



**Politecnico
di Torino**

Master 's Degree in Energy and Nuclear Engineering
Renewable Energy System
A.Y. 2023/2024
Graduation Session October 2024

A new code for geothermal potential assessment: power production in medium-high geothermal systems

Supervisor:
Prof. Federico Vagnon

Candidate:
Matteo Cornetto matr. 306174

Co-supervisors:
Dott.ssa PhD. Martina Gizzi
Dott. PhD. Gianluca Gola

This page has been left blank intentionally

Abstract

Geothermal energy is a reliable and effective resource for power generation due to its continuous operation, particularly in medium-to-high geothermal systems. This thesis proposes a new code for assessing geothermal potential, which is essential for identifying suitable locations and ensuring the profitability of geothermal plants.

The proposed code combines temporal variations in underground conditions with the plant's production metrics. It consists of a three-step process: first, a subsurface evaluation, followed by a thermodynamic simulation of a specific cycle within the power plant and then an economic analysis, in terms of the Levelized Cost of Energy (LCOE) and Net Present Value (NPV), to assess the feasibility of the plant's location.

Starting from a comprehensive review of methods and codes for evaluating geothermal potential, then it focuses on the models implemented to compute the three components of the proposed code. Validation of the implemented thermodynamic cycles is performed by comparing the power output calculated by the code with that of existing geothermal power plants. The results show that only a few power outputs exceed a 10% error margin, indicating strong performance. Finally, the code is applied to a real case study in the Cesano-Sabatini area. The geothermal potential of this area is analysed based on the chosen depth and the plant's typology.

Acknowledgements

First of all, I would like to thank my supervisor Prof. Federico Vagnon, to make me in touch with my external supervisor Doc. Gianluca Gola, and therefore giving me the opportunity to write this thesis on a subject I have always been interested in, for remembering my request about thesis argument after months, and for its availability.

I would like to extent my sincere gratefulness to my external supervisor Doc. Gianluca Gola, to guide me during these months, for letting me take part in this project, and to always ask my opinion and my point of view.

Table of Contents

Abstract	
List of figures	VI
List of tables	VIII
1. Introduction	1
2. A brief introduction to geothermal energy.....	3
3. Review of Geothermal Potential Evaluation	5
3.1. Single point methods.....	5
3.1.1. Surface heat flux	5
3.1.2. Planar fracture	6
3.1.3. Total well flow.....	7
3.1.4. Magmatic heat budget.....	7
3.1.5. Power density	7
3.1.6. Volumetric method.....	8
3.1.7. Mass-in-place	11
3.2. History method	12
3.2.1. Lumped parameter	12
3.2.2. Decline analysis	13
3.2.3. Numerical reservoir simulation	14
3.3. Probabilistic resource assessment method.....	15
4. Methodology.....	17
4.1. Geological model.....	17
4.1.1. Mathematical model.....	18
4.1.2. Thermal properties of the rocks.....	20
4.1.3. Wellbore heat losses.....	22
4.2. Thermodynamic cycles of plants	24
4.2.1. Dry steam power plant	25
4.2.2. Flash power plant.....	27
4.2.3. Binary power plant.....	28
4.3. Model of the components	30
4.3.1. Filter unit.....	30
4.3.2. Turbine	31
4.3.3. Direct contact condenser	32
4.3.4. Compressor.....	33
4.3.5. Pumps.....	34
4.3.6. Cooling tower	36
4.3.7. Heat exchanger	36
4.3.8. Separator	43

4.4.	Performance evaluation parameters	43
4.5.	Inputs required from the user.....	45
4.6.	Validation of the models	50
4.6.1.	Dry steam model.....	51
4.6.2.	Single flash model	51
4.6.3.	Binary model	52
4.7.	Economic model	54
4.7.1.	Levelized cost of energy	54
4.7.2.	Net Present Value.....	55
4.7.3.	Costs evaluation.....	55
5.	Application of the code: a case study of the Cesano-Sabatini area.....	61
5.1.	Geological setting of Central-Southern Italy	61
5.1.1.	Geodynamic and paleogeographic backgrounds of Central-Southern Italy	61
5.1.2.	Geological model and temperature distribution of Lazio Region.....	64
5.1.3.	The Cesano-Sabatini case study	66
5.2.	Temperature and pressure distribution for code's simulation.....	68
5.3.	Results	69
6.	Conclusion and future developments	79
	Bibliography	81
	Appendix	86
I.	Askinput_binary.....	86
II.	Askinput_dry_steam	91
III.	Askinput_dry_steam_top_spillation	96
IV.	Askinput_single_flash.....	101
V.	Binary.....	106
VI.	Dry_steam_pure_electricity.....	112
VII.	Dry_steam_top_spillation	116
VIII.	Single_flash.....	120
IX.	Economic_model_binary.....	125
X.	Economic_model_dry_steam	128
XI.	Economic_model_flash.....	131
XII.	Economic_model_dry_steam_top_spillation	134
XIII.	Ramey model.....	138
XIV.	Main.....	139

List of figures

Figure 1. Schematic diagram illustrating the planar fracture model (Muffler et al. 1978,[18])	6
Figure 2. Plot of temperature and power density of 66 geothermal field (Wilmarth & Stimac 2015,[22])	8
Figure 3. Thermal energy evaluated with three different volumetric heat stored methods of four geothermal fields (Ciriaco et al. 2020, [15]).....	11
Figure 4. Lumped-parameter method, reservoir as a closed tank (Ciriaco et al. 2020,[15]).....	12
Figure 5. Water level variation in time considering different tank models (Gaoxuan et al. 2010, [31]).....	13
Figure 6. Decline analysis of Palinipinon-2 area (Orizonte et al. 2005,[32]).....	14
Figure 7. Spatial discretization of the reservoir (left) and temperature distribution at different depths (right). (Zhang et al. 2016, [37]).....	15
Figure 8. Porosity probability distribution: a) Uniform distribution, b) Triangular distribution, c) log-normal distribution (Ciriaco et al. 2020, [15]).....	15
Figure 9. Probability distribution of the plant's power output (Ciriaco et al. 2020, [15]).....	16
Figure 10. The conceptual model of a magmatic, hydrothermal geothermal system	17
Figure 11. Setting of the numerical model: the 3D geometry (A) including the cap-rock, the reservoir and the basement domains. The numerical mesh employed to solve the thermal model	20
Figure 12. Compaction exponential curves by lithology derived from porosity measurements on core samples coming from different depth intervals	21
Figure 13. Laboratory thermal conductivity measurements as function of temperature from ambient to 1000 °C.	21
Figure 14. Dry steam power plant for electricity production reference schematic. Author's own elaboration based on the scheme presented by DiPippo (1980)[51]	26
Figure 15. Dry steam power plant for cogeneration reference schematic. Author's own elaboration based on the scheme presented by DiPippo (1980)[51].....	26
Figure 16. Thermodynamic cycle for dry steam power plant on the left (Mulyana et al. 2016,[54]). Author's own elaboration using MATLAB on the right	27
Figure 17. Flash power plant reference schematic. Author's own elaboration based on the scheme presented by DiPippo (1980)[51]	27
Figure 18. The thermodynamic cycle of a single flash power plant on the left (Andrès et al. 2011,[56]). Author's own elaboration using MATLAB on the right.....	28
Figure 19. Binary power plant reference schematic. Author's own elaboration based on the scheme presented by DiPippo (1980)[51]	29
Figure 20. Binary thermodynamic cycle: theoretical on the left (Kanoglu et al. [60]). Author's own elaboration using MATLAB on the right.....	29
Figure 21. Pressure drops across the cyclone separator. Author's own elaboration	31
Figure 22. Effectiveness as a function of NTU for counter-current heat exchanger (Ezci (2017),[72]).....	37
Figure 23. Conditional statement to choose the effectiveness. Author's own elaboration using MATLAB....	38
Figure 24. Slope variation of the geofluid temperature profile inside the heat exchanger. Author's own elaboration using MATLAB	39
Figure 25. Crossover temperature in the heat exchanger at year 30 with wrongly evaluated organic fluid's mass flow rate. Author's own elaboration using MATLAB	40
Figure 26. Example of temperature profile inside the condenser. Author's own elaboration using MATLAB.....	42
Figure 27. An example of a While cycle to set the value within the specific range. Author's own elaboration using MATLAB	45
Figure 28. Code to find pressure and temperature in function of the given enthalpy. Author's own elaboration using MATLAB	50
Figure 29. Enthalpies found by the search code respect the one given as input. Author's own elaboration using Excel	51
Figure 30. Generated power of dry steam thermodynamic cycle versus the running capacity of real power plants. Author's own elaboration using Excel	51

Figure 31. Generated power of single flash thermodynamic cycle versus the running capacity of real power plants. Author's own elaboration using Excel	52
Figure 32. Trend of the saturated-vapor pressure in function of the temperature. Author's own elaboration using Excel	53
Figure 33. Generated power of binary thermodynamic cycle versus the running capacity of real power plants. Author's own elaboration using Excel	53
Figure 34. Plate reconstruction for the western Mediterranean area (after Carmignani et al. (2004)[111]). ...	62
Figure 35. Borehole locations (black crosses) from BNDG (Trumpy and Manzella, 2017) in Southern Italy. The volcanic areas and the main structural elements of Italian peninsula are displayed.	63
Figure 36. Topographic map of Lazio Region.....	65
Figure 37. Depths (km below sea level) of the top (blue) and bottom (red) surfaces describing the geometry of the geothermal reservoir.	65
Figure 38. Temperature distribution at the top of the geothermal reservoir.	66
Figure 39. Temperature profiles (a) and position (b) of the boreholes.....	67
Figure 40. Production temperature and pressure during the plant's lifetime. Author's own elaboration.....	68
Figure 41. Temperature distribution at 2 and 3 kilometers depth. Author's own elaboration	69
Figure 42. Pressure distribution at 2 and 3 kilometers depth. Author's own elaboration	69
Figure 43. Geothermal potential of dry steam power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 . Author's own elaboration.....	70
Figure 44. Levelized Cost of Energy of dry steam power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 . Author's own elaboration	71
Figure 45. Net Present Value at end of plant's lifetime of dry steam power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 . Author's own elaboration	72
Figure 46. Geothermal potential of single flash power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 . Author's own elaboration.....	73
Figure 47. Levelized Cost of Energy of single flash power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 . Author's own elaboration.....	74
Figure 48. Net Present Value of single flash power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 . Author's own elaboration	75
Figure 49. Geothermal potential of binary power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 . Author's own elaboration.....	76
Figure 50. Levelized Cost of Energy of binary power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 . Author's own elaboration.....	77
Figure 51. Net Present Value of binary power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 . Author's own elaboration.....	78

List of tables

Table 1. Geothermal Energy Generation by region	4
Table 2. Geothermal Energy Generation by country	4
Table 3. Pump formula error respect Aspen Plus for water (k_1 and k_2 are coefficients). Author's own elaboration using the results from Aspen Plus and MATLAB.....	35
Table 4. Pump formula error respect Aspen plus for isobutane (k_1 and k_2 are coefficients). Author's own elaboration using the results from Aspen Plus and MATLAB.....	35
Table 5. Input variables asked to the user in the dry steam plant. Author's own elaboration.	46
Table 6. Input variables asked to the user in dry steam plant with cogeneration. Author's own elaboration...	47
Table 7. Input variables asked to the user in single flash plant. Author's own elaboration.	48
Table 8. input Variable asked to the user in binary plant. Author's own elaboration.....	49
Table 9. Production Price Index (PPI) and Chemical Engineering Plant Cost Index in years 2002/2003 and 2023. Author's own elaboration. Sources: U.S. Bureau of Labor Statistics [82], and Cost Indices [83] .	56
Table 10. Dry steam power plant results	72
Table 11. Single Flash power plant results	75
Table 12. Binary power plant results	78

This page has been left blank intentionally

1. Introduction

The transition towards Renewable Energy Sources (RES) to replace fossil fuels is one of the main objectives of global political agendas in response to the growing energy demand. Since the Second Industrial Revolution, production, transportation, and domestic activities have become more and more dependent on fossil fuels. While natural gas, oil and coal have the advantage of producing a significant amount of thermal energy, they also contribute to air pollution, leading to rising global temperatures, and cause significant environmental impacts from extraction activities at various sites[1].

Among RES, geothermal energy stands out due to its continuous operation. It is defined as the heat from the Earth, and it is a statute-recognized renewable resource [2].

Geothermal energy has a considerable growth potential, since the amount of heat within the first 10'000 meters of the earth's surface is estimated to contain 50'000 times more energy than all oil and gas resources worldwide [3]. However, the adoption of geothermal energy is progressing more slowly compared to other renewable energy sources, such as solar photovoltaic (PV) and wind energy. Since 2010, solar PV and wind energy have experienced significant annual growth rate. In 2022, electricity generation from solar energy rose to 1294 TWh, 25.6% increase from the previous year, while wind energy production grew by 14.0% reaching 2098 TWh [4].

In contrast, geothermal power generation was equal to 94 TWh in 2020 and increase to 97 TWh in 2022, reflecting a modest growth of 3 TWh in 2 years [4]. To reach the Net Zero Emission target by 2050 and avoid the emission of 800 megatons of CO₂, geothermal power generation needs to reach 330 TWh by 2030 and up to 1'400 TWh by 2050[5],[6]. From these data is possible to understand how low and far is the growth rate to reach the target.

Despite its slow growth rate, geothermal energy remains fundamental because it provides a base-load generation, unaffected by weather conditions and seasonal variation, and has a high-capacity factor, also up to 95% in modern plants. Moreover, increased deployment of this energy does not require additional load-balancing for the electricity system, and it is compatible with both centralised and distributed generation. It can also produce both electricity and heat in combined heat and power (CHP) systems [6].

The feasibility of installing geothermal power plants that utilise medium or high-temperature resources depends on site-specific conditions. Therefore, evaluating geothermal potential is essential for selecting suitable locations and ensuring the plant's profitability. There are various methods to calculate the geothermal potential, with the simplest being the assessment of the yearly amount of recoverable geothermal energy. More specific definitions limit the potential for the recoverable energy produced by existing facilities during a year of continuous operation. Alternatively, geothermal potential can be defined as the thermal power possible to produce from a proven resource in a given region, assuming the necessary wells and surface facilities are in place, or as the geothermal output consumed by an operation's proven geothermal resources [7].

The goal of this thesis is to develop a new code to assess geothermal potential through a thermodynamic and an economic analysis. This process involves an initial subsurface evaluation, followed by a thermodynamic simulation of a specific cycle within the power plant, and concludes with an economic analysis to determine the feasibility of the plant's location.

Starting with a literature review and examining two existing codes, the best features were integrated into the new code for both the underground and the surface plant components. The code provides users

with a map of the geothermal potential and Levelized Cost of Energy (LCOE) at the selected depth. After the user select the plant location, the code performs an economic evaluation, including the Net Present Value (NPV), the possible payback time (PBT) and the revenue for the plant's lifetime.

The following chapters introduce geothermal energy, outline different methods for evaluating geothermal potential with a focus on the Volumetric heat stored method, present the methodology guiding the new codes thermo-economic analysis, and validate the code to assess the geothermal potential of the Cesano-Sabatini area, located in the centre of Italy, with results discussed in the subsequent chapter. The thesis concludes with reflections on this work and suggestions for future improvements.

2. A brief introduction to geothermal energy

Geothermal energy originates from heat within the Earth's, primarily from two sources: the primordial heat, which is the heat generated during the Earth's formation, and the heat generated from the decay of long-lived radioactive isotopes. Among all the isotopes, the key ones contributing to this heat, due to their half-lives comparable to Earth's age (4.5 billion years) and relative abundant are: ^{40}K , ^{232}Th , ^{235}U and ^{238}U [8].

The heat flows from the interior of the Earth to the surface: the two forms of heat transfer making it possible are conduction and convection. Conduction involves the transfer of kinetic energy between the molecules, where moving molecules cause their neighbouring to vibrate faster and thus transferring heat. This is the main heat transfer process in solids. Convection, on the other hand, consists of the movement of hot fluid, in liquid or gas state, from one place to another. Because it involves the physical movement of material, this process is significantly more efficient than conduction [8].

The amount of heat flow reaching the surface has been estimated by Pollack et al. (1993) [9], considering 24'774 observations from 20'201 sites, which allows to cover the 62% of the Earth's surface. It results in a continental mean heat flow of $65 \left[\frac{\text{mW}}{\text{m}^2} \right]$, and an oceanic of $101 \left[\frac{\text{mW}}{\text{m}^2} \right]$, which means a global mean heat flow of $87 \left[\frac{\text{mW}}{\text{m}^2} \right]$, and so a global heat loss of $44.2 \times 10^{12} [W]$.

Heat and temperature are closely related, but the latter is more indicative to understand which type of technology is better to use in a specific site. The temperature inside the Earth change with the increase of depth, and it indicates the free energy available at a location. The temperature gradient describes the direction of the heat flow, from a region at higher temperatures to a region at lower temperatures. It is defined as the temperature difference of two points over their distance:

$$\frac{T_1 - T_2}{\Delta z} = \frac{\partial T}{\partial z} \left[\frac{^\circ\text{C}}{\text{km}} \right]$$

Conventionally, the positive sign is in the direction of increasing temperature[10].

The temperature is usually divided into three groups: high (greater than 150°C), medium ($90\text{-}150^\circ\text{C}$) and low (less than 90°C). Electricity generation is most viable at high temperatures, where a minimum resource of about $150\text{-}180^\circ\text{C}$ is required[11]. Lower temperature resources can be used for direct heating applications, such as space and district heating, aquaculture, horticulture, water heating and industrial processes[12].

High-temperature resources are typically located near volcanoes, in tectonically and volcanically active areas, such as the Pacific Ring of Fire, the mid-Atlantic ridge, parts of Europe and the East African Rift. Depending on the local temperature gradient, these resources can be found at different depths, from few hundred metres to several kilometres [11].

Low- and medium-temperature resources are more widely distributed and can often be found along faults and fractures in tectonically active areas and in sedimentary basins, which usually require deeper drilling to reach a sufficient temperature[11].

The worldwide scenario sees Asia leading the share of geothermal energy generation, followed by North America and Europe, as shown in Table 1, based on data from the International Renewable Energy Agency (2024) [13].

Regions	Geothermal Energy Generation [GWh]
Asia	30'227
North America	23'678
Europe	12'343
Eurasia	11'567
Oceania	8'640
Africa	5'353
C.America & Carabian	4'602
South America	465

Table 1. Geothermal Energy Generation by region

From the same study, the classification of the more productive countries in the geothermal field is also reported (Table 2)[13].

Countries	Geothermal Energy Generation [GWh]
United States	19'142
Indonesia	16'667
Turkey	11'119
Philippine	10'425
New Zeland	8'544
Iceland	5'916
Italy	5'836
Kenya	5'324
Mexico	4'536

Table 2. Geothermal Energy Generation by country

Some barriers to the spread of geothermal energy use are the high initial capital cost and the resource deployment risk, mainly due to the drilling cost, which increases with depth and usually is significant, and the possible failure in finding suitable productivity and accessibility of the reservoir. Other barriers include perceived environmental issues, limited public awareness about geothermal energy's benefits, and insufficient incentives schemes [12].

For what concerns the emissions, geothermal power plants emit very negligible levels of nitrogen oxide (NO_x) and particulate matter (PM). Sulphur dioxide (SO₂) is produced in very low quantities, in the range of $0 \div 0.16 \left[\frac{kg}{MWh} \right]$. Carbon dioxide (CO₂) emissions range from $0 \div 40.28 \left[\frac{kg}{MWh} \right]$, far lower than the $993.82 \left[\frac{kg}{MWh} \right]$ emitted by coal power plants, as reported by By et al. (2005)[2]. There are other chemical species as hydrogen sulphide (H₂S), which is converted at 99.9% in sulphur, and rarely mercury (Hg), which are reduced by 90%.

Some additional environmental issues are:

- Noise pollution;
- Water consumption in cooling systems, which is around $18.93 \left[\frac{l}{MWh} \right]$;
- Water quality, since the geothermal fluid could contaminate the groundwater system, but the brine is usually injected back into the reservoir in wells with thick casing, which also helps to increase the geothermal reservoir resilience;
- Land use, in terms of square meters required by the plant, but also for the event which could occur, as induced seismicity caused by production and injection operation which could result in microearthquakes, and subsidence, that is the downward sinking of land due to the pressure reduction inside the reservoir;
- Impact on vegetation and wildlife [2].

3. Review of Geothermal Potential Evaluation

The evaluation of geothermal reservoir potential is generally complicated and uncertain due to the low availability of data. However, it is important because these reservoirs are valuable energy resources that can contribute to reaching the Net Zero Emission goal by 2050 and shifting the energy use from fossil fuel to RES. The economic viability of harnessing a geothermal reservoir depends on its size in terms of energy. Usually, the energy content of the reservoir is evaluated through its heat capacity, using temperature measurement and information on rock properties[14].

The scientific literature regarding the estimation of geothermal potential spans from shallow geothermal resources to medium and deep geothermal potential. For this thesis, only studies focused on medium or deep geothermal potential estimation were taken into account.

There are several methods, both simple and complex, for geothermal potential assessment; they can generally be divided into two main categories based on the nature of the input data:

1. Single point, or static methods: they are not based on historical data. In this group are included the following methods:
 1. Surface heat flux;
 2. Planar fracture;
 3. Total well flow;
 4. Magmatic heat budget;
 5. Power density;
 6. Volumetric method;
 7. Mass-in-place;
2. History or dynamic methods: based on production history data. In this group are included the following methods:
 1. Lumped-parameter;
 2. Decline analysis;
 3. Numerical reservoir simulation[15].

Among these methods, the simplest and most widely used are the power density and volumetric methods[16].

3.1. Single point methods

3.1.1. Surface heat flux

This method approximates the potential as the theoretical minimum amount of heat that can be withdrawn from a geothermal resource, measuring the heat loss or gain at the ground surface. This is done through observation of from hot springs and geysers, fumaroles and steaming grounds, seepages, mud pools and thermal grounds.

The total amount of natural heat is equal to the sum of the convective heat (q_{si}) and conductive heat (q_c). The first term is the thermal energy estimated from the individual manifestation, while the second one could be evaluated by the following equation:

$$q_c = Ak \frac{dT}{dz}$$

where:

- A is the surface of the hot ground [m^2];
- k is the thermal conductivity of the rock [$\frac{W}{m^\circ C}$];
- $\frac{dT}{dz}$ is the thermal gradient [$\frac{^\circ C}{m}$];

The equation for the evaluation of the total heat is:

$$q_{tot} = \sum_i^n q_{si} + q_c$$

After estimating the total heat flow, the producible energy is obtained by multiplying the results by a recovery factor or conversion efficiency to give power production. However, this method is subject to large uncertainties and tends to underestimate the potential capacity of the geothermal field due to errors, approximations and subjectivity in the data [15],[17].

3.1.2. Planar fracture

This method was introduced by Bodvarsson in 1972, and consists of a planar fracture, or impermeable rock, in contact with flowing water. The heat is transferred by conduction from the rock to the fluid, and it is possible to calculate the heat theoretically extractable per unit area, using heat-conduction theory[18].

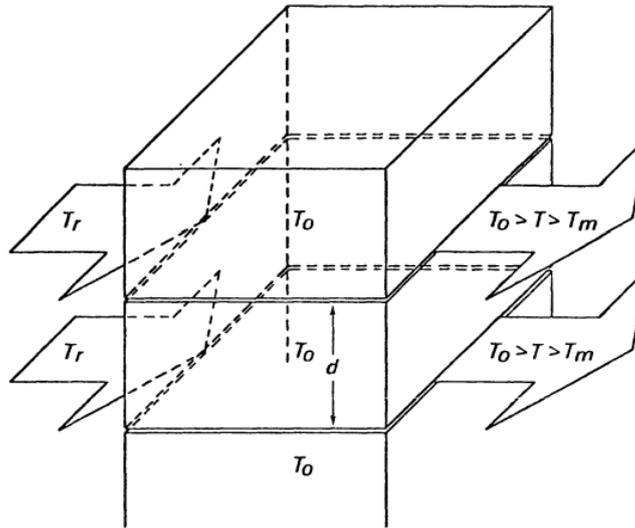


Figure 1. Schematic diagram illustrating the planar fracture model (Muffler et al. 1978,[18])

The geothermal potential can be evaluated knowing the outlet fluid temperature, and it is function of the initial temperature of the rock and the “end temperature ratio”, which is:

$$r = \frac{T_m - T_{rech}}{T_0 - T_{rech}}$$

Where:

- T_m is the minimum rock temperature [$^{\circ}C$];
- T_0 is the initial rock temperature [$^{\circ}C$];
- T_{rech} is the recharge water temperature [$^{\circ}C$];

It is important to point out that T_m is the temperature after a period of production t_0 .

This method can be extended also to multiple fractures, the only constraint is that the distance between the fractures is sufficient to not allowing thermal interaction. The minimum distance between is evaluated as:

$$\frac{d}{2} = 3\sqrt{\alpha t_0}$$

Where:

- α is the thermal diffusivity [$\frac{m^2}{s}$];
- t_0 is the production period [s];

The planar fracture method could be easily applied to fractures with simple and predictable geometry, especially to basaltic sediment, but generally the fracture orientation and distribution are unknown. Due to these limitations, this method is not a useful tool for the evaluation of geothermal potential, and it is not widely used [15],[18].

3.1.3. Total well flow

This method involves using the measured output from wells undergoing intensive discharge tests and summing these results. The total is taken as the field capacity, but it is not completely correct since the total flow of the wells is the current ability to deliver fluid, and it increases as the number of wells increases, without altering the reserves[19].

Moreover, the exploration drilling programs do not exploit the full potential of a site, and so the capacity coming out as a result is limited compared to the real one [15].

3.1.4. Magmatic heat budget

This method focuses on the fraction of magma which does not erupt, and lodges in the upper crust as igneous intrusions, which can be used as heat sources for geothermal systems. It gradually heats the surrounding rocks by conduction, which heat up by convection the fluid flowing in fractures and faults. Therefore, it is possible to estimate the geothermal potential of a region, or restricted areas, knowing the number, position and size of this igneous intrusion, and an analysis of the cooling history, as reported by Muffler et al.[17].

Sanyal et al. (2002) [19] states that knowing the temperature distribution around the magmatic body, it is possible to evaluate the energy reserve using the equation introduced by Brook et al. (1978)[20]:

$$E = \frac{dc_v(T - T_f)R_f\eta_{conv}}{LF}$$

Where:

- E is the power density of the reserves at a given distance from the centre of the caldera $\left[\frac{MW_e}{km^2}\right]$;
- d is the depth where the energy reserves are to be estimated $[m]$;
- c_v is the volumetric specific heat of the reservoir $\left[\frac{J}{kgK}\right]$;
- T is the calculated average temperature between the ground surface and depth at a given distance from the centre of the caldera $[K]$;
- T_f is the average annual ambient temperature $[K]$;
- F is the capacity factor of the plant $[-]$;
- R_f is the recovery factor (defined as the fraction of thermal energy in-place) $[-]$;
- η_{conv} is the conversion efficiency $[-]$;
- L is the lifetime of the power plant $[s]$.

The limitation of this method is in the possibility of using it only for magmatic regions or areas, as the name suggests [14].

