 0.4470 0.7410;  0.8500 0.3250 0.0980];
for k=1:size(Y,2)

     b(k).CData=colors(k,:);
end
hold on
grid on
patch(NaN, NaN, colors(1,:), 'DisplayName', 'Distributed');
patch(NaN, NaN, colors(2,:), 'DisplayName', 'Concentrated');
hold off;


legend('Location', 'best');
ylabel('Pressure [kPa]')
title('Pressure losses in PSC')
```