3.1.5. Power density

This method wants to correlate the power production per unit area with the hot water temperature of the geothermal reservoir, using a simple empirical formula, which is based on a study from James (1984)[21]:

$$\frac{MW_e}{km^2} = \left(\frac{T}{86.9}\right)^2$$

Compared to other methods, it requires fewer assumptions for estimating the geothermal potential, and its reliability and usability are as good as the data used to produce the empirical correlation. However, this method is not suitable in projects at exploration phases and is not appropriate when there are only

a few exploration wells drilled. It remains a good item to provide a rough estimation of the resource capacity [14].

Wilmarth and Stimac (2015)[22] reported the evaluation of the power density for 66 geothermal fields above 10 MW_e and more than 5 years of production history. They find that the power density is strongly correlated with the geological setting rather than the temperature (Figure 2).

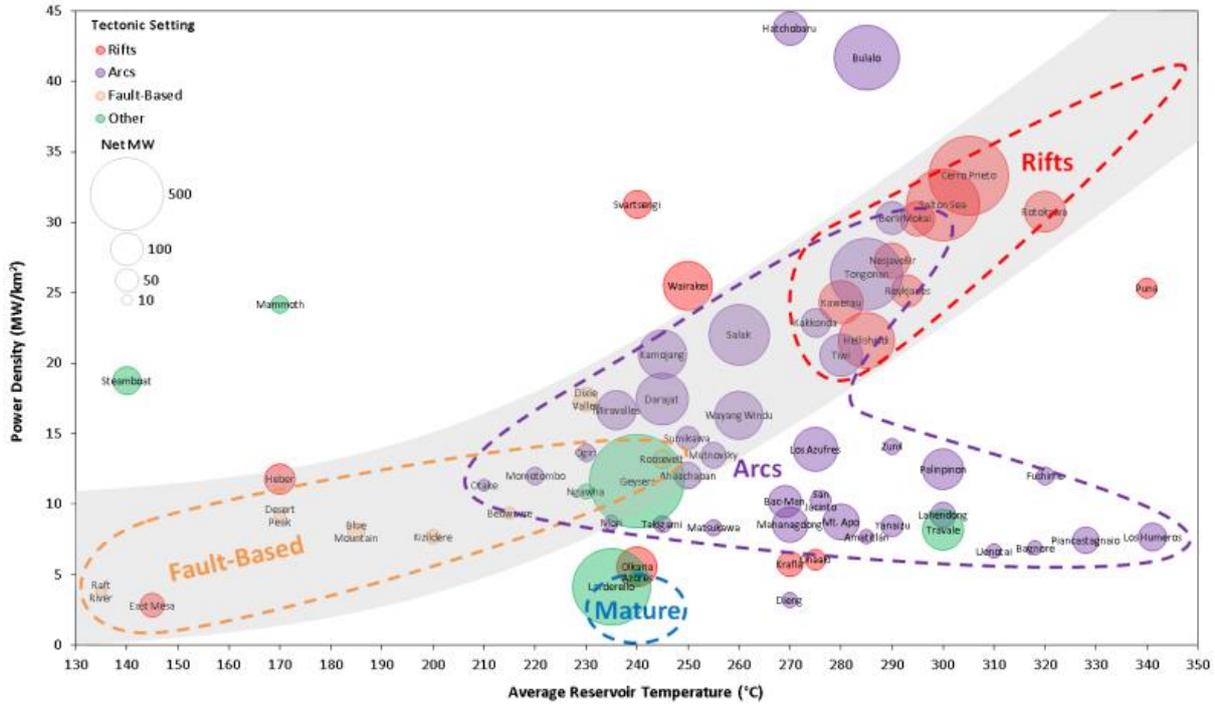


Figure 2. Plot of temperature and power density of 66 geothermal field (Wilmarth & Stimac 2015,[22])

3.1.6. Volumetric method

This method is one of the most used for the evaluation of the geothermal potential and it is also called stored heat method. It was developed by the United States Geological Survey (USGS) in 1975 during the first study of the geothermal resource potential of the United States at a regional scale [23].

The power that can be extracted from the reservoir is evaluated by multiplying the recoverable thermal energy in a volume of porous and permeable rock by the power plant efficiency:

$$MW_e = \frac{q \times R_f \times \eta_{conv}}{F \times L}$$

Where:

- MW_e is the power potential [MW_e];
- q is the thermal energy stored in the reservoir [MJ];
- R_f is the recovery factor [-];
- η_{conv} is the conversion efficiency [%];
- L is the plant life [s];
- F is the capacity or load factor [%].

The reservoir is divided into n different region of volume V_i , each with its own temperature T_i . The heat stored q_i inside each region can then be calculates using the following formula [14],[17]:

$$q = \sum_{i=1}^n \rho_i c_i V_i (T_i - T_f)$$

Where:

- $\rho_i c_i$ is the volumetric heat capacity of a saturated rock [$\frac{J}{m^3 \cdot ^\circ C}$];

- V_i is the volume of i^{th} region of n lithology, which is the product of area A and thickness h of the reservoir ($V = A \times h [m^3]$);
- T_i is the initial temperature of i^{th} lithology [$^{\circ}C$];
- T_f is the cut-off or final abandoned reservoir temperature [$^{\circ}C$].

Since the 90% of heat is stored in the rock, and only the 10% into the fluid [24], it is possible to simplify the previous formula as follows:

$$q = \rho_r c_r V (T_i - T_f)$$

Where:

- $\rho_r c_r$ is the volumetric heat capacity of a saturated rock $\left[\frac{J}{m^3 \cdot ^{\circ}C}\right]$;
- V is the volume of the productive reservoir. Product of area A and thickness h of the reservoir ($V = A \times h [m^3]$);

There are three different methods to estimate the geothermal potential through the volumetric method:

1. The USGS method which considers only the energy contained inside the rock;
2. The USGS method which considers the energy contained inside the rock and the fluid;
3. The AGRCC method which considers the energy contained inside the rock and the fluid.

The latter is a variation from the USGS method, introduced by the Australian Geothermal Reporting Code Committee[25], which presents guidelines to report and calculate the geothermal Resource and Reserve[26].

3.1.6.1. The USGS volumetric method

As previously mentioned, the USGS introduce two methods: the first considers the reservoir as a whole, taking into account only the rock, while the second considers also the presence of fluid inside the porosity of the rock, subdividing the reservoir into sub-regions.

In both cases the initial temperature T_i is the average temperature, and T_f is the reference at a dead-state temperature[14].

The thermal energy which can be extracted at the well-head q_{wh} is evaluated as:

$$q_{wh} = m_{wh} (h_{wh} - h_{ref})$$

Where:

- m_{wh} is the extractable mass flow rate in $\left[\frac{kg}{s}\right]$;
- h_{wh} is the enthalpy of the produced fluid $\left[\frac{kJ}{kg}\right]$;
- h_{ref} is the enthalpy at some reference temperature $\left[\frac{kJ}{kg}\right]$.

The recovery effect inside the reservoir is considered through the recovery factor R_f , which is defined as the ratio between the thermal energy extracted at the well-head and the total one:

$$R_f = \frac{q_{wh}}{q}$$

Then it is possible to evaluate the available thermal power:

$$MW_{th} = m_{wh} [h_{wh} - h_f - T_f (s_{wh} - s_f)]$$

Where:

- h_f is the reference enthalpy $\left[\frac{kJ}{kg}\right]$;
- s_{wh} is the entropy of the extractable fluid at the wellhead $\left[\frac{kJ}{kgK}\right]$;
- s_f is the reference entropy $\left[\frac{kJ}{kgK}\right]$;
- T_f is the reference temperature [K];

The available electric power is obtained by multiplying the available thermal power by the utilization efficiency:

$$MW_e = MW_{th} \times \eta_{util}$$

The second USGS method evaluates the total thermal power inside the reservoir as the sum between the heat stored in the rock and the one stored in the fluid.

Considering also the porosity, the total thermal energy is:

$$q = q_r + q_f = Ah(T_i - T_f)[(1 - \varphi)\rho_r c_r + \varphi\rho_f c_f]$$

Where:

- $\rho_f c_f$ is the fluid volumetric heat capacity $\left[\frac{J}{m^3 \cdot ^\circ C}\right]$;
- φ is the porosity [-].

This equation can be improved by introducing the saturation term to distinguish the fraction of water S_l and steam S_v in the rock pores with $S_l + S_v = 1$. This has been done by Garg and Combs (2015) and O'Sullivan (2016)[15]. It becomes:

$$q = q_r + q_f = Ah(T_i - T_f)[(1 - \varphi)\rho_r c_r + \varphi(S_l \rho_f c_f + S_v \rho_v c_v)]$$

3.1.6.2. The AGRCC volumetric method

This method is very similar to the second one of the USGS, it considers separately the two thermal energy, from rock and fluid. The total heat stored is calculated as:

$$q = Ah \times \{[\rho_r c_r (1 - \varphi)(T_i - T_f)] + [\rho_{wi} \varphi S_{wi} (h_{wi} - h_{wf})] + [\rho_{si} \varphi (1 - S_w)(h_{si} - h_{wi})]\}$$

Where:

- A is the areal extent of the reservoir [m^2];
- h is the average reservoir thickness [m];
- φ is the average porosity of the fluid-saturated rock [%];
- T_i is the average initial temperature [$^\circ C$];
- T_f is the rejection temperature, similar to the reservoir reference temperature [$^\circ C$];
- ρ_{wi} and ρ_{si} are respectively density of steam and water at the initial reservoir condition $\left[\frac{kg}{m^3}\right]$;
- h_{wi} and h_{si} are respectively water and steam enthalpies at reservoir temperature $\left[\frac{kJ}{kg}\right]$;
- h_{wf} is the water enthalpy at rejection temperature $\left[\frac{kJ}{kg}\right]$;
- S_w is the relative water saturation of the reservoir [%].

This formula can be also rewritten using internal energy instead of specific enthalpy, as proposed by Zarrouk and Simiyu (2013) [27]:

$$q = Ah \times \{[\rho_r c_r (1 - \varphi)(T_i - T_f)] + [\rho_{wi} \varphi S_{wi} (u_{wi} - u_{wf})] + [\rho_{si} \varphi (1 - S_w)(u_{si} - u_{wi})]\}$$

Where:

- u_{wi} and u_{si} are respectively water and steam internal energy at reservoir temperature $\left[\frac{kJ}{kg}\right]$;
- u_{wf} is water internal energy at rejection temperature $\left[\frac{kJ}{kg}\right]$.

Since there is the internal energy and not the specific enthalpy, the total mass of fluid inside the reservoir volume can be calculated as:

$$m_r = \varphi(\rho_l S_l + \rho_v S_v)$$

Where:

- φ is the rock porosity;
- ρ_l and ρ_v are the density of water and steam;
- S_l and S_v are the saturation of liquid and vapour ($S_l + S_v = 1$).

The differences between these three methods are minor (Figure 3), as demonstrated by of Tiwi, Makban, Ngatamariki, Ohaaki geothermal fields. The USGS method considering rock and fluid reports a 1% higher thermal energy than USGS method considering only rock, and the AGRCC method reports only 0.5% thermal energy higher compared to USGS method considering only rock. As suggested by Muffler and Cataldi (1978)[18], the USGS method considering only the rock should be used [15].

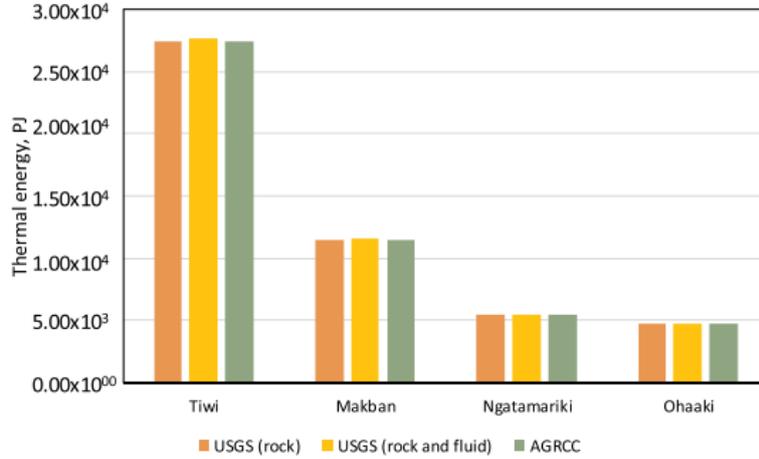


Figure 3. Thermal energy evaluated with three different volumetric heat stored methods of four geothermal fields (Ciriaco et al. 2020, [15])

The last parameter for the evaluation of the power which can be extracted from the reservoir is the conversion efficiency, which can be estimated with two different empirical formulas, one is a function of the temperature, while the other one is a function of the enthalpy[14]:

$$\eta_{conv} = (0.0484 \times T - 0.5096) \times 0.01 \quad \eta_{conv} = 7.6301 \ln(h) - 43.9589$$

The issues in using this method are the uncertainties, the first is related to the estimation of the size of the reservoir, usable for the production, the second is the problem in estimating the recoverable thermal energy fraction, and all the other uncertainties related to the values of input parameters, as volumetric heat capacity, reservoir temperature, enthalpy, porosity, etc. [14].

3.1.7. Mass-in-place

This method was introduced by Parini and Rield (2000) [28], and is very similar to the volumetric one, but instead of using the volume, it refers to the total mass of the reservoir, which is described as:

$$MIP = V\varphi\rho(T_i)$$

where:

- V is the volume of the reservoir [m^3];
- φ is the average porosity [-];
- $\rho(T_i)$ is the fluid density [$\frac{kg}{m^3}$] at the temperature T_i [$^{\circ}C$].

The electrical power possible to produce is:

$$MW_e = \frac{MIP * R_m}{SR * F * L}$$

where:

- R_m is the mass recovery factor defined as the total steam produced (M_s) over the initial mass in place (MIP);
- SR is the steam rate, or usage factor [$\frac{kg}{s} MW_e$];
- L is the plant life [s];
- F is the capacity or load factor [%].

By combining numerical simulations with the volumetric method, the Mass-in-place method evaluates both the recoverable and total available mass of geothermal resource. Unlike the volumetric method, the recovery factor in this approach refers to the mass and not to the heat. In general, this method tends to underestimate the real potential of geothermal resource [14].

3.2. History method

3.2.1. Lumped parameter

In this method, the reservoir is modelled as a closed tank that can exchange fluid with the external environment. When water is drawn from the reservoir, there is a decline in pressure, described as a linear function of the cumulative production. As a result, the mass and energy equations are reduced to ordinary differential equations [14].

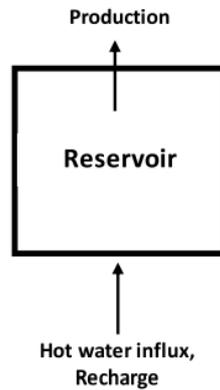


Figure 4. Lumped-parameter method, reservoir as a closed tank (Ciriaco et al. 2020,[15])

For example, if the reservoir undergoes a pressure change, it expels a fluid mass proportional to it, following the conservation of mass:

$$m \frac{dP}{dt} + W_{prod} - W_{rech} = 0$$

where:

- m is the mass of geothermal fluid [kg];
- $\frac{dP}{dt}$ is the variation of the pressure over time $\left[\frac{Pa}{s}\right]$;
- W_{prod} is the rate of production [-];
- W_{rech} is the rate of recharge [-].

This method uses average reservoir properties to describe its condition, and the time is the only independent variable. It is widely used to study field response to production, injection returns and thermal decline. It is also used to understand and describe the reservoir behaviour during exploration and pre-exploration stages. While valuable for short-term production capacity assessment, its predictive accuracy remains limited [14],[29].

Figure 5 shows an example of the lumped parameter method used to assess the Hofstadir geothermal field, carried out by Gaoxun et al. (2010)[31]. They want to study the connection between extraction well HO-01 and the injection well HO-02, to predict the future water level considering the potential re-extraction of a portion of injected water. They find that the system could sustain a stable production of 20 l/s through 2032 without re-injection. Then they use a Monte Carlo simulation to estimate the potential power generation, indicating a 90% probability of producing 25 MW_e for 30 years, 12 MW_e for 60 years and 7 MW_e for 100 years.

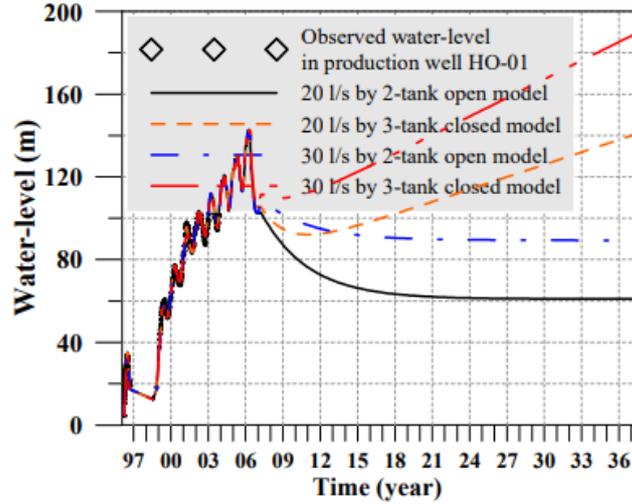


Figure 5. Water level variation in time considering different tank models (Gaoxuan et al. 2010, [31])

3.2.2. Decline analysis

This method is used to predict the resource potential in a short period. It requires history production data, which is fitted with a proper equation and then predict the future production.

To analyse the decline over time, it is possible to use different equations:

1. Exponential: $\frac{W(t)}{W_t} = \frac{1}{\exp(Dt)}$;
2. Harmonic: $\frac{W(t)}{W_t} = \frac{1}{(1+Dt)}$;
3. Hyperbolic: $\frac{W(t)}{W_t} = \frac{1}{(1+bDt)^{\frac{1}{b}}}$.

They are all based on the assumption that the decline rate $\frac{dW}{dt}$ is proportional to the production rate W raised to an empirical exponent b :

$$\left(\frac{1}{W}\right) \frac{dW}{dt} = -DW^b$$

where:

- W is the production rate;
- $b=0$ (exponential) or $b=1$ (harmonic);
- D is the decline rate.

The Decline analysis is not suitable for long-term reserve estimation and, also in this case, the predictive capability is limited, and it is inferior to a well-calibrated 3D numerical reservoir model [14].

Orizonte et al. (2005)[32] used the decline analysis to understand which power is possible to extract from the already existing production well situated in the Palinipinon-2 area of the Southern Negros Geothermal Production Field. It results that the production well could still support an additional modular plant from 20 MW_e to 38.3 MW_e .

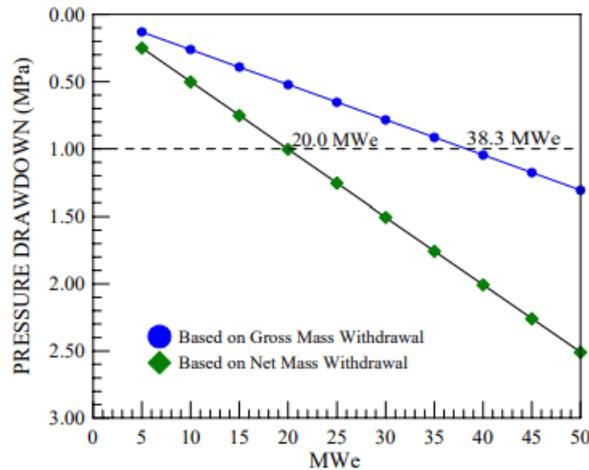


Figure 6. Decline analysis of Palinipinon-2 area (Orizonte et al. 2005,[32])

3.2.3. Numerical reservoir simulation

This method is the most reliable and advanced tool for the evaluation of the geothermal potential. It models the physics of the fluid flow and the heat transfer inside the geothermal reservoir, but it requires a considerable amount of data because reservoir modelling is a continuous process of data interpretation, model calibration and validation [14].

There are 3 main stages for the numerical model simulation:

1. Development of Conceptual Model: it requires to understanding the geological, hydrogeological and thermogeological properties of the reservoir;
2. Numerical Model Calibration: in this stage the pre-exploration data regarding temperature and pressure profiles and surface manifestation must be matched. Then, using that state of the reservoir as starting point, the historical production data are matched to simulate the injection and withdrawal;
3. Forecasting: at this stage the calibrated model is used to forecast future production scenarios[30].

The three most used reservoir model simulators are:

- I. STAR [31]
- II. THOUGH2 [32]
- III. TETRAD [33]

All these simulators can handle multiphase and multicomponent flows[30].

Numerical models of reservoirs are difficult to build, and the calibration process is also time-consuming, but they are the best solution to simulate the inner processes. They also ensure consistency with conceptual models and help in planning exploration and monitoring programs [15].

Using both the volumetric method and numerical simulation can be helpful in the early-stage development. While numerical simulation must be preferred, the volumetric method can provide quick and reliable results in the estimation of the geothermal potential when there is a limited information about the resource [14].

Zhang et al. (2016)[37] performed a numerical reservoir simulation to understand potential production of the Chingshui geothermal power plant, which was decommissioned in 1993 due to a continuous decline in production. Using this method, they find that the decline in production was caused mainly by carbonate scaling, and it is possible to have a suitable water production to run a 3 MW_e power plant.

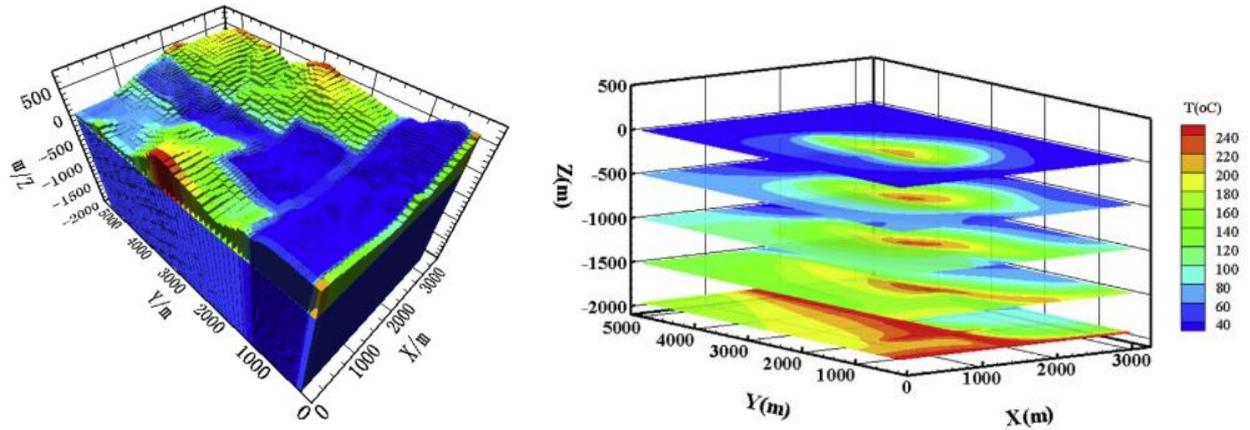


Figure 7. Spatial discretization of the reservoir (left) and temperature distribution at different depths (right). (Zhang et al. 2016, [37])

3.3. Probabilistic resource assessment method

It is possible to combine volumetric method, numerical simulation and probabilistic methods to estimate the geothermal potential. In this combined method each parameter has a range of possible values, and so a probability distribution of the produced capacity is generated [14].

The volumetric method is the most suitable for this approach, each parameter can have a different probability distribution function as triangular, logarithmic or uniform, and at each repetition, one value is randomly carried out based on the PDF and the power output is calculated. This process is repeated multiple times until the results are reliable and reproducible [14],[28].

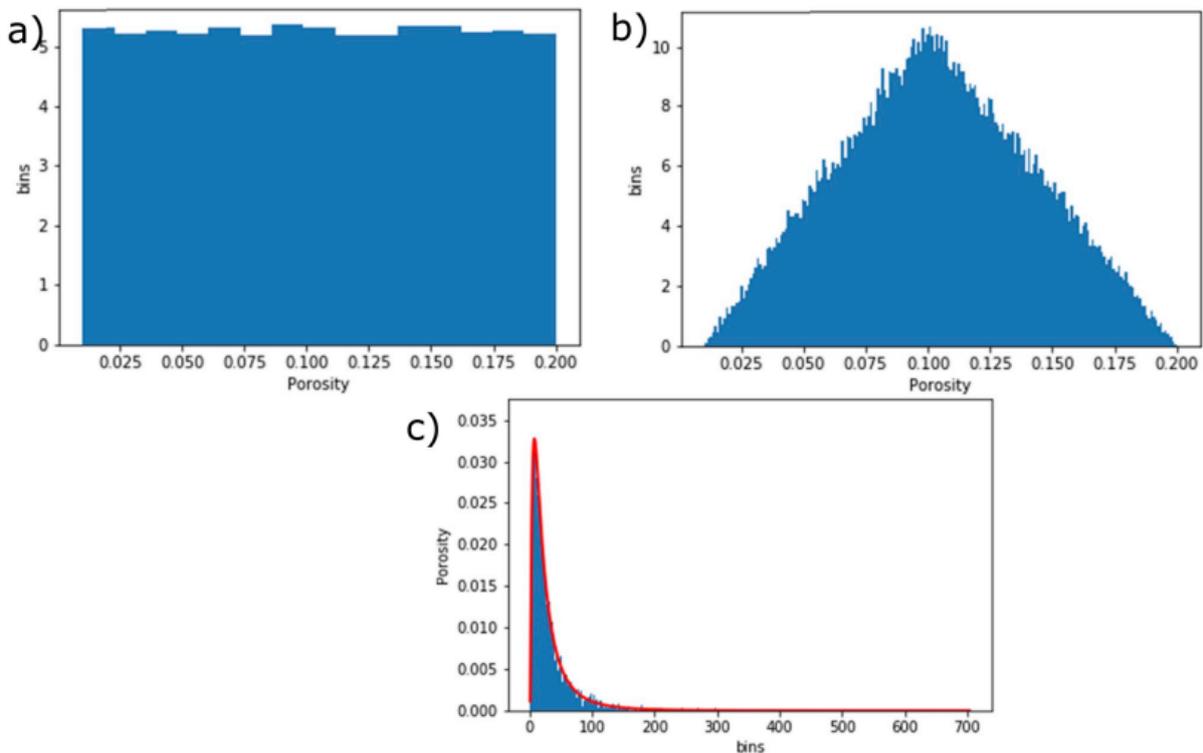


Figure 8. Porosity probability distribution: a) Uniform distribution, b) Triangular distribution, c) log-normal distribution (Ciriaco et al. 2020, [15])

Usually, the Monte Carlo method is used, and the results are reported in terms of P90, P50 and P10, which refer to the low, best and high estimates. However, this method is time-consuming due to the high number of repetitions required, and it is not suggested in an early stage of development [14].

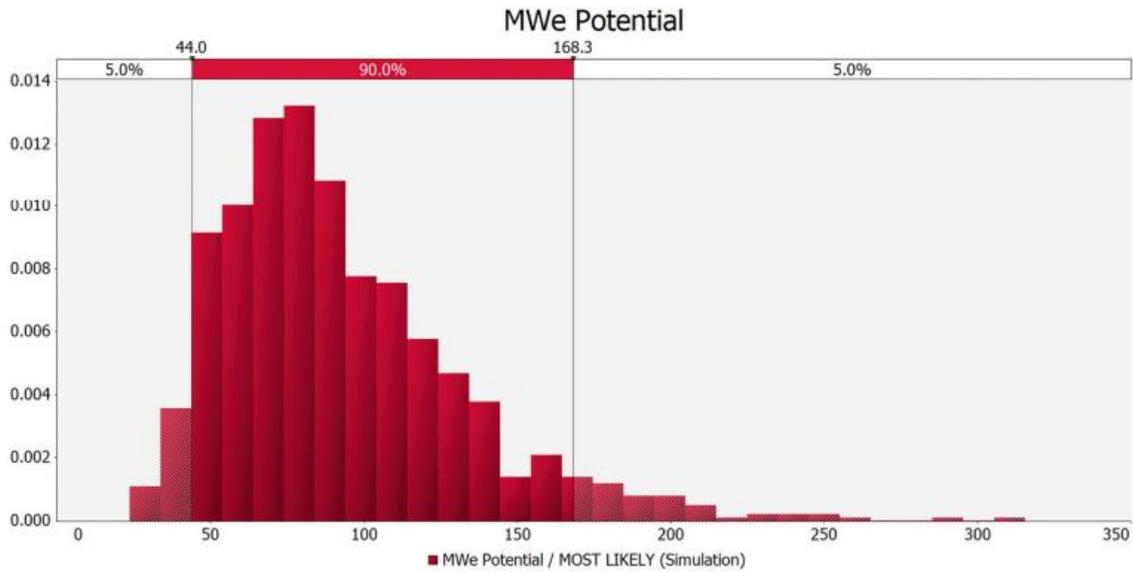


Figure 9. Probability distribution of the plant's power output (Ciriaco et al. 2020, [15])

In conclusion, there are different methods for estimating the geothermal potential, and the choice of method depends on the accuracy and the availability of data. The best solution is to combine the volumetric heat stored method with numerical reservoir simulation. While this process requires time and a considerable amount of data, it offers the most reliable results. The Monte Carlo method can also be introduced when probabilistic analysis is required, as it complements the volumetric method well, but it is also time-consuming. In case of low data availability or if a preliminary estimation is required, methods like power density or surface heat flux can be used, although they will likely underestimate the potential.

The proposed new code combines some features of the volumetric heat stored method, which is a static model, with some features of the numerical simulation method, which is dynamic. The temperature, pressure and mass flow rate change over time, depending on external factors such as injection temperature, pressure, and withdrawal rate.

4. Methodology

The code includes three main parts: a geological model of the area of interest, a thermodynamic cycle for the power plant and an economic model.

The area is divided into cells based on the desired resolution; usually, the area of each cell is in the order of kilometers, while the depth is on the order of hundred meters.

The code calculates the temperature, pressure and extractable mass flow rate of geothermal fluid for each cell. Using these parameters, it evaluates the thermodynamic cycle of the chosen plant. The outputs from this evaluation are then used to conduct an economic analysis, in which the components are sized to meet the plant's requirements. At the end, the code produces plots showing the geothermal potential, the Levelized Cost of Energy (LCOE) and the Net Present Value (NPV) at the end of the power plant's life. For this purpose, a review of existing codes, genGEO (2020)[38] and GEOPHIRE (2020)[39], has been performed. The former emphasizes the accurate evaluation of the thermodynamic cycle within the power plant, with less focus on surface conditions, while the latter is more concentrated on subsurface characteristics, often at the expense of evaluating electricity production or heat utilization.

This code combines elements of the volumetric heat storage method, which is a static model, with features of numerical simulation methods, which are dynamic. Thus, the temperature, pressure and mass flow rate change over time.

4.1. Geological model

Accessing the deeper levels of the crust in areas subjected to elevated thermal gradients is an expensive phase of a geothermal project. Therefore, the development of a thermal numerical model is an early stage practice to understanding the geothermal system and assessing its potential before the drilling exploration start. Numerical modelling techniques constitute an indirect method and a powerful tool through which geoscientists simulate the physical conditions of the crust in geothermal areas. The models incorporate geological information, rock properties, fluid characteristics to simulate the fluid flow in porous/fractured materials (rocks) and convective and conductive heat transfer processes. By comparing the results with the field observations (e.g. borehole temperatures), researchers can validate and refine their understanding of the geothermal system under study. The workflow includes the developments of:

1. a conceptual model, obtained by the integration of several evidence provided by different disciplines of the geosciences, namely surface geology, geophysical and geochemical investigations. The available the information are employed to characterize the main components of a geothermal system: the impermeable cover (or cap-rock), the reservoir, the recharge area, the geothermal fluids and the heat source, as depicted in Figure 10.

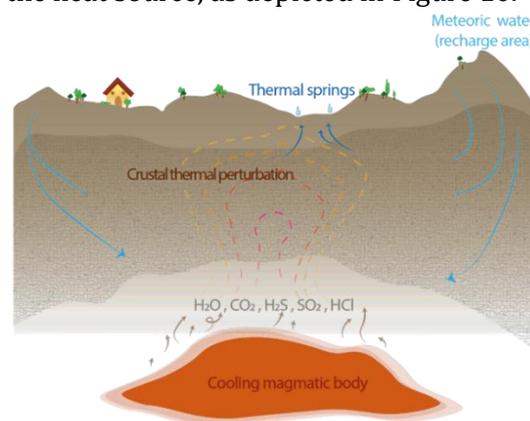


Figure 10. The conceptual model of a magmatic, hydrothermal geothermal system

Figure 10 illustrates the fluid flow resulting from a deep magmatic body (the heat source) which release mass (magmatic fluids) and heat, the reservoir whose temperature results from the interference between the lateral flow from the recharge areas and the convective flow due to the buoyancy forces acting on the fluid in presence of high geothermal gradients, the cap-rock which prevent the surface dispersion of fluid and heat, and the surface manifestations (e.g., thermal springs, fumaroles) as natural evidence of the existence of a deep-seated hydrothermal reservoir.

2. a geological model, obtained by the integration of geological (surface geology and boreholes) and geophysical (mainly seismic) data. The geological model describes the depth to the main geological discontinuities, and it represents the geometrical framework to simulate heat transfer processes.
3. a petrophysical model, obtained by the integration of thermal and hydraulic properties measured in laboratory on representative rock samples collected at surface or cored in the exploratory boreholes
4. a natural state thermal model, obtained by the integration of the above-mentioned information. It results from the optimization of the boundary condition of the numerical model to minimizing the misfit between the simulated and observed temperatures.

4.1.1. Mathematical model

Comsol Multiphysics® (v6.1), a simulator based on the finite-element scheme, and it has been employed for numerically simulate the three-dimensional temperature distribution in the Sabatini-Cesano geothermal field. The heat and fluid transfer processes are described by the combination of the mass, momentum, and energy conservation equations:

$$\begin{aligned}\rho_w \nabla \cdot u &= Q_m \\ u &= -\frac{K_r}{\mu_w} (\nabla p + \rho_w g \nabla z) \\ (\rho c_p)_r \frac{\partial T}{\partial t} &= k_r \nabla^2 T - (\rho c_p)_w u \nabla T + HS + A\end{aligned}$$

Where:

- u is the Darcy flux $\left[\frac{m}{s}\right]$;
- Q_m is the mass flow $\left[\frac{kg}{s}\right]$;
- g is the gravitational acceleration $\left(9.81 \left[\frac{m}{s^2}\right]\right)$;
- p is the pressure $[Pa]$;
- z is the depth, positive below ground surface $[m]$;
- ρ is the density $\left[\frac{kg}{m^3}\right]$;
- K is the intrinsic permeability $[m^2]$;
- μ is the dynamic viscosity $[Pa \cdot s]$;
- T is the absolute temperature $[K]$;
- t is the time $[s]$;
- c_p is the specific heat $\left[\frac{J}{kgK}\right]$;
- k is the thermal conductivity $\left[\frac{W}{mK}\right]$;
- HS and A are the heat source terms due to the magmatic intrusion and the radiogenic heat production of crustal rocks, respectively $\left[\frac{\mu W}{m^3}\right]$;

The subscripts r and w refer to the porous rock and water properties, respectively.

In the definition of the convective problem, the Boussinesq approximation is assumed. Then, the density variations have a negligible effect on the fluid volume via conservation of mass, which reduces to the

condition of incompressibility. Concerning the thermal boundary conditions, a fixed basal temperature is defined at the bottom of the numerical domain and a stationary surface temperature (T_{surf}) is set at the ground surface:

$$T_{surf} = T_0 - G_{air} \cdot z$$

Where T_0 is the mean annual air temperature at the sea level, G_{air} ($= 9.8 \left[\frac{^\circ \text{C}}{\text{km}} \right]$) the adiabatic lapse rate for the Earth's atmosphere and z (m a.s.l.) is the topographic elevation. The differential equations are approximated through the finite element method on a tetrahedral mesh grid counting more than 106 nodes. The geometrical models consider three main lithothermal units, from the top to the bottom:

- the impervious sedimentary cover unit acting as cap-rock;
- the carbonate units hosting the main regional reservoir;
- the basement unit whose upper boundary has been inferred from aero-magnetic data [40].

Different rocks having similar thermal and hydraulic properties composed each lithothermal unit. The rocks have been treated as a homogeneous and downward anisotropic porous material. Mixing laws have been applied to estimate the effective thermal and hydraulic properties of the rock-water system accounting for the in-situ conditions (depth and temperature). Since the cap-rock and the basement units consist of mainly impermeable rocks, we evaluated the thermal effects of the interplay of the free convection and topographically driven groundwater flow in the permeable reservoir domain. The fluid-velocity and the pore-pressure fields in the impermeable units are fixed to zero and to follow a hydrostatic profile, respectively. In those domains, the fluid is stationary and then a purely conductive heat transport takes place. Instead, in the carbonate reservoir regional hydraulic gradients and hydrothermal convection affect the temperature field by mass and energy transport. The buried upper and basal boundaries of the reservoir are impermeable to fluid flow allowing only conductive heat transfer. Accounting for the recharge zones, it is applied a stress boundary condition where the reservoir units crop out. As the reservoir is assumed fully saturated, the pressure on those boundaries is set equal to the freshwater head calculated with a reference water density of $1000 \left[\frac{\text{kg}}{\text{m}^3} \right]$ and the sea level as datum. Furthermore, in the Latium Magmatic Province has been simulated the effects on the thermal field of deep crustal magmatic bodies by setting a constant temperature to the intrusive body surfaces. The vertical boundaries surrounding the multilayers 3D block do not allow horizontal flow of fluid and heat. As physical assumptions about rock properties and boundary conditions often suffer of large uncertainties, different scenarios varying reservoir hydraulic permeability and thermal constraint have been evaluated. The scheme of the parametric study consists in several solutions resulting from all the combinations of the values of permeability and basal temperature variable in the range $10^{-16} \div 10^{-13} \text{ [m}^2\text{]}$ and $300 \div 500 \text{ }^\circ\text{C}$, respectively. The temperature of the magmatic bodies varies from 400 to 800°C depending on the nature of magmatic source. In Figure 11 the setting of a thermal model is displayed.

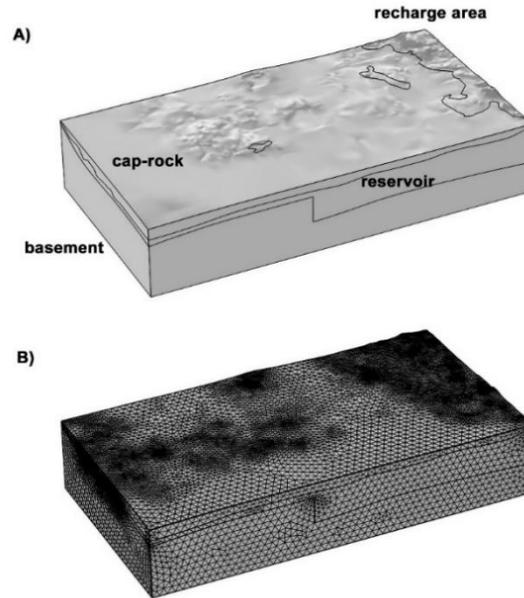


Figure 11. Setting of the numerical model: the 3D geometry (A) including the cap-rock, the reservoir and the basement domains. The numerical mesh employed to solve the thermal model

4.1.2. Thermal properties of the rocks

Knowledge of thermal properties and high-temperature behavior of rock/fluid systems has become increasingly important with the widespread interest in thermal processes occurring in the subsurface. In the Earth's lithosphere, there are two principal mechanisms which contribute to the effective thermal conductivity of rocks:

1. the conduction of heat by phonon propagation (lattice conductivity k_{lat});
2. the transfer of heat through emission and absorption of photons (radiative conductivity k_{rad}).

Thus, it is generally recognized that the rock's effective thermal conductivity is the sum of two components:

$$k_{eff} = k_{lat} + k_{rad}$$

The lattice thermal conductivity of materials depends on different factors such as mineral composition, pressure, and temperature. Under the assumption that rocks are composed of randomly distributed mineral grains and void spaces, the lattice component of thermal conductivity is estimated by the geometric mean model [41]:

$$k_{lat} = k_m^{(1-\phi)} \cdot k_w^{(\phi)}$$

where k_m and k_w are the matrix and pore-fluid thermal conductivity, respectively, and ϕ the porosity index. Igneous and metamorphic rocks generally have higher thermal conductivities compared to sedimentary rocks and soils, making them more efficient at conducting heat. Thermal conductivity in some rocks is, to a good approximation, isotropic, particularly for volcanic and plutonic rocks. Thermal conductivity of many sedimentary and metamorphic rocks, in contrast, is strongly anisotropic, and lateral heat flow will be significant. Hence information on anisotropy is often required, demanding laboratory measurements in different directions.

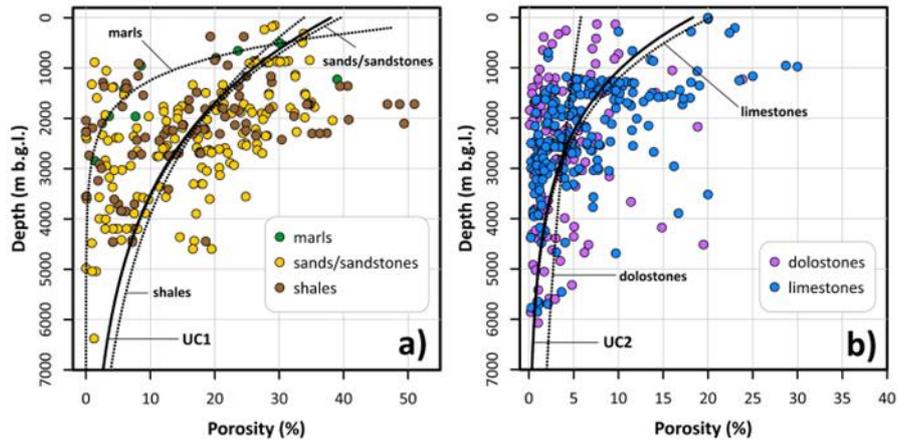


Figure 12. Compaction exponential curves by lithology derived from porosity measurements on core samples coming from different depth intervals

The increasing overburden pressure (lithostatic pressure) reduces the porosity. To describing the decreasing trend of porosity as function of depth, it is employed an exponential function:

$$\phi(z) = \phi_0 \cdot e^{-c \cdot z}$$

Where ϕ_0 is the porosity at the surface ($z=0$) and c is the compaction factor (km^{-1}).

According to Debye's theory of phonon scattering, k_{lat} decrease as the temperature increases. Theory indicates that thermal conductivity of solid materials should vary with the reciprocal of temperature. In the case of mixture of crystals or highly disordered crystals, thermal conductivity varies more slowly than T^{-1} and experimental data indicate that the rock's thermal conductivity as function of temperature follows a non-linear relationship. The temperature dependence of the matrix thermal conductivity may be described by the Sekiguchi (1984) empirical formula [42]:

$$k_m(T) = k_M + \left[\frac{T_o \cdot T_M}{T_M - T_o} \cdot (k_{mo} - k_M) \cdot \left(\frac{1}{T} - \frac{1}{T_M} \right) \right]$$

Where:

- k_M is the matrix thermal conductivity at the temperature $T_M (= 1473 K)$;
- k_{mo} is the matrix thermal conductivity at ambient temperature $T_o (= 293K)$.

k_M and k_{mo} are the function's coefficients, whoso values can be calculated by least-square fitting of the laboratory thermal conductivity data measured at increasing temperatures Figure 13.

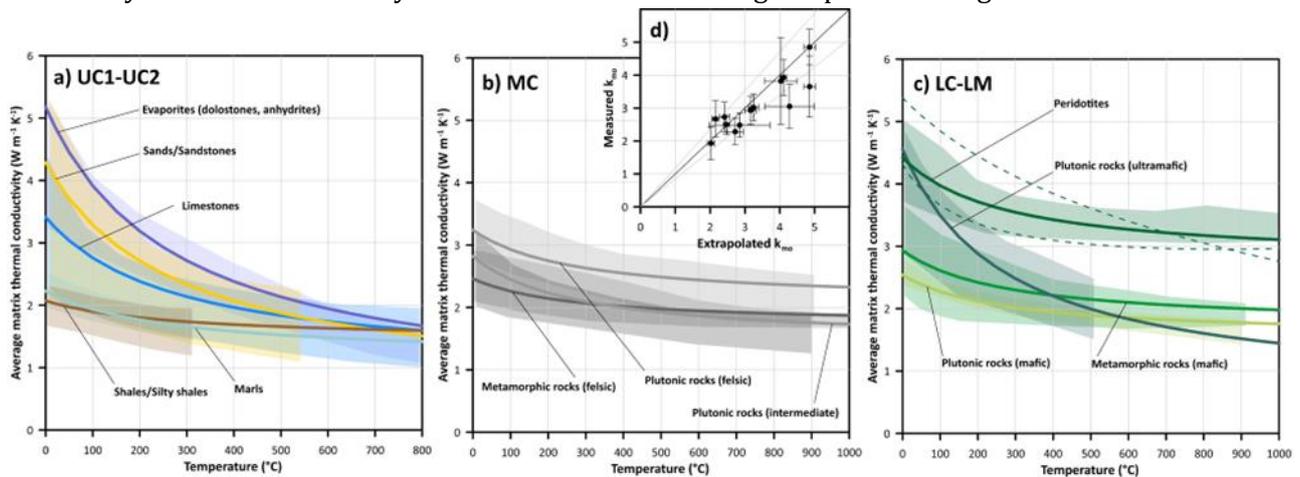


Figure 13. Laboratory thermal conductivity measurements as function of temperature from ambient to 1000 °C.

At high temperatures ($T > 800 - 1000 \text{ }^\circ\text{C}$) the radiative component of thermal conductivity becomes more efficient, and it may compensate for the decreasing trend of k_{lat} or even reverse the resulting k_{eff} . All materials at finite temperature emit thermal radiation and the magnitude of thermal radiation may

be roughly approximated to follow a T^3 law. Involving the emission and absorption of electromagnetic waves, particularly infrared radiation, this mode of heat transfer significantly complements phonon conduction mechanism in rocks. Due to its effectiveness at high temperature and large depths, the radiative component is not considered as one of the primary contributions to the thermal budget of the crust and it is usually evaluated for the lithospheric mantle. At present, only approximate results and limited information on the pressure, temperature, composition, and grain-size dependence of k_{rad} are available. Due to the lack of data, k_{rad} should be accounted for the mantle according to Hofmeister (2014)[43]:

$$k_{rad}(T) = 1.9 \cdot 10^{-10} T^3$$

Where T is the temperature (K).

Apart from the heat content of the Earth immediately after formation, the radiogenic decay of the unstable isotopes of uranium, thorium, and potassium provides the largest internal source of heat. Most of these isotopes are enriched in the Earth's crust and mantle. During radioactive decay, mass is converted into energy. Experimental findings demonstrate that the concentration of radioactive elements, and subsequently the radiogenic heat production, is influenced by magmatic differentiation processes. The generation of continental crust through magmatic differentiation significantly enriched the upper portions of the lithosphere with radioactive elements, among which uranium-238 (^{238}U), uranium-235 (^{235}U), thorium-232 (^{232}Th), and potassium-40 (^{40}K) account for almost the total heat production. This implies that radiogenic heat decreases with depth within the Earth's crust and lithosphere. The most widely accepted model is the exponential model by Lachenbruch (1970)[44]:

$$A(z) = A_0 \cdot e^{-\frac{z}{D}}$$

Here, A_0 (in $\mu\text{W m}^{-3}$) represents the radiogenic heat at the Earth's surface, and D (in km) is the decreasing rate with depth. This model accurately accounts for the variation in radiogenic heat in surface rocks within a region with a specific tectono-thermal history, attributed to the superficial removal of radioisotopes. The parameter D typically ranges from 5 to 15 km and averages around 10 km [45]. Surface radiogenic heat in continental areas is generally estimated to be around $3 \mu\text{W m}^{-3}$. Radioactivity measurements of igneous and metamorphic rocks have constrained the actual range of radiogenic heat to be within $2.5\text{--}3.5 \mu\text{W m}^{-3}$.

4.1.3. Wellbore heat losses

"As fluids move through a wellbore, there is transfer of heat between fluids and the earth due to the difference between fluid and geothermal temperatures", Ramey (1962)[46].

The heat losses and the following temperature drop during the ascent of the geothermal fluid act as link between the subsoil and the surface power plants. Using the model proposed by Ramey, giving as inputs the parameters outcome from the mathematical model, the output will be the temperature of the geothermal fluid at the wellhead.

To evaluate the heat losses in the wellbore and so the production temperature, the mass flow rate must be known. It can be evaluated using the equation reported by Dake (1983)[47]:

$$q = \frac{2\pi K h}{\mu\beta \left(\ln \left(\frac{r_e}{r_w} \right) + S - 0.75 \right)} (p_{res} - p_{bh}) \left[\frac{\text{m}^3}{\text{s}} \right]$$

Where:

- K is the permeability [m^2];
- h is the height of the open window in the well to let the fluid enters [m];
- μ is the viscosity of the fluid [$\text{Pa} \cdot \text{s}$];
- β is the compressibility factor, evaluated as ratio between the specific volume of the fluid at reservoir and surface condition [–];

- $\frac{r_e}{r_w}$ is the ratio between the drainage radius and the inner radius of the well [-];
- S is the skin factor and it accounts for possible damages of the well caused by drilling operations (assumed equal to zero) [-];
- p_{res} is the pressure inside the reservoir [Pa];
- p_{bh} is the dynamic pressure of the borehole one it is in production (assumed equal to $0.7p_{res}$) [Pa].

The drainage radius is approximate as:

$$r_e \approx 0.2\sqrt{\Delta x \cdot \Delta y} [m]$$

Where Δx and Δy represent the distances between two wells.

The pressure inside the reservoir must be calculated, as the viscosity, in order to evaluate the mass flow rate which will enter the well, while the other parameters will be assumed constant. It can be done starting from 3D matrix of the temperature distribution within the soil given as output from the mathematical model. Hence it is possible to evaluate the density of the fluid and then the pressure. To make these calculations the assumption of only liquid phase inside the reservoir has been done, because it allows to use the XSteam function [48]. Starting from the higher level of the 3D temperature matrix, the following equations are implemented for the evaluation of the pressure 3D matrix:

$$\rho_L(x_i, y_j, z_k) = XSteam(T(x_i, y_j, z_k)) \left[\frac{kg}{m^3} \right]$$

$$p(x_i, y_j, z_k) = p(x_i, y_j, z_{k-1}) + \rho_L(x_i, y_j, z_k) \cdot g \cdot \delta z [Pa]$$

Where x_i, y_j, z_k indicate the position inside the matrix, g is the constant gravity acceleration, and δz is the difference in height between two cells on the z-axis.

Once found the 3D matrices for pressure, a section at the wanted depth is suggested to evaluate the mass flow rate and the production temperature in function of the time. To consider the time variation some equations must be applied to each cell of the 2D matrices of temperature and pressure. If the time variation is not considered it is possible to simply repeat the matrix for the length of the time vector.

Then viscosity, as thermal conductivity and specific heat capacity, which will be used later on, are evaluated through XSteam:

$$\mu_z(x_i, y_j, t_k) = XSteam(p_z(x_i, y_j, t_k), T_z(x_i, y_j, t_k)) [Pa \cdot s]$$

$$c_{p,z}(x_i, y_j, t_k) = XSteam(p_z(x_i, y_j, t_k), T_z(x_i, y_j, t_k)) \left[\frac{k}{kgK} \right]$$

$$k_z(x_i, y_j, t_k) = XSteam(p_z(x_i, y_j, t_k), T_z(x_i, y_j, t_k)) \left[\frac{W}{mK} \right]$$

The following step is the evaluation of the global heat transfer coefficient inside the well. So, the Reynold number is calculated, followed by the Prandtl number, the friction factor, the Nusselt number, and the convective heat transfer.

$$Re_z = \frac{q_z \rho D_1}{\mu_z A} [-]$$

$$Pr_z = \frac{c_{p,z} \mu_z}{k_z} [-]$$

$$f_{D,z} = \left[-1.8 * \log_{10} \left(\frac{6.9}{Re_z} + \left(\frac{\varepsilon}{3.7D_1} \right)^{1.11} \right) \right]^2 [-]$$

$$Nu_z = \frac{(8f_{D,z}) \cdot (Re_z - 1000) \cdot Pr_z}{1 + 12.7 \cdot 8f_{D,z} \cdot \left(Pr_z^{\frac{3}{2}} - 1 \right)} [-]$$

$$h_z = Nu_z \left(\frac{k_z}{D_{in}} \right) \left[\frac{W}{m^2K} \right]$$

$$U_z = \frac{1}{\left(\frac{R_2}{R_1} \frac{1}{h_z}\right) + R_2 \left(\ln\left(\frac{R_2}{R_1}\right) \frac{1}{k_{pipe}} + \ln\left(\frac{R_3}{R_2}\right) \frac{1}{k_{grout}}\right)} \left[\frac{W}{m^2K} \right]$$

Where R_1, R_2, R_3 are the inner, external, and well radius respectively. The last one takes into account the grout which cover the inner pipe. k_{pipe} and k_{grout} are the thermal conductivity of pipe and grout, and ε is the roughness of the pipe.

Then it is evaluated the dimensionless time factor (t_D), to find the correct time function (f_t) to use in the evaluation of the heat losses:

$$t_D = \frac{k_r t}{\rho_r c_{p,r} R_3^2} [-]$$

$$\begin{cases} f_t = (0.4063 + 0.5 \ln(t_D)) \cdot \left(1 + \frac{0.6}{t_D}\right), & t_D > 1.5 \\ f_t = 1.1284 \cdot \sqrt{t_D} \cdot (1 - 0.3\sqrt{t_D}), & t_D < 1.5 \end{cases}$$

Where t is the time in a year.

At this point the last two calculations are the heat losses and the production temperature for each cell of the 3D matrix:

$$B_z(x_i, y_j, t_k) = \frac{q_z(x_i, y_j, t_k) \cdot \rho_z(x_i, y_j, t_k) \cdot c_{p,z}(x_i, y_j, t_k) \cdot (k_r + R_1 \cdot U(x_i, y_j, t_k) \cdot f_t(t_k))}{2\pi R_1 \cdot U(x_i, y_j, t_k) \cdot k_r} [m]$$

$$T_{l,z}(x_i, y_j, t_k) = G_z(x_i, y_j, t_k) \cdot B_z(x_i, y_j, t_k) [^\circ C]$$

$$T_{prod,z}(x_i, y_j, t_k) = G_z(x_i, y_j, t_k) \cdot Z + T_s - T_{l,z}(x_i, y_j, t_k) + (T_z(x_i, y_j, t_k) + T_{l,z}(x_i, y_j, t_k) - T_s) \cdot e^{-\frac{Z}{B_z(x_i, y_j, t_k)}} [^\circ C]$$

Where:

- $B_z(x_i, y_j, t_k)$ is the borehole thermal characteristic length [m];
- $G_z(x_i, y_j, t_k)$ is the thermal gradient $\left[\frac{^\circ C}{m}\right]$;
- $T_{l,z}(x_i, y_j, t_k)$ is the temperature loss along the borehole [°C];
- T_s is the surface temperature [°C];
- Z is the chosen depth [m].

The thermal gradient is estimated as the ratio of the temperature difference between the surface and a chosen depth, divided by the depth.

To completely link the subsurface model with the surface one must calculate the pressure at the production well. It is approximated as the difference between the reservoir pressure and the hydrostatic pressure of the column of fluid at a temperature equal to the production one, which influences the density:

$$p_{prod} = p_{res} - \rho(T_{prod})gZ [Pa]$$

4.2. Thermodynamic cycles of plants

The type of thermodynamic cycle influences the geothermal potential, as each operates under different conditions. For example, a dry steam power plant requires water in the form of steam to exit from the wells. Therefore, the user can choose the type of plant for analysis, including:

1. Dry steam power plant;
2. Single Flash power plant;
3. Binary power plant.

These three types operate at different geothermal resource temperatures, covering all the possible range of temperatures that can be found within the soil. They represent the most basic type of cycle for each temperature range, although they may not be the most efficient, excluding dry steam plant. This is

because the thermodynamic cycle of each type does not incorporate upgrades to the basic design. For example, a recuperator in the binary cycle preheats the organic fluid before it enters the heat exchanger, thereby increasing efficiency.

To run the thermodynamic cycle, the idea is to evaluate some physical properties at each point in the plant using external functions, such as X Steam (2024) [48] for water and Realprop (2022) [49] for the organic fluid inside the binary cycle. For example, knowing the temperature and pressure at a specific point allows the evaluation of enthalpy through these external functions.

In addition to external functions, the user must set some variables, such as the efficiencies of components; otherwise, default values are provided by the program.

Each point in the plant is referred to as “state”, followed by a number corresponding to its position inside the thermodynamic cycle, in accordance with the reference maps of the plants. These “state” are matrices that collect the physical properties through the lifetime of the plant, as conditions within the soil change over time. The matrix dimension is 9×30 , where 9 is the number of physical properties recorded, and 30 is the lifetime of the plant, a value that the user can modify if a different lifespan is chosen.

The physical properties saved inside the matrix are:

- Pressure, p_i [bar];
- Temperature, T_i [$^{\circ}C$];
- Mass flow rate, m_i [$\frac{kg}{s}$];
- Specific heat capacity at constant pressure, cp_i [$\frac{kJ}{kgK}$];
- Density, ρ_i [$\frac{kg}{m^3}$];
- Enthalpy, h_i [$\frac{kJ}{kg}$];
- Entropy, s_i [$\frac{kJ}{kgK}$];
- Vapor fraction, $xvap_i$ [-];
- Specific exergy, e_i [$\frac{kJ}{kg}$].

The correct symbols correspond to those used in the code.

Specific exergy is the only property evaluated using a formula rather than a function, defined as:

$$e_i = (h_i - h_0) - T_0(s_i - s_0) \left[\frac{kJ}{kg} \right]$$

where i stands for the i -th point inside the plant, while 0 is the dead state condition.

4.2.1. Dry steam power plant

The dry steam power plant uses the simplest thermodynamic cycle among the three proposed, but it requires the geothermal fluid to be in a steam state at the wellhead outlet. Unfortunately, this condition is the rarest among geothermal resources, and there are only a few locations in the world where type of plant can be used [50].

The thermodynamic cycle is composed of six stages (Figure 14). Firstly, the steam exiting from the extraction well enters a filter unit, made by a cyclone filter, to separate any moisture particles from the steam. Then, the steam expands through the turbine and condenses in a direct contact condenser before entering the cooling tower, where it is further cooled down to the injection temperature. Inside the cycle, there are pumps placed at three points. The first is the condensation pump, placed after the condenser. The second is the circulation pump, placed after the cooling tower, which recirculates part of the cooled fluid back into the direct contact condenser. The last is the blowdown pump, also located after the cooling tower, which pumps the remaining geothermal fluid back into the reservoir.

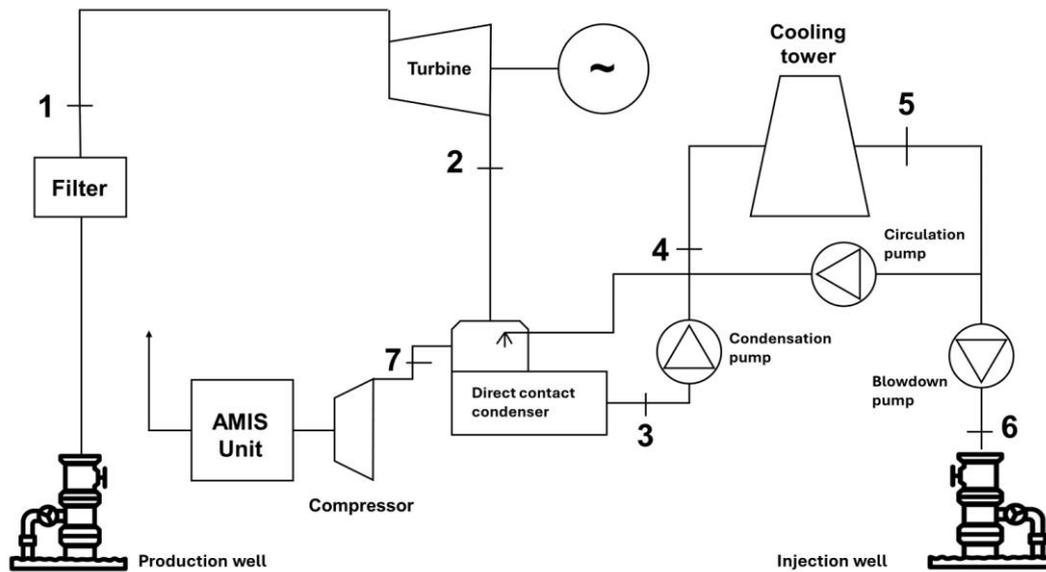


Figure 14. Dry steam power plant for electricity production reference schematic. Author's own elaboration based on the scheme presented by DiPippo (1980)[51]

Figure 14 refers to the scheme shown in DiPippo (1980)[51] with the addition of the AMIS unit, needed for the abatement of non-condensable gas (NCG) and improvement of air quality[52], as well as the inclusion of a compressor to extract these non-condensable gases. If left unexhausted, these gases can cause an increase in pressure inside the condenser, disrupting the turbine expansion process[53]. In the code, the AMIS unit is not modelled, and it does not consume electrical power; therefore, the only relevant component is the compressor. The power required by the compressor is calculated and subtracted from the total power generated.

In the code, this type of plant has an alternative configuration that allows for cogeneration, where a portion of the steam is extracted before entering the turbine and is directed to the end user. The amount of heat sold is taken into account as income in the economic model.

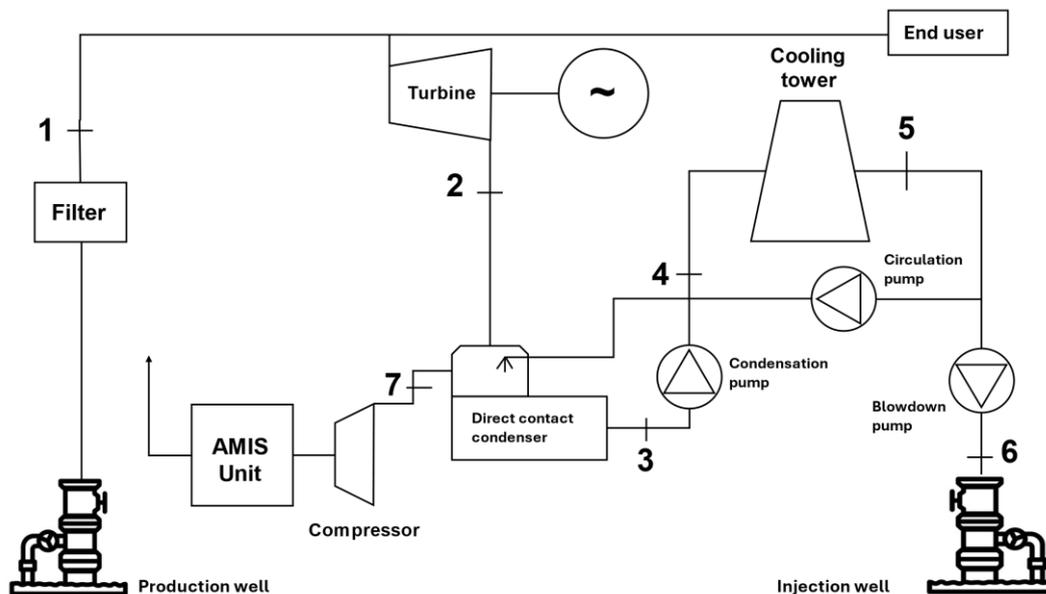


Figure 15. Dry steam power plant for cogeneration reference schematic. Author's own elaboration based on the scheme presented by DiPippo (1980)[51]

The thermodynamic cycle is well represented on the T-s diagram (Figure 16). Below there is a comparison between a theoretical thermodynamic cycle for the discussed type of plant and the one coming out from the code in one of the simulations.

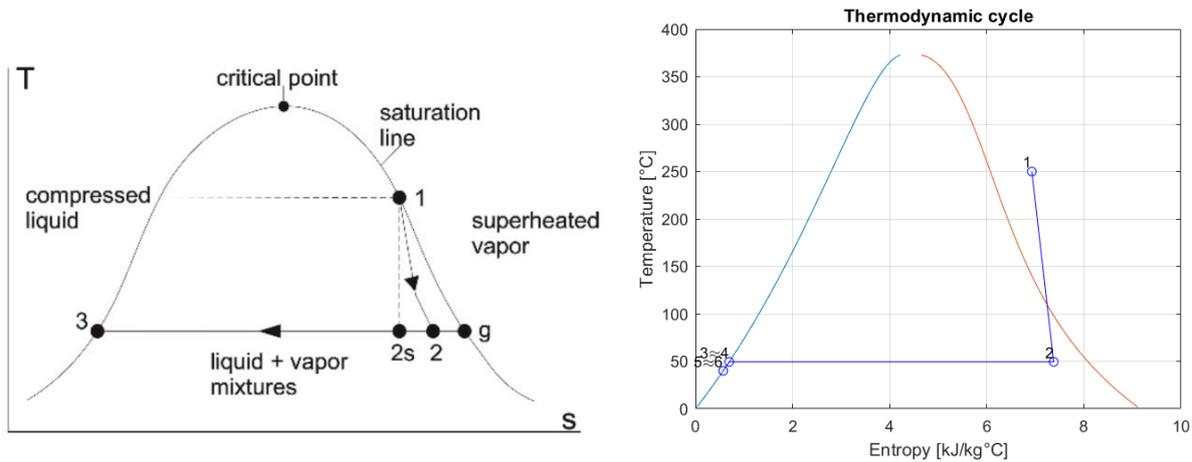


Figure 16. Thermodynamic cycle for dry steam power plant on the left (Mulyana et al. 2016,[54]). Author's own elaboration using MATLAB on the right

This simulation is not based on real values of pressure, temperature and mass flow rate. Here the only purpose is to show that the thermodynamic cycle is well represented by the code. The only differences between the two are the superheated vapour at 250°C and 10 bar, and the cooling before the injection.

4.2.2. Flash power plant

This type of plant is the most widespread worldwide, accounting for about 32% of all geothermal plants and representing 42% of the total geothermal capacity [55]. It is very similar to a dry steam power plant, with the primary difference being that the geothermal fluid is a hot, compressed liquid. When the fluid exits the well, it goes through a flasher or separator, where the pressure reduction causes it to split into saturated vapor and saturated liquid. At this point, the saturated liquid is directly injected back into the ground, while the steam expands through the turbine, following the same cycle as in a dry steam plant.

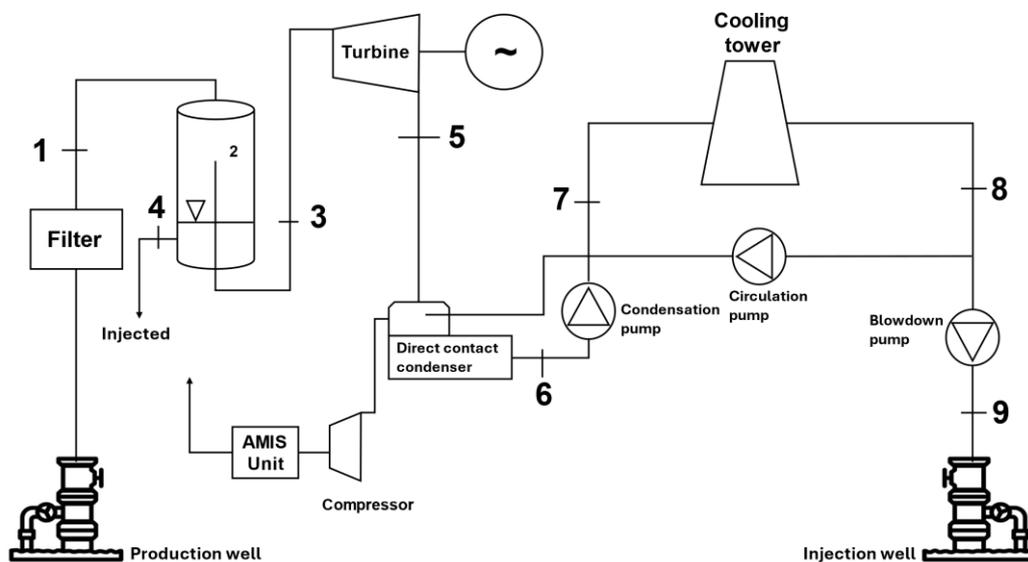


Figure 17. Flash power plant reference schematic. Author's own elaboration based on the scheme presented by DiPippo (1980)[51]

Point 2 (Figure 17) is internal to the separator, which indicates the pressure at which the geothermal fluid expands and consequently, the quality reached, usually between 10% and 50%. The plant schematic also, in this case, has in addition filter, compressor and AMIS unit, to the original one taken as a reference of DiPippo (1980)[51].

This type of plant can work with a wide range of temperatures of the geofluid, but compared to the dry steam plant the efficiency is lower. To increase the efficiency, if the fluid pressure is still relatively high,

it is possible to flash the liquid brine exiting from the first separator into a flasher and expand the steam in a low-pressure turbine. This increases the use of primary fluid, and the power produced. In the code, only one flash is considered.

As before, a comparison with the theoretical thermodynamic cycle of the geofluid inside a single-flash power plant with the ones generated through the code is reported in Figure 18.

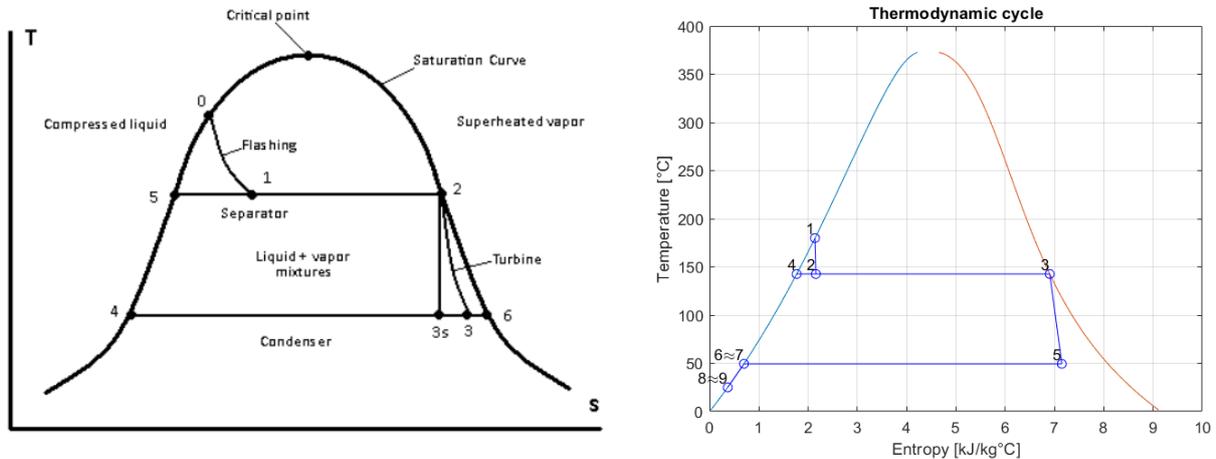


Figure 18. The thermodynamic cycle of a single flash power plant on the left (Andrès et al. 2011,[56]). Author's own elaboration using MATLAB on the right

There are minimal differences between the two graphs; the flashing process is less accentuated in the MATLAB plot, and the turbine expansion follows a similar trend. Moreover, the isentropic expansion of the turbine is not reported in the right graph. It is important to note that this is not a simulation based on the true value of temperature, pressure and mass flow rate, but rather an illustration to demonstrate how accurately the MATLAB model represents the thermodynamic cycle. However, for the simulation, the geofluid temperature is set to 180°C with a pressure of 10 bar.

4.2.3. Binary power plant

The binary cycle uses as an organic working fluid, such as isobutane, isopentane, R113, R123, etc., to perform an Organic Ranking Cycle (ORC), with the geothermal fluid acting as a source of heat to vaporize the secondary fluid before being reinjected into the ground. The brine remains in a liquid state during the cycle since it exits as a hot, compressed liquid. For most of the organic fluid, the expansion in the turbine ends in the gas phase, due to the positive slope of the saturated-vapour line. This helps to prevent possible damage to the turbine.

This type of plant is suitable for low-temperature hydrothermal resources since organic fluids have low boiling temperatures. Some other advantages are the smaller size of the turbine due to the lower output power produced, but it is good also from an economic point of view. Moreover, the high-pressure operation eliminates the need for vacuum operation, and the isentropic efficiency is higher. However, disadvantages include the high cost of the secondary working fluid, the necessity to avoid leaks, the requirement for a large flow rate of geothermal fluid (which can cause issues during the injection phase), and the heat exchanger being the largest and most costly component [51][57].

An important design choice for this plant is the type of working fluid since it can significantly affect the plant's capital cost and operational and maintenance costs. Moreover, when choosing the organic fluid, other aspects should be considered as chemical stability and compatibility with the materials, the environmental impacts and the safety concerns [58]. In the work done by Augustine et al. (2009)[59], who analysed different working fluids in subcritical and supercritical binary cycles, found that different ORCs give better performance at different temperatures. According to this study, the fluid R134a gives the best efficiency in a temperature range between 110 and 130°C, isobutane (R600a) performs better in the range of 140 to 170°C, R245fa is the best between 180-190°, and isopentane is better at the highest

temperature studied of 200°C. Successively, in the study done by Yekoladio et al. (2015)[58] is reported that the ability to convert the total energy input into useful work output is higher for isobutane compared to R152a, R123 and n-pentane. Based on these considerations it was chosen to use the isobutane as the working fluid. However, if the user wants to change it, can go to the section of the code where the physical properties are evaluated and change the name of the fluid. Note that the external function Realprop works only with the following fluids: R134a, R12, R22, ammonia, propane, carbon dioxide, R125, R32, and R600a. Be warned that by changing the working fluid the code could not work.

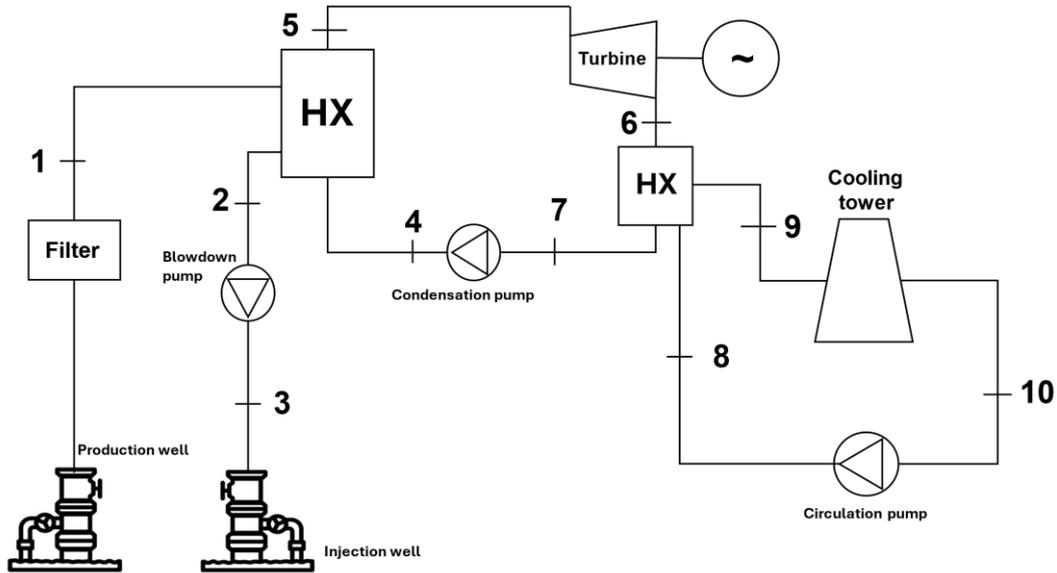


Figure 19. Binary power plant reference schematic. Author's own elaboration based on the scheme presented by DiPippo (1980)[51]

This plant is referring to DiPippo (1980)[51], also in this case there is a little modification to the original scheme. Originally the brine exiting from the heat exchanger was mixed with the make-up water used in the cooling tower to cool down the organic fluid before injection. While in the program the cooling tower has a closed loop, so the water is not injected with the geothermal fluid.

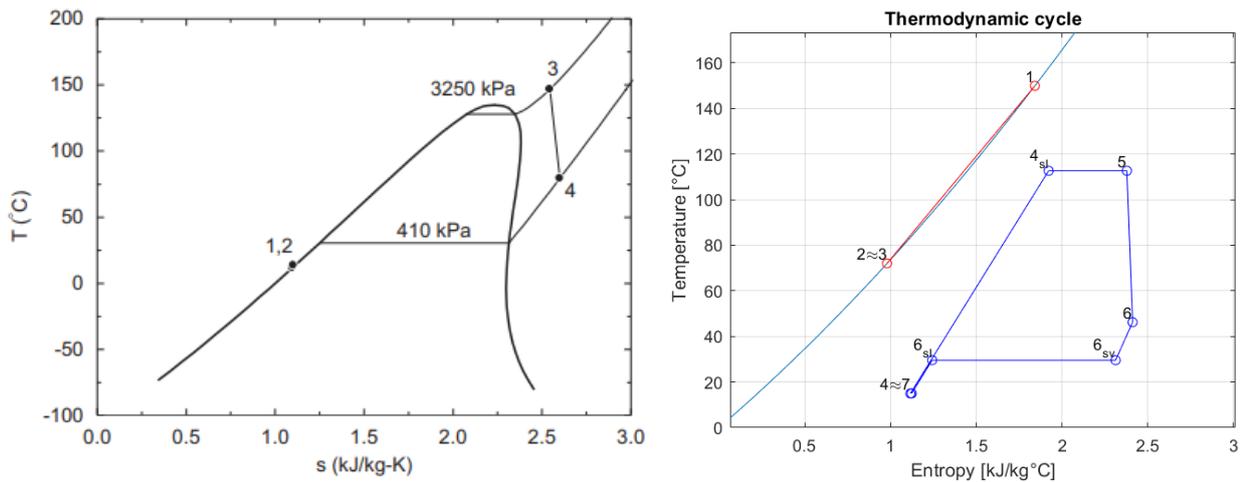


Figure 20. Binary thermodynamic cycle: theoretical on the left (Kanoglu et al. [60]). Author's own elaboration using MATLAB on the right

The modelled binary cycle is a subcritical one, in which the organic fluid expands after reaching saturated-vapour condition, differently from the right graph in Figure 20, in which the organic fluid is brought into the super-heated region before expanding.

From the MATLAB representation (Figure 20 on the left), it is possible to see in red the brine, which is cooled down in the heat exchanger, passing from point 1 to point 2. At the same time, it is possible to see the organic fluid, in this case, the isobutane, which is heated up to saturated-vapor condition in point 5. Point 4_{sl} represents the saturated-liquid conditions. Similarly, is shown in the condenser from points 6 to 7, where points 6_{sl} and 6_{sv} represent the saturated-liquid and saturated-vapor condition. In this simulation, the values of temperature and pressure, both for geothermal fluid and organic fluid, are not based on real plants. In the simulation, the brine exits from the wellhead at 8 bar and 150°C, while the organic fluid starts from a temperature of 15°C and is heated up to 112°C at a pressure of 25 bar.

4.3. Model of the components

This section reports the various models of the plant's components, detailing the formulas implemented in the code. These formulas are consistent across all plant types, with any variations specifically noted. Moreover, the formulas are written in a general form.

4.3.1. Filter unit

The filter unit is not present in the original reference map, but it is an addition to take into account the pressure losses, that change the inlet pressure in the thermodynamic cycle. In all three schematics, there is a cyclone separator which acts as a filter, and it is placed after the production well. It was chosen due to its possibility to work both with liquids and gases.

It is important to remove any solid particles from the steam in dry steam power plants to ensure the steam entering the turbine is of desired quality, while in flash and binary power plants, they are used to stop scouring or erosion in pipes and conduits [61].

The pressure losses across this type of filter are evaluated as [62]:

$$\Delta p_{filter} = \frac{1}{2g} \frac{\rho_g}{\rho_l} v^2 \Delta H \frac{98.0665}{10^5} [bar]$$

Where:

- $\frac{\rho_g}{\rho_l}$ is the ratio between densities of the brine in liquid and vapour phases [-];
- g is the gravitational acceleration $\left[\frac{m}{s^2}\right]$;
- v is the velocity of the fluid $\left[\frac{m}{s}\right]$;
- ΔH is the pressure drop in inlet velocity heads $[cmH_2O]$;
- $\frac{98.0665}{10^5}$ is the conversion factor from cmH₂O to bar $\left[\frac{Pa}{cmH_2O} \frac{bar}{Pa}\right]$.

The pressure drop in inlet velocity heads depends on geometrical design variables, and the type of inlet, but it will not be discussed here. It was calculated using different formulations in the work done by Leith and Mehta (1973)[62], and it ranges from a value of 1.8 to a value of 20.

To understand if the pressure losses introduced by this type of filter in the plant are sufficiently low to not bring any problems, like phase change in binary plants primary loop, the pressure losses were estimated within range of temperature from 150 to 300°C, a range of pressure from 5 to 30 bar, and a range of mass flow rate from 30 to 300 kg/s. A value of 20 for the pressure drop in inlet velocity heads (ΔH) was chosen. All the possible combinations between these three parameters were taken into account.

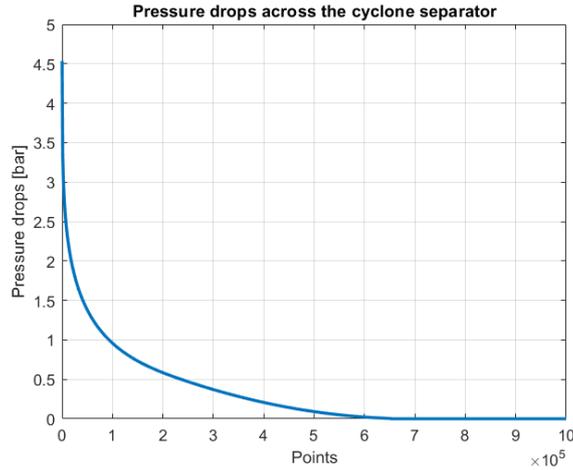


Figure 21. Pressure drops across the cyclone separator. Author's own elaboration

As it is possible to see from Figure 21 the maximum pressure drop is 4.5358 bar, when the pressure is 5 bar, the temperature is 300°C and the mass flow rate is 300 kg/s. This value is not negligible, but the number of combinations which give a pressure drop higher than 1 bar is limited to 9.135% of the total number of combinations (1 million). Generally, there are high pressure drop for low value of inlet pressure, high temperature and high mass flow rate. However considering the usual high pressure at the wellhead, they do not influence to much the thermodynamic cycle, and thus the results.

4.3.2. Turbine

The evaluation of the power generated by the turbine is the most important section of the thermodynamic cycle because it directly determines the geothermal potential. Two different approaches are covered by the reviewed codes. The method proposed by GEOPHIRE (2020) [39] starts by evaluating a parameter called “availability”, which represents the specific energy of the brine, and depends on the production, injection and ambient temperature:

$$availability = \left[(A - BT_0)(T_1 - T_2) + \frac{B - CT_0}{2}(T_1^2 - T_2^2) + \frac{C}{3}(T_1^3 - T_2^3) - AT_0 \log\left(\frac{T_1}{T_2}\right) \right] * \frac{2.2046}{947.83} \left[\frac{MJ}{kg} \right]$$

Where:

- A, B, and C are constants, equal to 4.04165, $1.204 \cdot 10^{-2}$, $1.605 \cdot 10^{-5}$;
- T_1 is the production temperature [K];
- T_2 is the injection temperature [K];
- T_0 is the ambient temperature [K];
- $\frac{2.2046}{947.83}$ are conversion coefficients.

The availability is then multiplied for the utilization efficiency (η_u) and the produced mass flow rate to find the power:

$$MW_{e,prod} = availability \cdot \eta_u \cdot \dot{m}_{prod}$$

This method often results in power values that are either overestimated or underestimated.

Therefore, the method used by genGEO (2020) [38] was chosen as a guideline, which evaluates the power as product between mass flow rate and difference in enthalpy.

In the code, the physical properties at the turbine inlet are known, and to find the power produced by the turbine the two parameters that the user must set are the isentropic efficiency of the turbine (η_{turb}) and the outlet pressure.

Knowing the outlet pressure, it is possible to evaluate the isentropic enthalpy of the outlet point, using the X Steam function [48], giving as input the outlet pressure and the inlet entropy. Then using the efficiency, it is possible to find the outlet enthalpy:

$$h_{out,s} = XSteam(p_{out}, s_{in}) \left[\frac{kJ}{kg} \right]$$

$$h_{out} = h_{in} - \eta_{turb} (h_{in} - h_{out,s}) \left[\frac{kJ}{kg} \right]$$

Knowing the outlet enthalpy, it is possible to evaluate the other physical properties always using the external function X Steam. One important property in this phase is the vapour fraction, which must be higher than a set threshold to not damage the turbine through the droplets of condensed water.

There is a check on this value: in case it is lower than the limit, the code stops running and returns a net power produced equal to zero. In order to comply with the limit is suggested to increase the outlet pressure of the turbine, because by doing it the steam expands less, and consequentially the vapour fraction is higher.

Then, the specific work of the turbine is calculated as the difference between the inlet and outlet enthalpies:

$$w_{turb} = h_{in} - h_{out} \left[\frac{kJ}{kg} \right]$$

The power produced by the turbine is calculated as:

$$\dot{W}_{turb} = \frac{w_{turb} \cdot \dot{m}_{in}}{1000} [MW]$$

Finally, the generated power is computed considering the efficiency of the generator (η_{gen}):

$$\dot{W}_{gen} = \dot{W}_{turb} \eta_{gen} [MW]$$

In the case of a binary cycle, the only difference is the use of an external function that evaluates the physical properties, in this case is Realprop[49]. A problem in using this external function is that is not possible to exactly know the vapour fraction of organic fluid, and check if it is under the threshold limit. To overcome this problem when there is control on the vapour fraction, if it is not only vapour, the code stops and returns zero power produced, but unlike what happens for water, it is suggested to reduce the outlet pressure, because of the positive slope in temperature entropy diagram.

4.3.3. Direct contact condenser

The direct contact condenser is used inside the cycle of both dry steam and single flash power plants, and it is placed at the turbine outlet, as shown in Figure 14 and Figure 17. In contrast, a binary power plant employs a heat exchanger because the organic fluid must be in a closed loop. In this type of condenser, the steam exiting the turbine is cooled down by contact with the sprayed water from the top of the condenser. The water arrives from the outlet of the cooling tower and is at injection temperature. One constraint is that the injection temperature cannot be higher than the temperature at the turbine outlet. This may happen because the pressure at the turbine outlet is usually low, leading to a reduced temperature. For example, referring to Figure 16, the turbine inlet temperature is 250 °C, while the outlet one is 49.42 °C. If the set injection temperature is higher, the code will return zero power output.

The amount of water needed to condense the steam into a saturated liquid is evaluated using mass and energy balance equations, as the inlet and outlet enthalpies can be easily calculated through the X Steam external function.

However, two assumptions are made:

1. the temperature in the condenser remains constant, and at the outlet, the fluid is on the saturated-liquid curve;
2. The sprayed water from the cooling tower outlet is supposed to be on the saturated-liquid curve.

Using these two assumptions, the enthalpies can be evaluated as a function of temperature:

$$h_{out,cond} = XSteam(T_{out,turb}) \left[\frac{kJ}{kg} \right]$$

$$h_{ws} = X_{Steam}(T_{inj}) \left[\frac{kJ}{kg} \right]$$

The mass and energy balance equations also account for the presence of non-condensable gases:

$$\begin{aligned} \dot{m}_{out,turb}(1 - X_{ncg}) + \dot{m}_{ws} &= \dot{m}_{out,cond} \\ \dot{m}_{out,turb}(h_{out,tubr} - h_{out,cond}(1 - X_{ncg})) &= \dot{m}_{ws}(h_{out,cond} - h_{ws}) \end{aligned}$$

Where:

- $\dot{m}_{out,turb}$ is the mass flow rate at the turbine outlet $\left[\frac{kg}{s} \right]$;
- \dot{m}_{ws} is the sprayed mass flow of water $\left[\frac{kg}{s} \right]$;
- $\dot{m}_{out,cond}$ is the mass flow rate at the condenser outlet $\left[\frac{kg}{s} \right]$;
- $h_{out,tubr}$ is the enthalpy at the turbine outlet $\left[\frac{kJ}{kg} \right]$;
- $h_{out,cond}$ is the enthalpy at the condenser outlet $\left[\frac{kJ}{kg} \right]$;
- h_{ws} is the enthalpy of the sprayed water $\left[\frac{kJ}{kg} \right]$;
- X_{ncg} is the fraction of non-condensable gases mixed with the steam [-].

Moreover, the heat removed from the condenser is calculated in the code as:

$$\dot{Q}_{cond} = \frac{\dot{m}_{out,turb} \cdot (h_{out,tubr} - h_{out,cond}(1 - X_{ncg}))}{1000} [MW_{th}]$$

The presence of non-condensable gases (NCG) has minimal influence on the results. Referring to the example reported in Figure 16, the heat removed from the condenser considering NCG, increases only by 1 MW_{th}, the sprayed mass flow rate increases by 100 kg/s, and the circulation pump power increases by 0.019 MW_e. Note that in the example the power produced by the generator is equal to 56.41 MW_e.

4.3.4. Compressor

The compressor is required to extract the non-condensable gases from the direct contact condenser since can be present within the geothermal fluid. The mass flow rate of NCG is a fraction of the mass flow rate that expands in the turbine. This fraction can either be chosen by the user or set to a default value of 5%. This component is not present in the binary cycle since a direct contact condenser is not used. To evaluate the power required by the compressor, it is needed to set the composition of the non-condensable gases. Usually, carbon dioxide (CO₂) is the gas with a higher percentage, followed by hydrogen sulphide (H₂S), and in low percentage mercury (Hg), which sometimes are not even present[63],[64].

The properties of the gas mixture are evaluated simply by calculating the weighted sum of each gas's properties based on their volumetric percentages. The properties which are required for the evaluation of the compressor power are the specific heat capacity at constant pressure (c_p), the specific heat capacity at constant volume (c_v), and the gas constant of the mixture.

The following equations are implemented:

$$R_{ncg} = \sum_{i=1}^n x_i \cdot \frac{R}{MM_i} \cdot 1000 \left[\frac{J}{kgK} \right]$$

Where:

- R is the universal gas constant, equal to 8.314 $\left[\frac{J}{molK} \right]$;
- MM_i is the molar mass of the gas species $\left[\frac{g}{mol} \right]$;
- x_i is the volumetric percentage of the species inside the mixture [-];
- 1000 is the conversion factor to pass from gram to kilogram $\left[\frac{g}{kg} \right]$.

$$c_{p,ncg} = \sum_{i=1}^n x_i \cdot c_{p,i} \left[\frac{kJ}{kgK} \right]$$

Where:

- x_i is the volumetric percentage of the species inside the mixture [-];
- $c_{p,i}$ is the specific heat at constant pressure of the gas species $\left[\frac{kJ}{kgK} \right]$.

$$c_{v,ncg} = \sum_{i=1}^n x_i \cdot c_{v,i} \left[\frac{kJ}{kgK} \right]$$

Where:

- x_i is the volumetric percentage of the species inside the mixture [-];
- $c_{v,i}$ is the specific heat at constant volume of the gas species $\left[\frac{kJ}{kgK} \right]$.

Knowing the specific heat at constant pressure and at constant volume it is possible to calculate the isentropic exponent (n_{ncg}):

$$n_{ncg} = \frac{c_{p,ncg}}{c_{v,ncg}} [-]$$

Then it is possible to evaluate the power required by the compressor to extract the gases [65]:

$$\dot{W}_{compressor} = \dot{m}_{ncg} R_{ncg} T_{ncg} \frac{n_{ncg}}{n_{ncg} - 1} \frac{1}{\eta_{compressor}} \left[\left(\frac{p_{out,compr}}{p_{condenser}} \right)^{\frac{n_{ncg}-1}{n_{ncg}}} - 1 \right] \cdot \frac{1}{10^6} [MW_e]$$

Where:

- T_{ncg} is the temperature at which the non-condensable gases are in the direct contact condenser, which is the same as the steam exiting from the turbine [K];
- $\eta_{compressor}$ is the isentropic efficiency of the compressor [-];
- $p_{condenser}$ is the pressure inside the direct contact condenser, which is the same as the one at the turbine outlet [bar];
- $p_{out,compr}$ is the pressure at the compressor outlet [bar];
- $\frac{1}{10^6}$ is the conversion coefficient $\left[\frac{MW}{W} \right]$;
- \dot{m}_{ncg} is the mass flow rate of non-condensable gases $\left[\frac{kg}{s} \right]$.

The value of $p_{out,compr}$ is by default set to ambient pressure (p_0), but the user can change it. Similarly, the NCG percentage can be changed or event set to 0, effectively eliminating their presence. Doing this, the power required by the compressor will be set to zero, using a conditional statement that checks the value of n_{ncg} . If NCGs are not present, n_{ncg} will be "Not a Number" (NaN).

4.3.5. Pumps

Inside the thermodynamic cycle are located some pumps, to increase the pressure of the fluid or to simply make it circulate inside the plant. The power required by each one is evaluated as [66]:

$$\dot{W}_{pump} = \frac{\dot{m}}{\rho} \Delta p \cdot 10^5 \cdot \frac{1}{\eta_{pump}} \cdot \frac{3600}{3.6 \cdot 10^6} \cdot \frac{1}{1000} [MW]$$

Where:

- \dot{m} is the mass flow rate circulating through the pump $\left[\frac{kg}{s} \right]$;
- ρ is the fluid's density $\left[\frac{kg}{m^3} \right]$;
- Δp is the pressure difference between the two points [bar];
- 10^5 is the conversion factor to change the bar in Pa $\left[\frac{Pa}{bar} \right]$

- η_{pump} is the efficiency of the pump[-];
- 3600 are the second in one hour $\left[\frac{s}{h}\right]$;
- $\frac{1}{1000}$ is the conversion factor from kW to MW $\left[\frac{MW}{kW}\right]$;
- $\frac{1}{3.6 \cdot 10^6}$ is the conversion coefficient to have a kW $\left[\frac{Nm}{s \cdot kW}\right]$.

This formulation was chosen instead of the following one:

$$\dot{W}_{pump} = \dot{m} \Delta h \cdot \frac{1}{1000} [MW]$$

Where:

- \dot{m} is the mass flow rate circulating through the pump $\left[\frac{kg}{s}\right]$;
- Δh is the enthalpy difference between the pump inlet and outlet $\left[\frac{kJ}{kg}\right]$;
- $\frac{1}{1000}$ is the conversion factor from kW to MW $\left[\frac{MW}{kW}\right]$.

The latter was firstly implemented because is easy to find the enthalpies by using the function X Steam [48] but it was found that it underestimates the pump's power required.

In this formulation the pump efficiency appears in the evaluation of the exit enthalpy, firstly evaluating the isentropic enthalpy at the pump outlet.

To choose between these two, they were compared with the pump power required in Aspen Plus[67], that is a chemical process simulator based on flowsheet simulation, setting the same condition at the inlet. The results are in the table below:

H ₂ O	\dot{W}_{pump} [MW]	ϵ_{rel} [%]
Aspen Plus	0,385	0
$\frac{\dot{m}}{\rho} \Delta p \cdot k_1$	0,377	1,95
$\dot{m} \Delta h \cdot k_2$	0,29	24,58

Table 3. Pump formula error respect Aspen Plus for water (k_1 and k_2 are coefficients). Author's own elaboration using the results from Aspen Plus and MATLAB

R600a	\dot{W}_{pump} [MW]	ϵ_{rel} [%]
Aspen Plus	0,393	0
$\frac{\dot{m}}{\rho} \Delta p \cdot k_1$	0,363	7,63
$\dot{m} \Delta h \cdot k_2$	0,132	66,41

Table 4. Pump formula error respect Aspen plus for isobutane (k_1 and k_2 are coefficients). Author's own elaboration using the results from Aspen Plus and MATLAB

As shown in Table 3 and Table 4, the error introduced by the first formula is significantly lower than that introduced by the second formula, for both water or isobutane used in the binary cycle. Moreover, the difference is mainly due to fluid density in both simulations, which explains why the error is larger in the case of isobutane. In the end, the first formula is implemented in the code.

The blowdown pump is most power-consuming of the three pumps, since it needs to raise the pressure of the geothermal fluid to at least the hydrostatic pressure at given depth:

$$p_{injection} = \frac{\rho g z}{10^5} [bar]$$

Where:

- ρ is the density of the fluid before the injection $\left[\frac{kg}{m^3}\right]$;
- g is the gravity constant $\left[\frac{m}{s^2}\right]$;

- z is the depth at which the fluid is injected [m];
- $\frac{1}{10^5}$ is the conversion factor to pass from Pa to bar.

In dry steam and single flash power plants, the condensation pump increases the pressure by 1.5 bar, while the circulation pump increases the pressure by 1 bar. In the case of binary power plant, the condensation pump increases the fluid pressure up to the set value, while circulation pump restores the pressure to the original one in the cooling tower loop. This is done under the assumption that there is a pressure drop equal to 3% of the inlet pressure in the heat exchanger, as suggested by Kazemi et al. (2018) [68].

4.3.6. Cooling tower

The cooling tower is not specifically modelled inside the code, since the main interest is in the evaluation of the power required by the fans to remove the heat from the geothermal fluid, in dry steam and flash power plant, or from water, in the binary cycle.

The power required by the fans is calculated using a coefficient that directly converts the thermic power into electric power. It depends on the type of fans implemented inside the power plant: for centrifugal fans, it is equal to $40 \frac{kW_{th}}{kW_e}$, while for axial fans, it is equal to $80 \frac{kW_{th}}{kW_e}$ [69].

The pressure and temperature at the inlet and outlet of the cooling tower are known, allowing for the enthalpy to be calculated using the external function X Steam, assuming negligible pressure losses across the cooling tower.

The amount of heat that must be removed can be determined from the enthalpies:

$$\dot{Q}_{cooling,tower} = \frac{\dot{m}_{in}(h_{in} - h_{out})}{1000} [MW_{th}]$$

Where:

- \dot{m}_{in} is the mass flow rate cooled down inside the cooling tower $\left[\frac{kg}{s}\right]$;
- h_{in} is the enthalpy at the cooling tower inlet $\left[\frac{kJ}{kg}\right]$;
- h_{out} is the enthalpy at the cooling tower outlet $\left[\frac{kJ}{kg}\right]$;
- $\frac{1}{1000} \left[\frac{MW}{kW}\right]$ is the conversion coefficient.

From the heat extracted, the fans power is:

$$\dot{W}_{tower,fan} = \frac{\dot{Q}_{cooling,tower}}{coeff} [MW_e]$$

where the coefficient is one of the two values reported before.

4.3.7. Heat exchanger

The heat exchanger (HX) is a component used to transfer heat from a hot fluid to a colder one, preventing them from being in contact. It is present only in the binary cycle, where two heat exchangers are used: the first one heats the organic fluid from its initial temperature to the evaporation point, and the second, placed after expansion in the turbine, cools the working fluid back to its starting state.

The heat exchange between fluids is modelled by applying the laws of mass and energy conservation, without considering any geometrical parameter, except for the heat exchange area, which is needed for the economic evaluation.

A counter-current heat exchanger is considered, since its higher efficiency respect to the co-current heat exchanger.

4.3.7.1. Heat transfer efficiency

The efficiency of the process is evaluated using the ε -NTU method, while the heat exchanger area is evaluated using the mean logarithmic temperature difference method[70].

The effectiveness (ε) is calculated using a graphical method, under the assumption of a high number of transfer units (NTU). This is fundamental since it is a function of the heat exchange area (A), which has been evaluated lately:

$$NTU = \frac{AU}{C_{min}} [-]$$

Where:

- A is the heat exchange area [m^2];
- U is the global heat transfer coefficient [$\frac{W}{m^2K}$];
- C_{min} is the minimum between two fluid's thermal capacities [$\frac{W}{K}$].

The value of the parameter U is set as default equal to $500 \left[\frac{W}{m^2K} \right]$, because as reported by Peters et al. (2003)[71], it ranges from $375-750 \left[\frac{W}{m^2K} \right]$ for heat transfer between light-organic fluid and water inside the condenser, and it ranges from $500-1000 \left[\frac{W}{m^2K} \right]$ inside the evaporator. With this set value, the size of the first heat exchanger might be overestimated, increasing the cost of this component and reducing economic profits, representing the worst-case scenario. For what concern the condenser, which has the larger heat exchange area, this set value prevents both underestimation or overestimation size. The only parameter required to use the graphic method is the ratio between the fluid's thermal capacities:

$$C = \frac{C_{min}}{C_{max}} [-]$$

Where C_{min} and C_{max} are the minimum and the maximum between the calculated fluid's thermal capacity. Usually, the minimum thermal capacity comes from the working fluid, since it has a lower mass flow rate (\dot{m}) and lower specific thermal capacity at constant pressure (c_p)[70]. The thermal capacity is generally evaluated as:

$$C_{fluid} = \dot{m}_{fluid} c_{p,fluid}$$

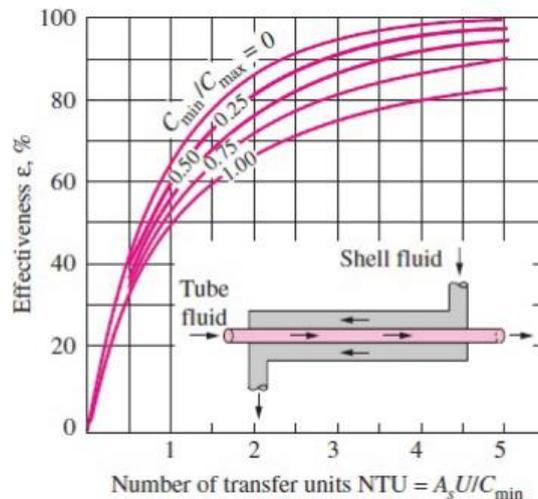


Figure 22. Effectiveness as a function of NTU for counter-current heat exchanger (Ezci (2017),[72])

As seen in Figure 22, at high NTU values, the effectiveness is constant. In the evaporator, the C-ratio is close to zero, so a series of conditional statements is implemented to divide its value in the bands shown in Figure 22. The efficiency is set as the same as the one corresponding to the higher C-ratio inside the band. For example, if the C-ratio is equal to 0.4, which falls into the range between 0.25 and 0.5, the efficiency is set according to the 0.5 line.

In the condenser, due to the high mass flow rate of water required to prevent significant temperature increment, the C-ratio tends to zero, and with a high NTU value, the efficiency is set at 0.99.

```

Cmin=min(max(m1.*cp1),max(m4.*cp4)); %[kW/°C]
Cmax=max(min(m1.*cp1),min(m4.*cp4)); %[kW/°C]
Cratio=Cmin/Cmax;
if Cratio<0.25
    eps_hx=0.97;
else if Cratio>=0.25 && Cratio<0.5
    eps_hx=0.95;
else if Cratio>=0.5 && Cratio<0.75
    eps_hx=0.90;
else
    eps_hx=0.80;
end
end
end

```

Figure 23. Conditional statement to choose the effectiveness. Author's own elaboration using MATLAB

To evaluate the C_{min} and C_{max} , given that mass flow rate and specific heat capacity at constant pressure change over time, it was decided to find the maximum and minimum values of the vectors before finding the minimum and maximum between them. It was done to have the highest C-ratio, and so to be in the worst-case scenario.

4.3.7.2. Evaporator

To model this component, it is important to determine the mass flow rate of organic fluid, ensuring that a pinch point lower than a set threshold is not reached. To evaluate the mass flow rate, a power balance inside the heat exchanger is performed, considering the heat exchanged during the evaporation phase (\dot{Q}_x). Since isobutane is changing phase, a difference in enthalpy must be used, while for the geothermal fluid, a formulation that emphasises the temperature difference is applied:

$$\dot{Q}_x = \dot{m}_{r600a} (h_{evap,out} - h_{evap,in}) [kW]$$

$$\dot{Q}_x = \dot{m}_1 c_{p1} (T_1 - T_x) [kW]$$

Where:

- $h_{evap,out}$ and $h_{evap,in}$ are respectively the enthalpy at which the evaporation of the isobutane ends and starts $\left[\frac{kJ}{kg}\right]$;
- \dot{m}_{r600a} and \dot{m}_1 are the mass flow rate of isobutane and geothermal fluid $\left[\frac{kg}{s}\right]$;
- T_1 and T_x are the temperature of the geofluid entering the heat exchanger and the one after \dot{Q}_x is exchanged $[^{\circ}C]$.

The enthalpies can be found through the Realprop function after selecting the desired pressure. The only other unknown, excluding the mass flow rate of the organic fluid, is the temperature T_x of the brine at that point. The temperature T_x can be rewritten as a function of the evaporation temperature (T_{evap}) of the working fluid, which can be found using Realprop, and the minimum temperature difference (ΔT_{pp}) to avoid a pinch point inside the HX:

$$T_x = T_{evap} + \Delta T_{pp} [^{\circ}C]$$

Then it is possible to find the mass flow rate of isobutane:

$$\dot{m}_{r600a} = \frac{\dot{m}_1 c_{p1} [T_1 - (T_{evap} + \Delta T_{pp})]}{(h_{evap,out} - h_{evap,in})} \left[\frac{kg}{s}\right]$$

Two considerations must be done: the first is that this formulation does not consider the efficiency of the heat exchange process, which will bring to a higher mass flow rate value. However, to evaluate that parameter, the mass flow rate itself is required. The second consideration regards the parameters entity \dot{m}_1 , c_{p1} and T_1 . They are parameters that change over time, so the calculation is done for the last year of

operation of the hypothetic plant since their values decrease in time, especially mass flow rate and temperature.

In the cumulative heat exchange versus temperature graph, the slope is proportional to the inverse of the product between mass flow rate and specific heat capacity at constant pressure [73]:

$$\frac{\Delta T}{\dot{Q}} = \frac{1}{\dot{m}_1 c_{p1}} \left[\frac{K}{kW} \right]$$

Over the years the slope becomes steeper since $\dot{m}_1 c_{p1}$ decreases. If the working fluid mass flow is evaluated during the first year, there may be a temperature crossover at some point, since the slope of the organic fluid remains constant over time due to the unchanging mass flow rate.

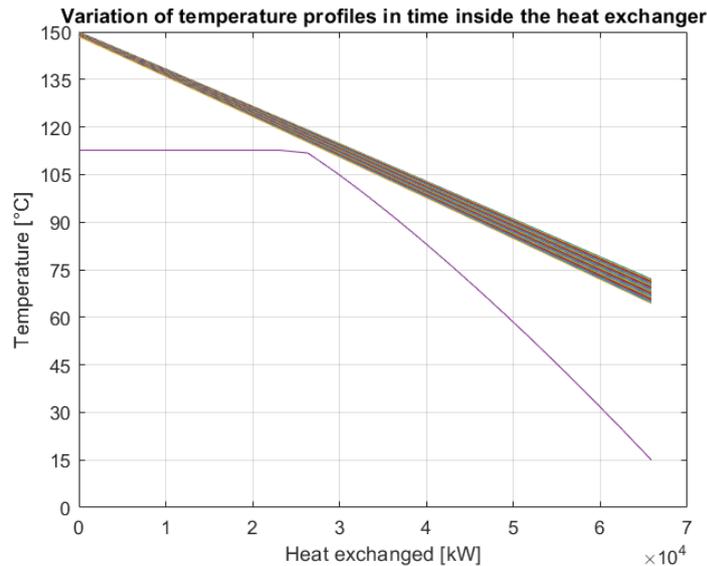


Figure 24. Slope variation of the geofluid temperature profile inside the heat exchanger. Author's own elaboration using MATLAB

The phenomenon described above is well visible in Figure 24, where the purple line refers to the organic fluid, while the other lines correspond to the geothermal fluid. The one representing the first year is the top line. It is noticeable that the minimum temperature difference in the first year is greater than the set minimum pinch point temperature. Only in the last years does the minimum pinch point temperature reach the value set, or approach it, as the heat exchange process, computed after the evaluation of the organic mass flow rate, considers the efficiency.

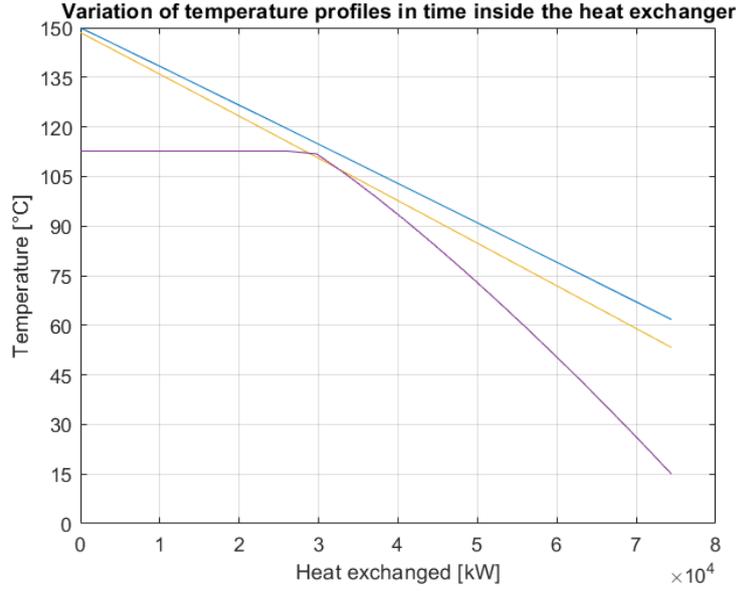


Figure 25. Crossover temperature in the heat exchanger at year 30 with wrongly evaluated organic fluid's mass flow rate. Author's own elaboration using MATLAB

Figure 25 reports the scenario in which the organic mass flow rate is evaluated considering the geothermal fluid parameters in the first year. It is possible to see that in the last year the line of the geothermal fluid (yellow), intersects the ones of the organic fluid (purple). Both the simulations in the two figures were done with the same parameters, the only difference is the year with which the isobutane mass flow rate is calculated.

The heat exchange process is modelled based on the laws of conservation of mass and energy in the evaporator, which are:

$$\begin{aligned}\dot{m}_1 &= \dot{m}_2 \\ \dot{m}_4 &= \dot{m}_5 \\ \dot{m}_1(h_1 - h_2)\varepsilon_{hx} &= \dot{m}_4(h_5 - h_4)\end{aligned}$$

The subscripts refer to the binary plant layout in Figure 19.

The unknown is the enthalpy at the outlet of the hot side (geothermal fluid), and so is its temperature. The outlet conditions of the working fluid are known once the pressure at the turbine inlet is defined since it is needed to find the enthalpy on the saturated-vapor line, which is done through the external Realprop function, and then the heat required by the cold side (organic fluid) is calculated:

$$\begin{aligned}h_{vap} &= \text{realprop}(p, x = 1) \left[\frac{kJ}{kg} \right] \\ \dot{Q}_{evap} &= \dot{m}_{r600a}(h_{vap} - h_{in}) [kW]\end{aligned}$$

Where:

- \dot{Q}_{evap} is the thermal power required to vaporize the working fluid [kW];
- \dot{m}_{r600a} is the mass flow rate of the organic fluid $\left[\frac{kg}{s} \right]$;
- h_{vap} is the enthalpy of the fluid evaluated in function of pressure and fraction of vapour (x) $\left[\frac{kJ}{kg} \right]$;
- h_{in} is the inlet enthalpy of the fluid inside the heat exchanger $\left[\frac{kJ}{kg} \right]$.

The exchanged heat (\dot{Q}_{exch}) inside the HX is a little bit higher since the efficiency in the process is not 100%, and it is:

$$\dot{Q}_{exch} = \frac{\dot{Q}_{evap}}{\varepsilon_{hx}} [kW]$$

To verify that inside the HX there is not a cross-over between temperatures, it has been discretized in 20 elements, called “increment”, and the heat transfer between each of them is:

$$d\dot{Q} = \frac{\dot{Q}_{evap}}{increment}$$

This number of “increment” was chosen to well represent the kink due to the phase change, shown by the purple line in Figure 24 and Figure 25.

The process to evaluate the evolution of temperatures and enthalpies of both fluids is like the one used in genGEO[38], but a little bit simplified. In the proposed code for each increment is calculated firstly the enthalpy of each fluid, then the temperature calling one of the two external functions:

$$\begin{aligned} h_{1,in}(i) &= h_{1,in}(i-1) - \frac{d\dot{Q}}{\varepsilon_{hx} m_1} \\ h_{4,in}(i) &= h_{4,in}(i-1) - \frac{d\dot{Q}}{m_4} \\ T_{1,in}(i) &= XSteam(p_1, h_{1,in}(i)) \\ T_{4,in}(i) &= realprop(p_4, h_{4,in}(i)) \end{aligned}$$

The index “i” represents the position inside the vector, and the element in position one of each vector is initialized as inlet enthalpy or temperature. At the end of the cycle, the outlet conditions are saved:

$$\begin{aligned} h_{1,in}(end) &= h_2; \quad h_{4,in}(end) = h_5 \\ T_{1,in}(end) &= T_2; \quad T_{4,in}(end) = T_5 \end{aligned}$$

At the end, a check is done to confirm that there is no crossover between temperatures. In case there is, the net power (\dot{W}_{net}) is set equal to zero and the code stops. It means that with the condition of the geothermal fluid is not possible to produce power.

4.3.7.3. Condenser

The condenser is modelled following the same procedure as the evaporator, but differently from how it is done in the previous component, the temperature difference between the isobutane outlet and water inlet is fixed. As default is set equal to 5°C, but it could be chosen by the user. This parameter is used to evaluate firstly the inlet temperature of water, from which all the physical properties are calculated through XSteam. Then it is used to evaluate the mass flow rate of water:

$$\begin{aligned} T_8 &= T_7 - \Delta T_{cooling,water} [^\circ C] \\ \dot{m}_8 &= \frac{\dot{Q}_{cooling}}{c_{p,8} \Delta T_{cooling,water}} \left[\frac{kg}{s} \right] \end{aligned}$$

The heat needed for the condensation is evaluated using the following formulation:

$$\dot{Q}_{cond} = \dot{m}_6 (h_6 - h_{cond}) [kW]$$

The parameter h_{cond} is the enthalpy at the end of the cooling process, not the one to reach the complete condensation of the working fluid. It has been evaluated using Realprop and giving as input the temperature of point 4, which is the starting point, and the pressure exiting from the turbine.

The energy conservation law applied to the condenser is:

$$\dot{m}_6 (h_6 - h_7) = \frac{\dot{m}_8 (h_9 - h_8)}{\varepsilon_{cond}}$$

Since there is the need to have the same temperature as the starting point, the losses are taken into account as heat not arriving at the cooling liquid, while before being taken into account in the geothermal fluid as it was cooled more to provide the required amount of heat to the isobutane.

Then the following steps are applied to the evaluation of the enthalpies at each point of the heat exchange process:

$$d\dot{Q}_{cond} = \frac{\dot{Q}_{cond}}{increment}$$

$$h_{6,in}(i) = h_{6,in}(i-1) - \frac{d\dot{Q}_{cond}}{\dot{m}_6} \left[\frac{kJ}{kg} \right]$$

$$h_{8,in}(i) = h_{8,in}(i-1) - \frac{d\dot{Q}_{cond}}{\varepsilon_{cond} \dot{m}_8} \left[\frac{kJ}{kg} \right]$$

$$T_{8,in}(i) = XSteam(p_8, h_{8,in}(i))$$

$$T_{6,in}(i) = realprop(p_6, h_{6,in}(i))$$

As before at the end of the cycle these parameters are saved:

$$h_{6,in}(end) = h_7; h_{8,in}(end) = h_9$$

$$T_{6,in}(end) = T_7; T_{8,in}(end) = T_9$$

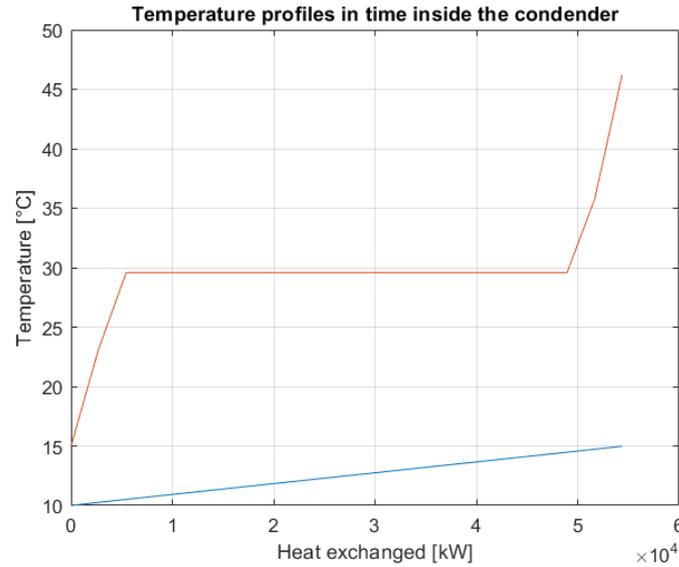


Figure 26. Example of temperature profile inside the condenser. Author's own elaboration using MATLAB

Figure 26 shows the temperature profile of the two fluids, in orange the isobutane and in blue the cooling water.

4.7.3.4. Area of heat exchange

The area required by the two heat exchangers is evaluated after each cycle, when all the temperatures are known, it is evaluated through the mean logarithmic temperature difference:

$$A_{hx} = \frac{\dot{Q}}{U \Delta T_{mlog}} \quad [m^2]$$

$$\Delta T_{mlog} = \frac{(T_{hot,in} - T_{cold,out}) - (T_{hot,out} - T_{cold,in})}{\log\left(\frac{T_{hot,in} - T_{cold,out}}{T_{hot,out} - T_{cold,in}}\right)} \quad [^{\circ}C]$$

Where:

- \dot{Q} is the heat exchanged during the process [W];
- U is the global heat transfer coefficient $\left[\frac{W}{m^2K} \right]$;
- $T_{hot,in}$ and $T_{hot,out}$ are the inlet and outlet temperature of the hot fluid ($T_{hot,in} > T_{hot,out}$) [$^{\circ}C$];
- $T_{cold,in}$ and $T_{cold,out}$ are the inlet and outlet temperature of the cold fluid ($T_{cold,in} < T_{cold,out}$) [$^{\circ}C$].

Since some parameters change with time, also the area required for the heat exchange varies in time, so in the sizing procedure, it is taken the maximum heat exchange area.

4.3.8. Separator

The separator is a component only used inside the single flash power plant, as it is possible to see in the schematic Figure 17. Its role is to allow the geothermal fluid to expand at a lower pressure, producing the steam that will expand in the turbine, while the quantity of fluid which remains liquid is injected. The pressure reduction happens thanks to an expansion valve, which is not represented, and it is also modelled with the separator. So, the separator is more generally a flashing device, as it is modelled by Wang et al. (2015)[74]. The pressure at which the brine expands is set by the user, so using the external function X Steam it is possible to find the temperature of both liquid and vapour phases after the expansion, considering the enthalpy constant through this process, and so equal to the ones of the geothermal fluid:

$$T_{sep} = XSteam(p_{sep}, h_{in}) [^{\circ}C]$$

For this model, the laws of mass conservation and energy conservation are:

$$\begin{aligned} \dot{m}_{in} &= \dot{m}_l + \dot{m}_v \\ \dot{m}_{in} h_{in} &= \dot{m}_l h_l + \dot{m}_v h_v \end{aligned}$$

Where:

- \dot{m}_{in} is the mass flow rate of the geothermal fluid $\left[\frac{kg}{s}\right]$;
- \dot{m}_l is the mass flow rate of liquid at the device's outlet $\left[\frac{kg}{s}\right]$;
- \dot{m}_v is the mass flow rate of the vapour at the device's outlet $\left[\frac{kg}{s}\right]$;
- h_{in} is the enthalpy at the separator's inlet $\left[\frac{kJ}{kg}\right]$;
- h_l is the enthalpy of the liquid phase at the separator's outlet $\left[\frac{kJ}{kg}\right]$;
- h_v is the enthalpy of the vapor phase at the separator's outlet $\left[\frac{kJ}{kg}\right]$.

The enthalpies of liquid and vapour phases are estimated using X Steam:

$$\begin{aligned} h_l &= XSteam(p_{sep}) \left[\frac{kJ}{kg}\right] \\ h_v &= XSteam('p_{sep}) \left[\frac{kJ}{kg}\right] \end{aligned}$$

The two variables that are possible to calculate using these formulas are the mass flow rates for the liquid and vapour phases since it is required to know the mass flow rate which expands in the turbine:

$$\begin{aligned} \dot{m}_l &= \dot{m}_{in} \frac{h_v - h_{in}}{h_v - h_l} \left[\frac{kg}{s}\right] \\ \dot{m}_v &= \dot{m}_{in} \frac{h_{in} - h_l}{h_v - h_l} \left[\frac{kg}{s}\right] \end{aligned}$$

4.4. Performance evaluation parameters

The performance of the thermodynamic cycle can be estimated using different parameters, which focus on different aspects of the cycle. One of them is the thermal efficiency (η_{th}):

$$\eta_{th} = \frac{\dot{W}_{net}}{\dot{Q}} [-]$$

Where:

- \dot{W}_{net} is the net power produced by the working fluid [MW_e];
- \dot{Q} is the rate at which heat is supplied to the working fluid [MW_{th}].

As stated by DiPippo (1980) [51], this formulation of efficiency loses its meaning since the value \dot{Q} is produced geothermally and not by a combustion fuel. So, he proposed the geothermal resource utilization efficiency (η_u) to properly evaluate the thermodynamic performance of a geothermal plant:

$$\eta_u = \frac{\dot{W}_{net}}{\dot{E}_{in}} [-]$$

Where:

- \dot{W}_{net} is the net power produced by the working fluid [MW_e];
- \dot{E}_{in} is the exergy of the geothermal fluid entering the power plant [MW].

The net power produced is evaluated by subtracting from the generated power (\dot{W}_{gen}), the power consumed by the different components inside the plant:

$$\dot{W}_{net} = \dot{W}_{gen} - (\dot{W}_{circ,pump} + \dot{W}_{cond,pump} + \dot{W}_{bd,pump} + \dot{W}_{compr} + \dot{W}_{tower,fans}) [MW_e]$$

Where:

- $\dot{W}_{circ,pump}$ is the power required by the circulation pump [MW_e];
- $\dot{W}_{cond,pump}$ is the power required by the condensation pump [MW_e];
- $\dot{W}_{bd,pump}$ is the power required by the blowdown pump [MW_e];
- \dot{W}_{compr} is the power required by the compressor [MW_e];
- $\dot{W}_{tower,fans}$ is the power required by the cooling tower fans [MW_e].

The exergy is the difference between the thermodynamic availability of the geothermal fluid entering the cycle (state 1), and its dead state (state 0)[51]:

$$\dot{E}_{in} = \dot{m}_{in} \cdot [h_{in} - h_0 - T_0(s_{in} - s_0)] [MW]$$

Where:

- \dot{m}_{in} is the mass flow rate of geothermal fluid [$\frac{kg}{s}$];
- h_{in} is the enthalpy at the inlet of the thermodynamic cycle [$\frac{kJ}{kg}$];
- h_0 is the enthalpy at dead state condition [$\frac{kJ}{kg}$];
- T_0 is the dead state temperature [K];
- s_{in} is the entropy at the inlet of the thermodynamic cycle [$\frac{kJ}{kgK}$];
- s_0 is the entropy at dead state condition [$\frac{kJ}{kgK}$].

In the proposed code, the dead state condition coincides with the ambient condition, which can be changed by the user.

The utilization efficiency allows us to understand how much power it is possible to extract from the working fluid. Dry steam power plants are the ones with the highest utilization efficiency among the proposed types of plants since all the steam produced from the wells is expanded in the turbine. Well-designed dry steam plant could operate with a utilization efficiency of 50% or 60%. A single flash power plant has lower utilization efficiency since a portion of the fluid is injected after the separator, without being used. Increasing the number of flash processes it is possible to increase the utilization of the working fluid, but it is not always economically viable[51].

In a binary power plant, this parameter loses its meaning because the working fluid is different from the brine, so in this case, it is better to refer to the first definition of thermal efficiency, considering the heat transferred to the ORC fluid as the parameter \dot{Q} .

Another parameter that provides information about the performance of the thermodynamic cycle is the so-called Specific Fluid Consumption (SFC), or Specific Steam Consumption (SSC). It represents the amount of steam required to produce 1 kWh of electricity.

It is versatile and applicable to all the types of geothermal power plants. It is calculated as:

$$SFC = SSC = \frac{\dot{m}}{E_{net}} \left[\frac{kg}{kWh} \right]$$

Where:

- \dot{m} is the mass flow rate of the working fluid $\left[\frac{kg}{s}\right]$;
- E_{net} is the energy produced in one hour $[kWh]$.

In the case of a dry steam power plant, this value can be around $8 \left[\frac{kg}{kWh}\right]$ [51].

A performance parameter which refers to the plant operations and not to the thermodynamic cycle is the capacity factor (CF), which is used in the economic evaluation of the code to estimate the total energy produced during a year of operation. It has two different formulations: one is the ratio between the energy produced during a given period and the maximum amount of energy which is possible to produce in the same period. The other is the ratio between the number of hours in which the plant operates and the hours in a year.

$$CF = \frac{\bar{L}}{C} [-]; \quad CF = \frac{H_{operation}}{H_{total}} [-]$$

Where:

- \bar{L} is the average load for a given period;
- C is the rated capacity of the plant unit;
- $H_{operation}$ is the number of hours which the plant operates;
- H_{total} is the number of hours in a year.

In the code, this factor is not calculated but is instead provided as an input by the user or set to a default value of 0.82, for the motivations reported in chapter 4.7.1. Geothermal power plants usually have higher CF due to the continuous availability of the resource, with modern plants reaching up to 0.95[6],[51].

4.5. Inputs required from the user

To run the code, several variables must be provided by the user, allowing flexibility in the possible design choices for the selected power plant. The requests for each variable differ due to varying input parameters.

For variables like efficiencies, where the value must be inside a specific range, a “while” cycle is implemented, preventing users from setting values that could lead to wrong results. An example is shown in Figure 27.

```
choiceEtaCondPumpValue=input('Want to set the efficiency of condensation pump? 1=yes, 0=no: ');
if choiceEtaCondPumpValue==1
    eta_condpump=input('set condensation pump efficiency: ');
    while eta_condpump<=0 || eta_condpump>=1
        eta_condpump=input('The efficiency must be between 0 and 1. Set condensation pump efficiency: ');
    end
    fprintf('The condensation pump efficiency is %f [-]\n', eta_condpump);
else
    eta_condpump=0.85;
    fprintf('The condensation pump efficiency is %f [-]\n', eta_condpump);
end
```

Figure 27. An example of a While cycle to set the value within the specific range. Author's own elaboration using MATLAB

As shown in Figure 27, the procedure for requesting these parameters involves prompting the user to choose whether to set the variable's value. If the user enters 1, they can input their preferred value; if they enter 0, the default value will be used.

To sum up all the variables, the following tables list each variable's name, its default value (if the user does not change it), the value range (if applicable), and the unit of measurement.

Dry steam	Value Range	Default Value	Unit
Ambient pressure	-	1	bar
Ambient temperature	-	20	°C
Lifetime	-	30	Year
Time step per year	-	1	1/year
Number of productions well	-	2	-
Number of injections well	-	1	-
Diameter of production well	0.0254-0.7620	0.2032	m
Diameter of the injection well	0.0254-0.7620	0.2032	m
Depth of production well	-	-	m
Diameter of the filter	0.1-0.5	0.4	m
Pressure outlet of the turbine	-	0.12	bar
Pressure outlet of the compressor	-	1	bar
Turbine efficiency	0-1	0.85	-
Vapour limit in expansion	0-1	0.80	-
Condensation pump efficiency	0-1	0.85	-
Blowdown pump efficiency	0-1	0.85	-
Circulation pump efficiency	0-1	0.85	-
Generator efficiency	0-1	0.95	-
Compressor efficiency	0-1	0.85	-
Fraction of NCG	0-0.05	0.05	-
%CO ₂ in NCG	0-1	0.95	-
%H ₂ S in NCG	0-1	0.04	-
%Hg in NCG	0-1	0.01	-
Thermic power to electric power	40 or 80	40	kW _e /kW _{th}
Type of geothermal field	1 or 2	1	-
Success rate in drilling operation	1(0.95), 2(0.75), 3(0.9)	0.9	-
Capacity factor	0-1	0.82	-
Electricity price	-	0.09	\$/kWh
Capex incentives	0-1	0	-
Opex incentives	0-1	0	-
Revenues incentives	0-1	0	-

Table 5. Input variables asked to the user in the dry steam plant. Author's own elaboration.

Dry steam with cogeneration	Value Range	Default Value	Unit
Ambient pressure	-	1	bar
Ambient temperature	-	20	°C
Lifetime	-	30	Year
Time step per year	-	1	1/year
Number of productions well	-	2	-
Number of injections well	-	1	-
Diameter of production well	0.0254-0.7620	0.2032	m
Diameter of injection well	0.0254-0.7620	0.2032	m
Depth of production well	-	-	m
Diameter of the filter	0.1-0.5	0.4	m
Pressure outlet of turbine	-	0.12	bar
Pressure outlet of the compressor	-	1	bar
Turbine efficiency	0-1	0.85	-
Vapour limit in expansion	0-1	0.80	-
Condensation pump efficiency	0-1	0.85	-
Blowdown pump efficiency	0-1	0.85	-
Circulation pump efficiency	0-1	0.85	-
Generator efficiency	0-1	0.95	-
Compressor efficiency	0-1	0.85	-
Fraction of NCG	0-0.05	0.05	-
%CO ₂ in NCG	0-1	0.95	-
%H ₂ S in NCG	0-1	0.04	-
%Hg in NCG	0-1	0.01	-
Thermic power to electric power	40 or 80	40	kW _e /kW _{th}
Fraction of split geothermal fluid	0-1	0.3	-
End-user efficiency	0-1	0.9	-
Type of geothermal field	1 or 2	1	-
Succes rate in drilling operation	1(0.95), 2(0.75), 3(0.9)	0.9	-
Capacity factor	0-1	0.82	-
Electricity price	-	0.09	\$/kWh
Heat price	-	0.02	\$/kWh
Capex incentives	0-1	0	-
Opex incentives	0-1	0	-
Revenues incentives	0-1	0	-

Table 6. Input variables asked to the user in dry steam plant with cogeneration. Author's own elaboration.

Single Flash	Value Range	Default Value	Unit
Ambient pressure	-	1	bar
Ambient temperature	-	20	°C
Lifetime	-	30	Year
Time step per year	-	1	1/year
Number of productions well	-	2	-
Number of injections well	-	1	-
Diameter of production well	0.0254-0.7620	0.2032	m
Diameter of injection well	0.0254-0.7620	0.2032	m
Depth of production well	-	-	m
Diameter of the filter	0.1-0.5	0.4	m
Pressure of separation	-	6.2	bar
Pressure outlet of turbine	-	0.12	bar
Pressure outlet of the compressor	-	1	bar
Turbine efficiency	0-1	0.85	-
Vapour limit in expansion	0-1	0.80	-
Condensation pump efficiency	0-1	0.85	-
Blowdown pump efficiency	0-1	0.85	-
Circulation pump efficiency	0-1	0.85	-
Generator efficiency	0-1	0.95	-
Compressor efficiency	0-1	0.85	-
Fraction of NCG	0-0.05	0.05	-
%CO ₂ in NCG	0-1	0.95	-
%H ₂ S in NCG	0-1	0.04	-
%Hg in NCG	0-1	0.01	-
Thermic power to electric power	40 or 80	40	kW _e /kW _{th}
Type of geothermal field	1 or 2	1	-
Success rate in drilling operation	1(0.95), 2(0.75), 3(0.9)	0.9	-
Capacity factor	0-1	0.82	-
Electricity price	-	0.09	\$/kWh
Capex incentives	0-1	0	-
Opex incentives	0-1	0	-
Revenues incentives	0-1	0	-

Table 7. Input variables asked to the user in single flash plant. Author's own elaboration.

Binary	Value Range	Default Value	Unit
Ambient pressure	-	1	bar
Ambient temperature	-	20	°C
Lifetime	-	30	Year
Time step per year	-	1	1/year
Number of productions well	-	2	-
Number of injections well	-	1	-
Diameter of production well	0.0254-0.7620	0.2032	m
Diameter of injection well	0.0254-0.7620	0.2032	m
Depth of production well	-	-	m
Turbine efficiency	0-1	0.85	-
Vapour limit in expansion	0-1	1	-
Condensation pump efficiency	0-1	0.85	-
Blowdown pump efficiency	0-1	0.85	-
Circulation pump efficiency	0-1	0.85	-
Generator efficiency	0-1	0.95	-
Global heat transfer coefficient in evaporator	500-1000	500	W/m ² K
Global heat transfer coefficient in condenser	375-750	500	W/m ² K
Thermic power to electric power	40 or 80	40	kW _e /kW _{th}
Pressure of isobutane	-	25	bar
Temperature of isobutane	-	15	°C
Pressure outlet of turbine	-	4	bar
Pressure in the cooling tower loop	-	2	bar
Minimum temperature difference in evaporator	-	5	°C
Temperature difference of fluids in the cooling tower	-	5	°C
Type of geothermal field	1 or 2	1	-
Succes rate in drilling operation	1(0.95), 2(0.75), 3(0.9)	0.9	-
Type of heat exchanger	1 or 2	1	-
Type of condenser	1 or 2	1	-
Capacity factor	0-1	0.82	-
Electricity price	-	0.09	\$/kWh
Capex incentives	0-1	0	-
Opex incentives	0-1	0	-
Revenues incentives	0-1	0	-

Table 8. input Variable asked to the user in binary plant. Author's own elaboration.

4.6. Validation of the models

The validation of these models is required to check the reliability of the codes simulating the thermodynamic cycles of the plants. This validation process was done by simulating the operations of the real power plants reported in the work done by Zarrouk et al. (2014)[63], which is a worldwide review of the efficiency of the geothermal power plants. The review categorizes plants by type, reporting their installed capacity, operational capacity, mass flow rate and the enthalpy of the geothermal fluid. The band error in which the plant is considered well-represented is 10%. It was arbitrarily chosen due to the lack of information regarding the operational parameters of the plants reported. Only for some flash power plants the separation pressure and the turbine outlet pressure are indicated. Consequently, during the validation simulations, the default values listed in the previous tables were used, and the difference between these and the real values significantly affected the power output.

The code requires temperature, pressure and mass flow rate as inputs, but only enthalpy is provided, which is a function of the other two variables. Therefore, it was needed to find the inlet condition of the geothermal fluid at the given value of enthalpy. A search code was used for this purpose, employing two vectors: one for pressure, ranging from 5 to 30 [bar], and one for entropy, ranging from 0 to 9 $\left[\frac{kJ}{kgK}\right]$.

Both vectors consisted of 1000 equally spaced points within their ranges. Entropy was used instead of temperature due to its greater precision when the given enthalpy lies below the saturation curve.

This search code takes into account all the possible combinations of the values inside the two vectors, creating a third vector saving the enthalpy values from all the combinations. Then it evaluated the difference between the input enthalpy value and those inside the vector, identifying the index of the minimum difference. This process enabled the determination of the corresponding pressure and entropy values. Finally, the code computed the temperature and the vapour fraction, which is used as a validation check: for a dry steam power plant, the vapor fraction should equal 1, while for a binary power plant, it should equal 0, and for a single flash power plant, it should be less than 1.

```
%% vector creation
pmin=5; % bar
pmax=30; % bar
smin=0; % kJ/kgk
smax=9; % kJ/kgk
pressure=linspace(pmin,pmax,500);
entropy=linspace(smin,smax,500);


---


%% h input
h_input=1026; % kJ/kg
% Use meshgrid to generate combinations
[X, Y] = meshgrid(pressure, entropy);
% Combine them into a single matrix
ps = [X(:), Y(:)];
for i=1:length(ps)
    h(i)=XSteam('h_ps',ps(i,1),ps(i,2));
end
diff=abs(h-h_input);


---


%% find index
index=find(min(diff)==diff(:));


---


%% output
p_out=ps(index,1);
s_out=ps(index,2);
h_find=XSteam('h_ps',p_out,s_out);
x_find=XSteam('x_ph',p_out,h_find);
T_find=XSteam('T_ph',p_out,h_find);
```

Figure 28. Code to find pressure and temperature in function of the given enthalpy. Author's own elaboration using MATLAB

To show how good is the search code in finding the right enthalpy, and consequently the right combination of physical properties characterizing the geothermal fluid, Figure 29 compares all evaluated enthalpy values with the input enthalpy. All of them are on a continuous line, indicating that they share the same value.

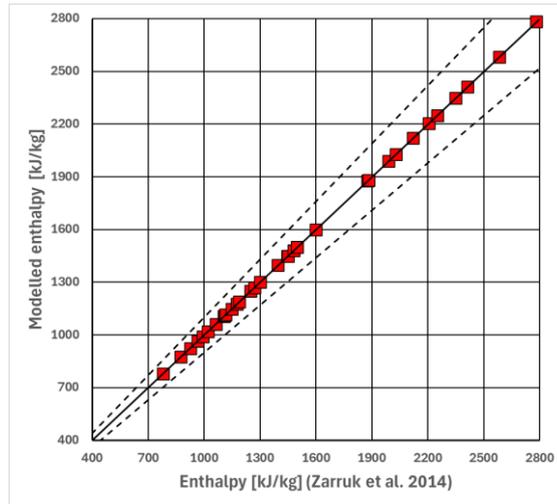


Figure 29. Enthalpies found by the search code respect the one given as input. Author's own elaboration using Excel

4.6.1. Dry steam model

Dry steam power plants are listed with flash power plants in the work of Zarruk et al. (2014)[63], because they are only six. Moreover, there is a consistent difference between the installed capacity and the operational capacity; therefore, for validation purposes, the latter was considered.

Figure 30 shows the graph in which the evaluated generated power (\dot{W}_{gen}) is compared with the running capacity. It is possible to see that five of six powers are inside the bandwidth and well represented, only one is a little bit overestimated, in which the generated power calculated from the code is greater than 30 MW with respect to the real one. The best-represented point is the one in which the vapour fraction is exactly one, for all the other cases this parameter is around 0.99, expected for the last power plant with the higher power, in which the vapour fraction is 0.92. However, it still be inside the error band chosen.

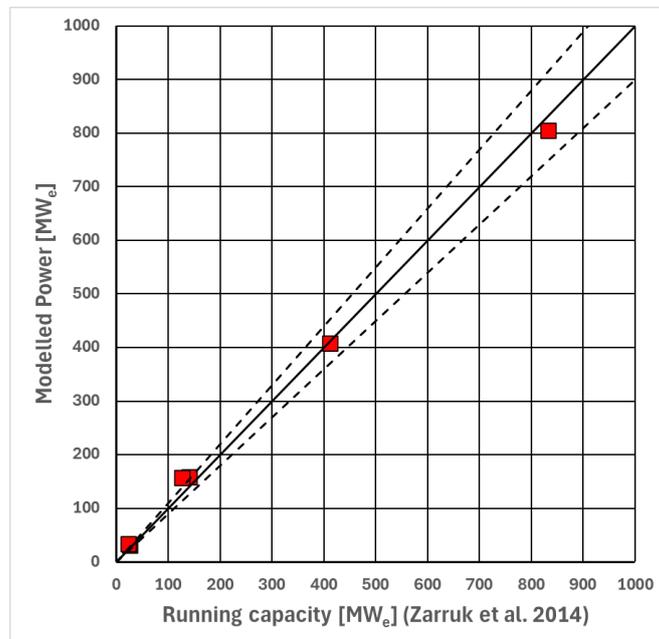


Figure 30. Generated power of dry steam thermodynamic cycle versus the running capacity of real power plants. Author's own elaboration using Excel

4.6.2. Single flash model

The number of single flash power plants reported is sufficiently high to have a good insight into the behaviour of the code. From Figure 31 it is possible to see that in general this thermodynamic cycle is

well represented. Only 7 cases out of 34 are overestimated or underestimated with respect to the running capacity, and it is also considered that in most cases the pressure of separation and the outlet pressure of the turbine are supposed. They are respectively 6.2 bar and 0.12 bar, as reported by Zarrouk et al. (2014)[63], they are the mean values based on the ones known.

These two values both influence the power generated by the turbine: the first commands the mass flow rate of geothermal fluid which becomes steam and expands, while the second commands the final enthalpy difference between the inlet and outlet of the turbine. So not knowing the exact value of them, it is possible to have an error in the evaluation.

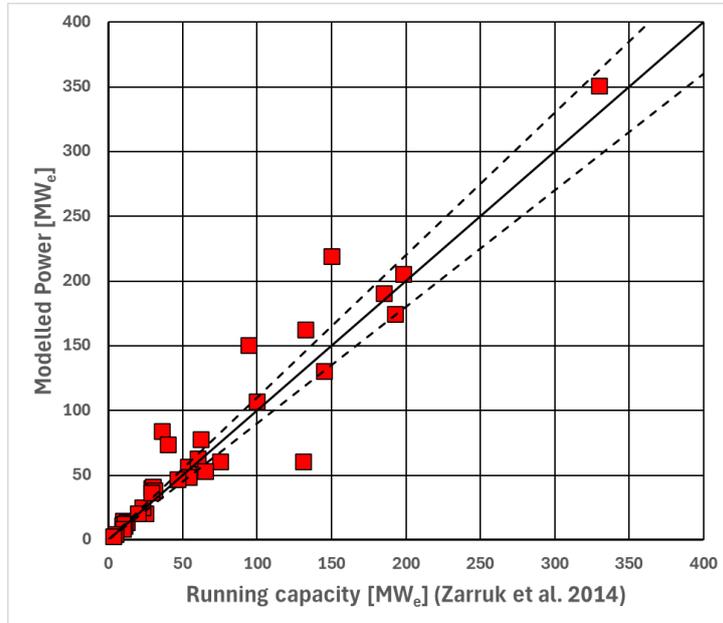


Figure 31. Generated power of single flash thermodynamic cycle versus the running capacity of real power plants. Author's own elaboration using Excel

4.6.3. Binary model

The binary model is the more complex of the three, and it has also a higher number of parameters that should be known to make the code work properly. In Zarrouk et al. (2014)[63] for binary plants are also reported the inlet and injection temperature of the geothermal fluid, but no other information regarding the loop of the organic fluid, neither the pressure at which it enters in turbine or at which it expands, nor the mass flow rate nor the temperature after the heat exchanger. All these unknowns make it easier to have errors in the simulation, and different power from the listed ones.

To simulate the thermodynamic cycle in the best way with the data known, the inlet pressure at the turbine was evaluated using the following formula, which evaluates the saturated-vapor pressure of isobutane in function of the temperature [75]:

$$p_{vap} = \exp \left(A + \frac{B}{T} + C \log(T) + DT^2 \right) [kPa]$$

where A , B , C , and D are constants and the temperature is given in Kelvin.

The saturated-vapor pressure follows the trend shown in Figure 32, which was plotted for a temperature range from 113.54 to 408.14 K, which is the one in which this function is valid.

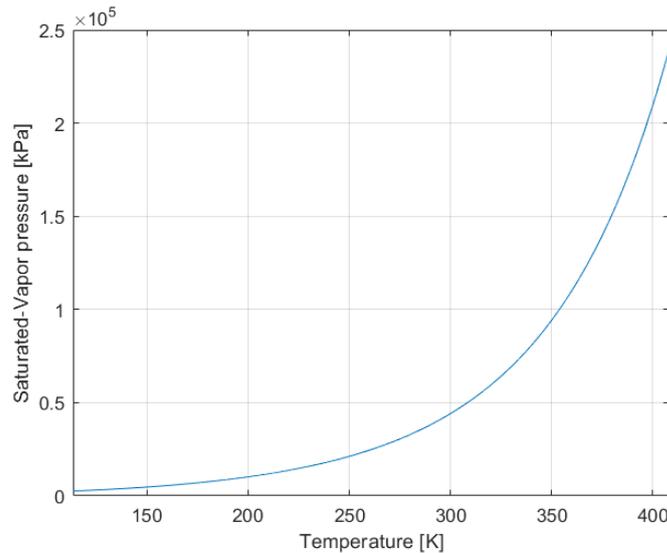


Figure 32. Trend of the saturated-vapor pressure in function of the temperature. Author's own elaboration using Excel

The input temperature given to the function to run the simulations was the difference between the inlet temperature of the geothermal fluid and the minimum temperature difference multiplied by a factor which increased as the temperature of the brine increased. This factor was changed manually trying to match the outlet temperature from the heat exchanger of the geothermal fluid when possible. It is worth to remember that this factor and this type of calculation are implemented only for these simulations, and they are not part of the final code, since the pressure of the organic fluid is asked in input to the user. From Figure 33 (left) it is possible to see that also for this type of plant, most of the simulation is inside the error bandwidth, and only 8 out of 31 are overestimated or underestimated, and in all these cases the error starts probably from a wrong evaluation of the main pressure of the organic fluid, which influences the mass flow rate of isobutane and consequentially the injection temperature, which does not influence the power generated.

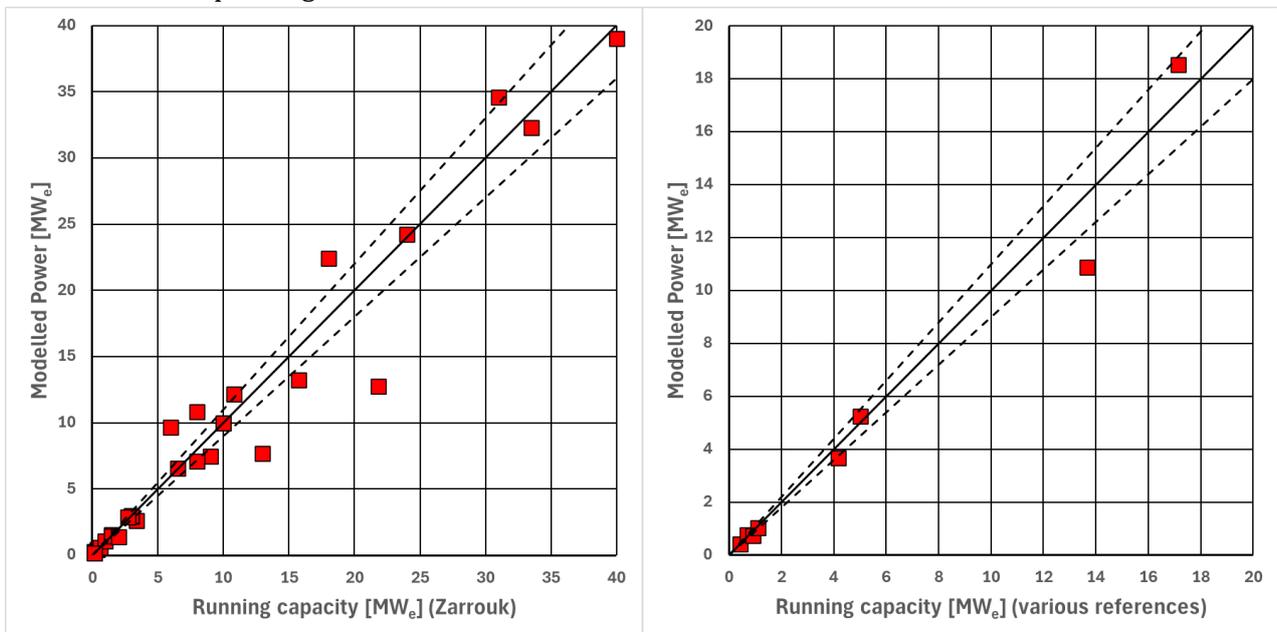


Figure 33. Generated power of binary thermodynamic cycle versus the running capacity of real power plants. Author's own elaboration using Excel

Since the parameters which can influence the output from this type of plant are more than the others, it was decided to make some other simulations, but only using data which gives also information about the organic loop. From how it is possible to see from Figure 33 (right), knowing more details on the

design parameters, the code shows better performance, with only one value outside the error bandwidth on the 8 simulations done.

4.7. Economic model

The economic model is implemented to understand if the plant's operation is feasible, and it is done through the evaluation of two parameters: the Levelized Cost of Energy (LCOE) and the Net Present Value (NPV). The LCOE is the unit cost of energy computed from the present cost value to the end of the plant's lifetime. It is used to compare different technologies with unequal lifetime, size, price, risk, etc., and generally, it allows to understand the competitiveness of the energy resource [76].

The NPV is the present value of the net cash flow at the required rate of return, in practical terms, it is a method to calculate the Return On Investment (ROI) of the plant, making it possible to decide if the project is worthwhile [77].

In the proposed code, the LCOE is a single value evaluated over the plant's lifetime, while the NPV is saved as a vector but only the final value is plotted. They are both influenced by the net power produced, which varies with time since the conditions of the geothermal fluid vary too.

The NPV calculation considers also the possibilities to have incentives to reduce the capital expenditure cost (CAPEX), to reduce operation and maintenance cost (OPEX), and to have different income from the electrical energy sales. They are percentage of the cost which refer, and the user is free to consider them or not in its analysis, and to change their values. The decision to make the user decide their percentage, outcomes from considering that there are different possibilities for all the three cases but vary from country to country. These incentives are not taken into account in the evaluation of the LCOE.

4.7.1. Levelized cost of energy

The LCOE is calculated using the Risk-Adjusted Discount method, as shown by Park et al. (2021) [76], and it is:

$$LCOE_{RAD} = \frac{\sum_{t=0}^n \frac{C_t + O_t + V_t}{(1 + R_{RAD})^t}}{\sum_{t=0}^n \frac{E_t}{(1 + R_{RAD})^t}} \left[\frac{\$}{MWh} \right]$$

Where:

- C_t is the capital cost at the year t ;
- O_t is the cost of operation and maintenance at the year t ;
- V_t is the variable cost at the year t ;
- E_t is the energy produced by the plant in the year t ;
- R_{RAD} is the discount rate.

In the model, the capital cost appears only at year zero, which is added to consider that the plant is constructed and there are no incomes in that year. Operation and maintenance costs are taken into account each year, excluding the year zero, and they are 0.5% of the capital cost as suggested by IRENA (2021) [78]. The variable costs are equal to zero in case of geothermal power plant, as reported by NREL (2024) [79]. However, the variable exits inside the code in the case the user wants to consider them.

The energy produced in the year is the net power multiplied by the hour in the year, and by the capacity factor:

$$E_{net} = \dot{W}_{net} \cdot CF \cdot 8766 [MWh]$$

The hours in one year are set equal to 8766, in order to consider the leap year. The value of the capacity factor is set as default equal to 0.82 because it is the mean value between the mean capacity factor of the three implemented technologies, based on data reported in IRENA (2021) [78], but the user can change it.

The discount rate can be evaluated with the method of the Weight Average Capital Cost (WACC), using the following formula reported by Park et al. (2021) [76]:

$$WACC = W_d \times K_d \times (1 - t) + W_e \times k_e$$

Where:

- W_d is the weight of debt as a percentage of the total capital;
- W_e is the weight of equity as a percentage of the total capital;
- K_d is the cost of debt;
- K_e is the cost of equity;
- t is the marginal tax rate.

Moreover, the cost of equity can be further evaluated as:

$$K_e = R_f + EMRP \times \beta_e$$

Where:

- R_f is the risk-free rate;
- $EMRP$ is the expected market risk premium;
- β_e is the equity beta.

Since these parameters can vary in time and by country, in the code it is asked to the user to insert the WACC, and these calculations are not implemented. However, the user can use these formulas to make their evaluation and then use that value. In the code, the default value for WACC is set equal to 5%, as done in the evaluation of IRENA (2021) [78]. Be warned that in that report the value of 5% is for countries which are part of the Organisation for Economic Co-operation and Development (OECD) and China, for other countries this value must be set to 10%.

4.7.2. Net Present Value

The NPV represents the yearly net cash flow of a project, and it is evaluated following the definition shown by Gallo (2014) [77], adding the possible incentives:

$$NPV = -CAPEX \cdot (1 - CAPEX_{incentives}) + \sum_{t=1}^n \frac{B_t}{(1 + R_{RAD})^t}$$

Where B_t is the cash flow at the year t , evaluated as the difference between the revenues of the year and the operating and maintenance cost. The revenues come from selling electricity, and in the case of cogeneration also from selling heat:

$$B_t = (E_t \cdot Price_{electricity} + Q_t \cdot Price_{heat}) \cdot (1 + REVENUE_{incentives}) - O_t \cdot (1 + OPEX_{incentives}) - V_t \text{ [\$]}$$

The price at which they are sold will greatly influence the NPV. The default value for electricity price is $0.09 \left[\frac{\$}{kWh} \right]$, based on the mean value of the wholesale electricity price in Europe from 2023 to September 2024 provided by EMBER (2024)[80], and multiplied by the conversion coefficient from Euro to Dollars, equal to $1.12 \left[\frac{\$}{\text{€}} \right]$ in the day consulted [81]. The heat price is set arbitrarily at $0.02 \left[\frac{\$}{kWh} \right]$. These values can be changed, and it is highly recommended since the price changes by country and time.

4.7.3. Costs evaluation

The evaluation of the capital cost is done following the method proposed by genGEO [38], with the addition of the cost evaluation of some components that are not considered in that model. For example, the compressor to extract the non-condensable gases.

Other than the costs of the components are also considered the cost for wells field, wells, surface piping, and exploration. The cost of stimulation here is not considered.

The cost of administration, planning and design are indirect costs (IC), and each component cost is multiplied by its own IC index. The same is done with the project contingency costs (PC), which are

included to take into account unknown costs. The value used for PC is always 1.15, while for IC is 1.05, excluding surface piping and plant for which is 1.12.

Another multiplication factor used to take into account the inflation in the cost is the Production Price Index (PPI), the index are taken by the U.S. Bureau of Labor Statistics (BLS) using the identification code reported by genGEO [38]. Then the values of 2002 and 2023 are taken, the first is used as a reference, and the second indicates the year in which the cost is evaluated. The cost of the compressor, direct contact condenser and AMIS unit are evaluated through the Chemical Engineering Plant Cost Index (CEPCI) since they implement formulation in which the cost is scaled to the year 2003 of the CEPCI index. The value of CEPCI reported in Table 9 is the mean value of the months of that year.

Pumps, compressor, direct contact condenser and AMIS unit have as reference year for indexes 2003.

ID code	Reference to	Index	2002/2003*	2023	Ratio
PCU332996332996	Pipe	PPI _{Pipe}	153.7	448.37	2.917
WPU1197	Turbine and generator	PPI _{T-G}	165.6	266.89	1.612
WPU1075	Heat Exchanger	PPI _{HX}	174.7	422.13	2.416
PCU332911332911	Process Equipment	PPI _{PE}	128.0	327.66	2.560
PCU213111213111	Oil and Gas well	PPI _{O&G}	153.5	402.82	2.624
PCU213111213111P	Drilling Service	PPI _{DS}	153.2	400.09	2.612
PCU5411--5411--	Permitting (legal service)	PPI _{Permit}	121.7	271.08	2.227
PCU213112213112	Oil and Gas support	PPI _{O&G-s}	137.4	214.94	1.564
PCU33391-33391-	Pump	PPI _{Pump}	102.2*	210.90	2.064
	Compressor, Direct contact condenser, AMIS unit	CEPCI _{index}	395.6*	797.94	2.017

Table 9. Production Price Index (PPI) and Chemical Engineering Plant Cost Index in years 2002/2003 and 2023. Author's own elaboration. Sources: U.S. Bureau of Labor Statistics [82], and Cost Indices [83]

4.7.3.1. Turbine and Generator cost

To evaluate the turbine cost is implemented the formula by genGEO, which considers not only the turbine but also the generator:

$$C_{T-G} = 0.67 \cdot PPI_{T-G} \cdot \left[S_{T-G} \cdot 2800 (\dot{W}_{turb} \cdot 1000)^{0.745} + 3860 (\dot{W}_{turb} \cdot 1000)^{0.617} \right] [\text{\$}]$$

Where the parameter S_{T-G} consider the nature of the fluid, so in the case of a single flash or dry steam plant, it is equal to one, while for organic fluid it is 1.20, as done in genGEO for using CO₂ as working fluid. The value is set equal since there are no other reference on the possible value using isobutane or other organic fluid.

4.7.3.2. Pump cost

The pump cost is evaluated using the cost correlation in function of the power required provided by GETEM [84]. The condensation and circulation pump follow the same equation, while the injection pump is a little bit different.

$$C_{cond,pump} = PPI_{pump} S_{pump} \cdot 1185 (1.34 \cdot \dot{W}_{cond,pump})^{0.767} [\text{\$}]$$

$$C_{circ,pump} = PPI_{pump} S_{pump} \cdot 1185 (1.34 \cdot \dot{W}_{circ,pump})^{0.767} [\text{\$}]$$

$$C_{inj,pump} = PPI_{pump} S_{pump} \cdot 1750 (1.34 \cdot \dot{W}_{inj,pump})^{0.7} [\text{\$}]$$

The pump power is in kW_e and S_{pump} is a factor which takes into account the material of the pump, in the three models is equal to one since the fluid is water and isobutane, which is not corrosive.

4.7.3.3. Cooling Tower cost

The cooling tower cost is evaluated considering a design coefficient (TDC), which is set equal to 0.252 since it works with water, considering the temperature difference between fluid outlet temperature and ambient (ΔT_{appr}), and the temperature difference between the fluid inlet and outlet of the cooling tower (ΔT_{range}). These last two parameters are used to evaluate the cooling-specific cost ($c_{cooling, cost}$), in which they multiply two regression coefficients which refer to dry cooling tower. The reference thermic power exchanged ($\dot{Q}_{ref, BAC}$) is equal to 1 MW_{th}.

$$c_{cooling, cost} = 7.31 \cdot 10^3 \left(\frac{1}{\Delta T_{appr}} \right) + 1.23 \cdot 10^3 \left(\frac{1}{\Delta T_{appr} + \Delta T_{range}} \right) \left[\frac{\$}{kW_{th}} \right]$$

$$C_{ref, BAC} = PPI_{PE} \cdot (\dot{Q}_{ref, BAC} \cdot 10^3) \cdot TDC \cdot c_{cooling, cost} \text{ [\$]}$$

$$C_{coolingTower} = C_{ref, BAC} \left(\frac{\dot{Q}_{coolingTower}}{\dot{Q}_{ref, BAC}} \right)^{0.8} \text{ [\$]}$$

Where $\dot{Q}_{coolingTower}$ is the thermic power exchanged inside this component evaluated in the code.

4.7.3.4. Direct contact condenser cost

The cost of a direct contact condenser is evaluated following the method shown in Woods (2007) [85], and it is only a function of the volumetric sprayed flow rate. The reference volumetric mass flow rate ($\dot{V}_{spray, ref}$) is 33 $\left[\frac{l}{s} \right]$, the reference cost ($FOB_{condenser}$) is 22'000\$.

$$C_{condenser} = CEPCI_{index} FOB_{condenser} \left[\frac{\left(\frac{\dot{m}_{spray} \cdot 1000}{\rho_{spray}} \right)}{\dot{V}_{spray, ref}} \right]^{0.6} \text{ [\$]}$$

4.7.3.5. Compressor cost

The evaluation of the compressor's cost follows the formulation proposed by Woods (2007) [85]. Among the various types of compressors, it was chosen the centrifugal compressor, with maximum compression pressure equal to 7 MPa. The reference cost (FOB_{compr}) is 875'000 \$ for a reference drive power ($\dot{W}_{compr, ref}$) of 1000 kW. The other factors which consider the outlet maximum pressure ($F_{pressure}$) at 1.7 MPa is equal to 0.8.

$$C_{compr} = CEPCI_{index} FOB_{compr} F_{pressure} \left(\frac{\dot{W}_{compr} \cdot 1000}{\dot{W}_{compr, ref}} \right)^{0.53} \text{ [\$]}$$

4.7.3.6. Heat Exchanger

For the heat exchanger, it is possible to choose between two different technologies: shell and tube or plate heat exchanger. The first costs more but it is more efficient with respect to the other one, also if in the code there is not taken into account a difference in efficiency in the heat exchange process based on the selected technology. This differentiation is done to make the economic evaluation more flexible, giving two options to the user since the cost of this component is not negligible.

The two formulas implemented come from Peters et al. (2003) [71], for shell and tube with floating head and welded plate, both in function of the area required for the heat exchange:

$$C_{shell\&tube} = PPI_{HX} (239 \cdot A + 13400)$$

$$C_{plate} = PPI_{HX} (69 \cdot A + 4670)$$

4.7.3.7. AMIS Unit

The Abatement of Mercury and Sulphide (AMIS) unit was developed and patented by Enel, and it has proven to be efficient and reliable, and the three main equipment which compose this unit are [64]:

1. A catalytic carbon active fixed bed reactor, which through absorption removes the mercury;
2. A catalytic titanium dioxide fixed bed reactor, which converts H₂S in SO₂;

3. A packed column wet scrubber, in which SO₂ is absorbed by the cooling water as sulphites.

The cost of the other equipment which composes the AMIS unit is considered in the secondary equipment cost factor (X_{SE}).

In literature there is no cost which refers to the unit in its entirety, so to find it, the cost of the three main components is summed and then multiplied by the CEPCI index since they are modelled with reference to Woods (2007) [85].

The catalytic carbon active and catalytic titanium dioxide fixed beds implement the same equation, one difference is the exponent which elevates the ratio between the volumetric flow rate of the gas and the reference one ($\dot{V}_{gas,ref}$), equal to 70 $\left[\frac{dm^3}{s}\right]$. The exponent depends on the range of admissible volumetric flow rate. The other difference is that titanium dioxide, used as material, influences the reference cost ($FOB_{remover}$), equal to 23'500 \$, with a factor (F_{TiO_2}) of 8.

$$C_{AC,bed} = FOB_{remover} \left[\frac{\frac{\dot{m}_{Hg^+} \cdot 1000}{\rho_{Hg^+}}}{\dot{V}_{gas,ref}} \right]^{0.32} \quad [\$]$$

$$C_{TiO_2,bed} = FOB_{remover} F_{TiO_2} \left[\frac{\frac{\dot{m}_{H_2S} \cdot 1000}{\rho_{H_2S}}}{\dot{V}_{gas,ref}} \right]^{0.67} \quad [\$]$$

The cost of the packed column wet scrubber is a function of the volumetric flow rate of SO₂ produced in the chemical reaction inside the previous reactor, and it is supposed equal to the volumetric flow rate of H₂S. The reference cost ($FOB_{wet,scrubber}$) is 35'000 \$, while the reference volumetric flow rate is equal to 1.65 $\left[\frac{m^3}{s}\right]$

$$C_{wet,scrubber} = FOB_{wet,scrubber} \left(\frac{\dot{V}_{SO_2}}{\dot{V}_{wet,scrubber,ref}} \right)^{0.39} \quad [\$]$$

In the end, the total cost of the AMIS unit is:

$$C_{AMIS} = CEPCI_{index} (C_{AC,bed} + C_{TiO_2,bed} + C_{wet,scrubber}) \quad [\$]$$

4.7.3.8. Total plant cost

The evaluation of the total plant cost starts with the evaluation of the primary equipment cost (PEC). In the proposed model all the components shown in the plant's reference map are considered primary components, excluding only the filter and the separator. Contrarily, in genGEO the primary components are the turbine, heat exchanger, pump and cooling tower.

Here the plant's PEC, in case of dry steam or single flash, is:

$$C_{plant,PEC} = C_{turb} + C_{cond,pump} + C_{circ,pump} + C_{inj,pump} + C_{coolingTower} + C_{condenser} + C_{compr} + C_{AMIS} \quad [\$]$$

For binary power plants, the PEC is:

$$C_{plant,PEC} = C_{turb} + C_{cond,pump} + C_{circ,pump} + C_{inj,pump} + C_{coolingTower} + C_{HX,evap} + C_{HX,cond} \quad [\$]$$

Then it is evaluated the plant's total equipment cost (TEC), considering the secondary equipment factor, which accounts for valves, control systems, minor components, etc., but this value has been reduced with respect to the one proposed by genGEO. Adams et al. [38] suggest a value equal to 1.39, but it is reduced to 1.15, because of two reasons. The first regards the number of primary components considered, which in this model is higher. The second resides in the fact that following the cost evaluation method proposed by Woods, some components already consider the secondary equipment cost within them.

$$C_{plant,TEC} = C_{plant,PEC} \cdot X_{SE} \quad [\$]$$

The plant's bare erected cost (BEC) is calculated, and it considers different parameters: construction labour and fringe cost (XCL=0.58), construction material (XCM=0.11), sales tax (XST=0), and freight cost (XF=0.04). These parameters are the same as the ones proposed in genGEO.

$$C_{plant,BEC} = C_{plant,TEC}(1 + X_{CL} + X_{CM} + X_{ST} + X_F)[\$]$$

The final step to evaluate the total cost of the power plant is to multiply this last value for the project contingency and indirect cost.

$$C_{plant} = C_{plant,BEC} \cdot X_{IC,p} \cdot X_{PC,p} [\$]$$

4.7.3.9. Cost of wells

The cost for drilling procedures is called in the code "well cost", it depends on the number of wells (n), the depth of the well (L_{well}), and the diameter of the well (d_{well}). The cost of both production and injection wells is estimated. Then the total is increased using a parameter which considers the probability of success in the drilling, and it depends on the type of geological conformation. For sedimentary conformation, the success rate is 0.95, for sedimentary is 0.75, and for basement is 0.9.

The cost is evaluated as done in genGEO [38]:

$$C_{inj,well} = X_{IC,well} \cdot X_{PC,well} \cdot PPI_{O\&G} \cdot n_{inj} (0.105 \cdot L_{inj,well}^{0.2} + 1776 \cdot L_{inj,well} \cdot d_{inj,well} + 275300) [\$]$$

$$C_{prod,well} = X_{IC,well} \cdot X_{PC,well} \cdot PPI_{O\&G} \cdot n_{prod} (0.105 \cdot L_{prod,well}^{0.2} + 1776 \cdot L_{prod,well} \cdot d_{prod,well} + 275300) [\$]$$

$$C_{well} = \frac{C_{inj,well} + C_{prod,well}}{SuccessRate} [\$]$$

The choice to evaluate separately the cost for injection and production wells has been taken considering that usually, the injection stops at lower depths.

4.7.3.10. Surface piping cost

The surface piping cost is a difficult value to estimate since it depends on the total length of the pipes which connect the wells to the power plant. This value is not evaluated inside the code since depends on the wells and plant's location. It is supposed to be equal to 5 km, but the user can change it. However, the genGEO formulation is implemented:

$$C_{surface,piping} = X_{PC,pipe} \cdot X_{IC,pipe} \cdot PPI_{pipe} \cdot L_{pipe} \cdot (2205 \cdot d_{pipe}^2 + 134) [\$]$$

4.7.3.11. Exploration cost

Then it is evaluated the exploration cost, is composed of the modelling cost and exploration well drilling cost. The second follows the same equation of the wells' cost, the only difference is the difference in the diameter.

$$C_{modeling} = X_{IC,expl} \cdot X_{PC,expl} \cdot PPI_{O\&G-s} \left(508200 \frac{\$}{site} \right) [\$]$$

$$C_{stim,hole} = X_{IC,well} \cdot X_{PC,well} \cdot PPI_{O\&G} \cdot n_{inj} (0.105 \cdot L_{stim,hole}^{0.2} + 1776 \cdot L_{stim,hole} \cdot d_{stim,hole} + 275300) [\$]$$

$$C_{exploration} = C_{modeling} + \frac{C_{stim,hole}}{SuccessRate} [\$]$$

Here the evaluation is different from the one used in genGEO. The difference is in the evaluation of the exploration well cost, that in genGEO is 20% of the well's cost, divided by the success rate in the drilling operation and multiplied by the number of explorations well.

4.7.3.12. Permitting cost

The cost for the permission considers the wellfield and power plant permission, following the formula proposed by genGEO it is:

$$C_{permitting} = X_{IC,wellfield} \cdot X_{PC,wellfield} \cdot PPI_{permit} \cdot \left(665'700 \frac{\$}{site} \right) [\$]$$

4.7.3.13. Grid cost

In addition to all these costs, it has been decided to introduce the grid connection cost, composed of two terms as suggested in GEOELEC [86], one is the cost of the grid connection which depends on the power installed, the other one is a variable cost, and it accounts for the distance from the grid.

$$C_{grid,investment} = \frac{80}{conversion_{EUR,USD}} \cdot \dot{W}_{gen} \cdot 1000 \text{ [\$]}$$

$$C_{grid,connection} = L_{grid} \cdot \frac{100}{conversion_{EUR,USD}}$$

$$C_{grid} = C_{grid,investment} + C_{grid,connection}$$

The constant price used in the formula is in Euro, while all the other costs are evaluated in Dollars, so to convert is used the conversion coefficient reported by Google Finance [81], at the day on which was consulted. Its value is $1.12 \left[\frac{\text{€}}{\text{\$}} \right]$.

4.7.3.14. Greenfield and Brownfield cost

When a geothermal field is not yet explored it is called “Greenfield”. In the evaluation of a greenfield, must be taken into account the cost of exploration, the cost of drilling the wells, the cost of making grid connection, the cost of permissions to work in the area, the cost of piping to connect the wells to the plant, and the cost of stimulation in some cases, other than the cost of the power plant. On the contrary, when a geothermal site is already explored, it is called “Brownfield”, and the project cost for this type of geothermal site is lower since there are not most of the costs.

In the proposed model these costs are evaluated, and the user can decide if the area is a greenfield or a brownfield.

Considering all the costs listed, it is possible to evaluate the total cost in the case of greenfield and brownfield, and they will be the CAPEX in the economic analysis of LCOE and NPV. Since the brownfield final cost is lower compared with the greenfield cost, the LCOE will be lower, while the NPV at the end of the plant’s lifetime will be higher.

$$C_{greenfield} = \frac{C_{plant} + C_{well} + C_{surface,piping} + C_{exploration} + C_{permitting} + C_{grid}}{10^6} \text{ [M\$]}$$

$$C_{brownfield} = \frac{C_{plant} + C_{prod,well} + C_{surface,piping} + C_{grid}}{10^6} \text{ [M\$]}$$

This economic model must be intended as a first estimation of the possible revenue from a hypothetical power plant, and a more accurate analysis is recommended. However, compared to the reviewed codes, the NPV estimation is an addition which can gives interesting insights on the possible profitability and feasibility of the studied zone.

5. Application of the code: a case study of the Cesano-Sabatini area

This chapter presents a case study on the Cesano-Sabatini area, in which the assessment of the geothermal potential is done using the proposed code. The analysis has been done for two different depths, and for two different values of permeability constants, to compare the technologies and to find which one could work better. The chosen depths are 2 km and 3 km. The permeability of the rocks influences the mass flow rate, so the analysis considers the best and the worst scenario, therefore high and low mass flow rates. The permeability is set respectively at $10^{-14} [m^2]$ and $10^{-15} [m^2]$.

The number of production wells is set as default equal to two, so the mass flow rate doubles. Using this number for production wells and considering only one injection well, neither hydraulic nor thermal interactions exist between them. This assumption is based on the choice of sufficient spacing between wells, since the dimension of one cell is 1 km², 1 km per edge. Using this distance between two wells, the drainage radius (r_e) results to be 200 meters. So, with the correct spacing, it is not a problem to have three wells in the same cell.

The thermodynamic cycles of the implemented power plants use as input the default value shown in Table 5, Table 7 and Table 8. The cogeneration option is not included in this analysis. The economic model has been applied without considering the incentives.

5.1. Geological setting of Central-Southern Italy

5.1.1. *Geodynamic and paleogeographic backgrounds of Central-Southern Italy*

The main areas of geothermal interest in Italy are, apart from Tuscany, almost all situated in Western Latium and in the Neapolitan area.

The geological setting of Central-Southern Italy is characterized by the presence of a complex arcuate thrust belt system belonging to the peri-Mediterranean orogenic belt, generated by the relative movements between the Eurasian and African plates and by the interactions with the Adriatic microplate (e.g., Dewey et al., (1989)[87]; Malinverno and Ryan, (1986)[88]; Doglioni et al., (1999)[89]; Faccenna et al., (2004)[90]; Rosenbaum and Lister, (2004)[91]).

In the Central-Southern Italy, the peri-Mediterranean orogenic belt is made up of three main segments, corresponding to the Central-Southern Apennines, the Calabrian Arc, and the Sicilian Maghrebides, while its internal side is marked by the Tyrrhenian Sea (Figure 34). Several geodynamic interpretations have been proposed to explain the development of this orogenic belt. A first group of interpretations consider a typical subduction related scenario controlled by an eastward-retreating west-directed subduction zone (e.g., Scandone, (1980)[92]; Malinverno & Ryan, (1986)[88]; Royden et al., (1987)[93]; Patacca et al., (1990)[94]; Doglioni, (1991)[95], Jolivet and Faccenna, (2000)[96]). Another group of interpretations consider instead both active or passive rift models in which the contractional deformation within the thrust belt is driven by the extensional processes along the Tyrrhenian side (Lavecchia (1988)[97]; Locardi and Nicolich, (1992)[98]; Lavecchia et al., (1995)[99]; Liotta et al., (1998)[100]; Lavecchia et al., (2003)[101]).

However, it is generally agreed that the three different orogenic segments reflect the original thickness and compositional lateral variation of the Mesozoic paleogeographic domains that were deformed during the growth of the thrust belt system. The Mesozoic paleogeographic configuration is ruled by the presence of the Ionian Oceanic basin and of the two associated passive margins located along its northern (Adriatic domain) and southern sides (African domain) and it may be considered the result of a continental rifting stage developed since Lower-Middle Triassic times following the breakup of

Gondwana and opening of the Neo-Tethys Ocean domains (Catalano et al., (2001)[102]; Ciarapica and Passeri, (2002)[103]). The mainly Meso-Cenozoic sedimentary covers originally deposited on the Ionian Oceanic basin and on the two conjugate Adriatic and African passive margins were off-scraped and incorporated respectively in the Calabrian Arc, Central-Southern Apennines and Sicilian Maghrebides. These sedimentary covers are mainly made up by clastics, evaporites, shallow water and pelagic carbonates, in the case of the two passive margins, and by deeper water terrigenous units and ophiolite bearing sequences, in the case of the Ionian basin (Casero et al., (1988)[104]; Catalano et al., (2001) [102]; Fantoni & Franciosi, (2010)[105]). Since Oligocene times, the presence in the Central-southern Apennines of both foredeep and piggy-back basins progressively younger towards the foreland clearly document the migration of the foreland flexure and of the thrust belt front (Patacca and Scandone, (1990)[94]; Patacca and Scandone, (2004)[106]). Starting from Middle-Late Miocene times, the progressive foreland-ward migration of the accretionary prism was accompanied by crustal stretching of the internal sectors of the belt, that is also progressively younger toward the foreland, and that eventually led to genesis of the Tyrrhenian Sea (Malinverno and Ryan, (1986)[88]; Jolivet & Faccenna, (2000)[96]; Scrocca et al., (2012)[107]). The extensional tectonics is accommodated by high-angle hinterland-dipping normal faults and by low-angle foreland-dipping normal faults as documented by the integrated interpretation of surface geological data, seismic reflection profiles and seismological information (e.g., Doglioni et al., (1999)[89]; Chiaraluce et al., (2007)[108], Brozzetti, (2011)[109]; Lavecchia et al., (2017)[110], and references therein). In this context, the Kabilo-Calabride metamorphic basement slices outcropping in the Peloritani Mountains (Calabrian Arc and north-eastern Sicily) may be considered relics of the stretched and boudinated Alpine belt (e.g., Carminati et al., (2004)[111]).

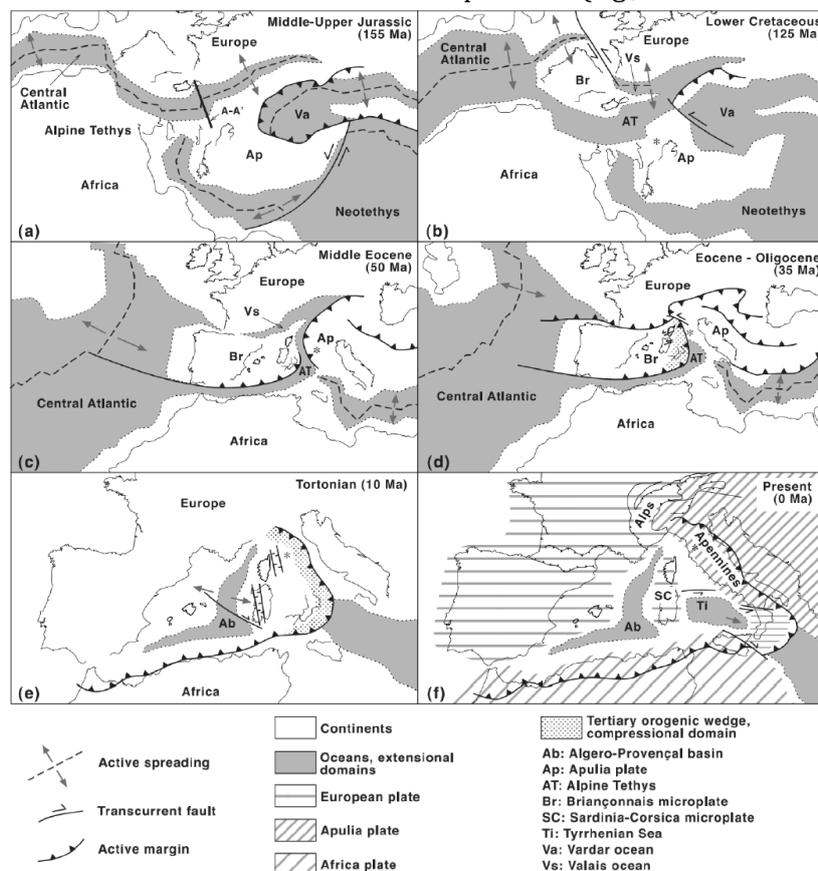


Figure 34. Plate reconstruction for the western Mediterranean area (after Carmignani et al. (2004)[111]).

In the internal sector of the Apennine belt, the eastward migrating extensional tectonic is responsible of the lithospheric thinning since the Middle-Upper Miocene. The lithosphere-asthenosphere boundary, which roughly corresponds to the 1250 °C isotherm, moved up from an initial value of 90 – 110 km to

the actual depth of 45 km. In the Tyrrhenian side of peninsula, the asthenosphere upwelling controlled during the Neogene – Quaternary period the emplacement of magmatic crustal bodies at relatively shallow depths (6–8 km) within the Earth's crust as well as the surface volcanic manifestations [112], [113], [114], [115].

In Figure 35, the main structural elements of Italian peninsula and the volcanic centres are displayed. The thrust front separates the two sectors of the Italian peninsula where are active compressional (to the East) and extensional (to the West) tectonic regimes. The volcanic centres are localized along the westernmost side of Italian territory and refer, from the North to the South, to the Latium Magmatic Province, the Campanian Magmatic Province, the Mount Etna and several eruptive centres in the Tyrrhenian Sea.

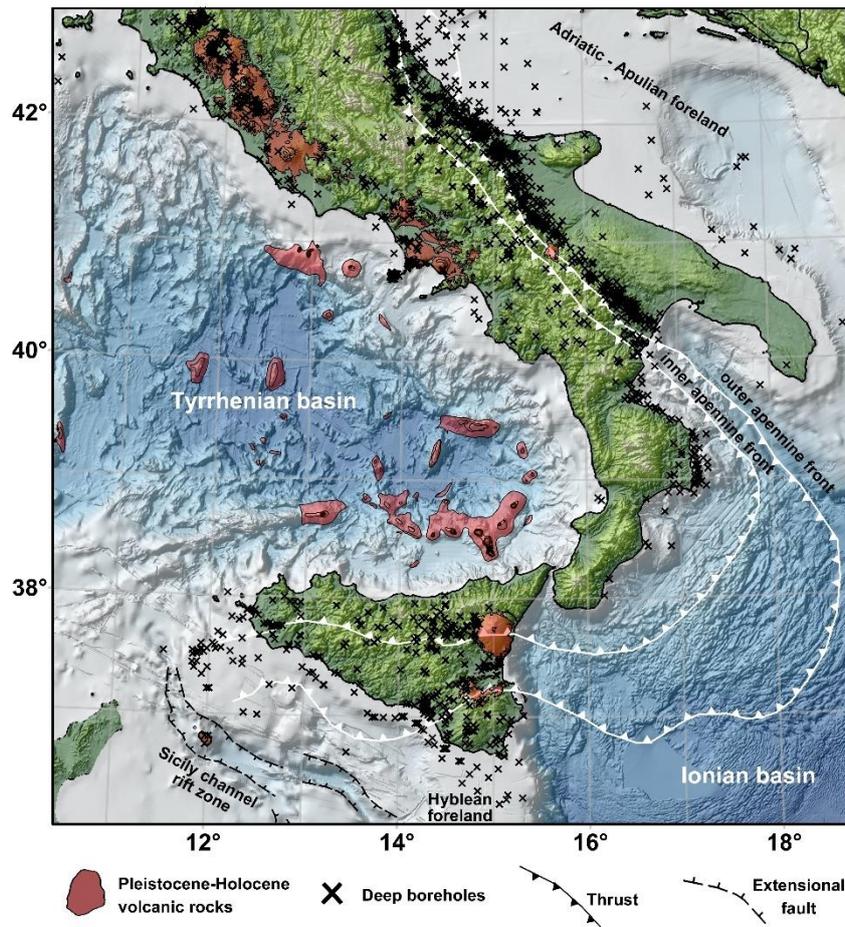


Figure 35. Borehole locations (black crosses) from BNDG (Trumpy and Manzella, 2017) in Southern Italy. The volcanic areas and the main structural elements of Italian peninsula are displayed.

The subsurface temperature field is constrained by a huge temperature dataset from deep exploratory wells. In Southern Italy a total of 1088 boreholes (Figure 35) are included in the Italian National Geothermal Database [116]. The final compilation of thermal data counts a total of 9313 temperatures measured down to a maximum depth of 6940 m. A pronounced and widespread thermal anomaly characterizes the peri-Tyrrhenian zone, which reflects the complex structural setting as well as its geodynamical and magmatic evolution [117], [118], [119]. Since the permeability in the shallow crustal levels, dominated by the carbonate platform units, can exceed the threshold value for a well-organized hydrothermal convection ($\sim 10^{-16} \text{ m}^2$), the measured terrestrial heat flow data represent the sum of two components: the purely conductive component plus the convective component. Where the

favourable geological conditions lead to the development of a deep-seated hydrothermal systems, the observed heat flow can exceed $10^2 \left[\frac{mW}{m^2} \right]$ and in young magmatic areas, it can reach $10^3 \left[\frac{mW}{m^2} \right]$ [120].

5.1.2. Geological model and temperature distribution of Lazio Region

The geothermal areas of Latium coincide with the volcanic centres of the Mt. Vulsini, Mt. Vico-Cimini and Mt. Sabatini (Figure 36). From a hydrogeological point of view, although characterised by both lateral and vertical anisotropies, the main regional aquifer is represented by the Mesozoic carbonate successions [121], [122]. Along the axial sector of the chain those units crop out and the succession are composed by a stack of the Adriatic (or Apulian) carbonate units overthrust by other carbonate units, originally located in a more internal paleogeographic position. These units are made up by Upper Triassic-Middle Miocene shallow water carbonates, transitional shelf-to-basin deposits, and pelagic sequences, namely the Apennine carbonate platform, and the shallow water to pelagic carbonates of the Tuscan and Umbria-Marche successions. It should be noted that, due the structural and stratigraphic complexities, the surface that identifies the top of the regional geothermal carbonate reservoir corresponds to different geological formations. The extensional regime active since Miocene time led to the development of dominantly NW- and minor NE-extensional faults, which arranged the Meso-Cenozoic carbonate units in a horst and graben pattern Figure 37. The reservoir is generally at a depth of 2500 m. The deep geothermal fluids are mainly alkaline-chloride with TDS content ranging from 7 to $10 \left[\frac{g}{l} \right]$ and variable amount of CO_2 and H_2S ; concentrated Li-rich brines may be locally present.

In this context, post-orogenic magmatism related to the Tyrrhenian back-arc extension occurred during the Pleistocene. Volcanism of both acidic (Tuscan Magmatic Province: Cimini, Tolfa and Ceriti districts) and alkaline-potassic type (Roman Comagmatic Province: Vulsini, Vico, Sabatini and Albani districts) covered the greater part of the Western Latium. The volcanic activity of Vulsini district (0.6 – 0.1 Ma old), Vico-Cimino district (1.3 – 0.09 Ma old), Sabatini district (0.8 – 0.07 Ma old) and Albani district (0.6 – 0.03 Ma old) generated the volcano-tectonic or explosive morphological depressions of the lakes of Bolsena, Vico, Bracciano and Albano, respectively. During the 1970s and the early '80 many wells were drilled in the area by the joint-venture Enel and Agip. The exploratory activities lead the identification of the Latera, Torre Alfina and Cesano geothermal fields. A further drilling during the early '90 was negative; as a matter of fact, only a productive structure was found in the Marta area (to the South of Bolsena Lake) and a few positive wells were drilled in the Mt. Sabatini area. The potential geothermal reservoir, hosted in the Meso-Cenozoic carbonate units, presents widespread self-sealing phenomena related to the recent thermo-metamorphic and hydrothermal events. The deep boreholes drilled the following geological formations (from the top to the bottom):

- Post-orogenic complex (Neogene - Quaternary), made of clays, sands and, secondarily, conglomerates at the base of the volcanites;
- Complexes in the Ligurian facies, made of arenaceous, argillaceous or marly-calcareous formations (Cretaceous - Oligocene);
- Complexes of the Tuscan or Umbria-Marche Series, made of carbonate, with an anhydride-dolomitic formation and successive prevalent carbonate formations (Upper Trias - Eocene).

The underlying regional metamorphic basement of the Paleozoic age (phyllites and quartzites) outcrops only to the west of the Mt. Vulsini. The crystalline basement (Figure 37), resulting from the geophysical modelling of aeromagnetic data [40], limits downward the geothermal reservoir which thickens eastward.

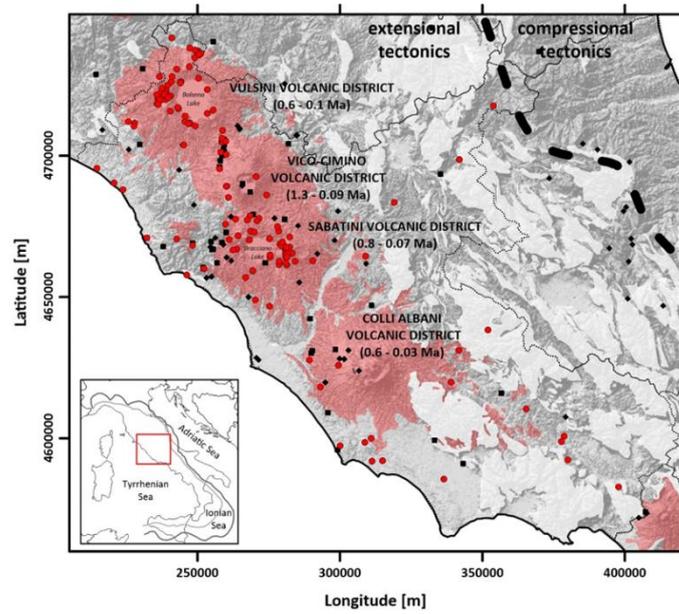


Figure 36. Topographic map of Lazio Region

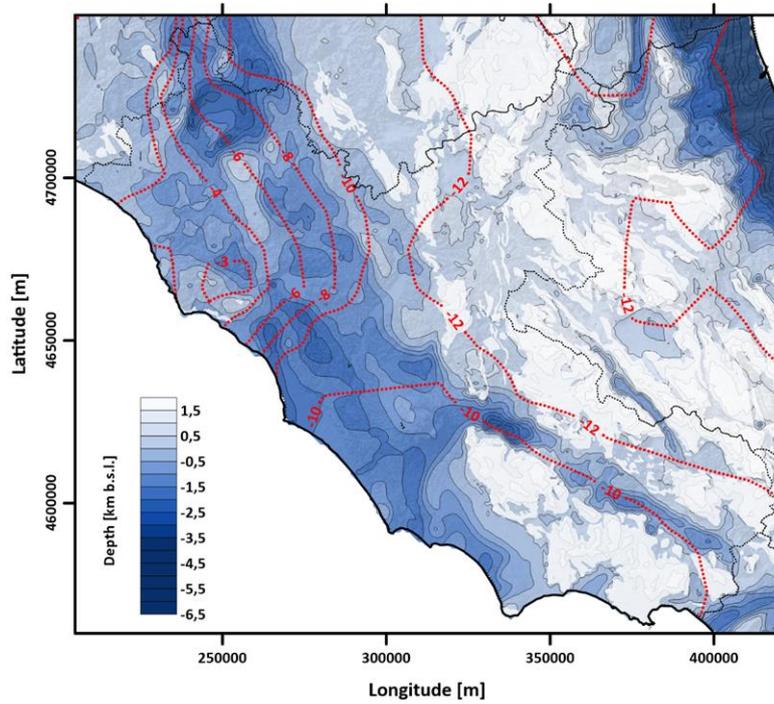


Figure 37. Depths (km below sea level) of the top (blue) and bottom (red) surfaces describing the geometry of the geothermal reservoir.

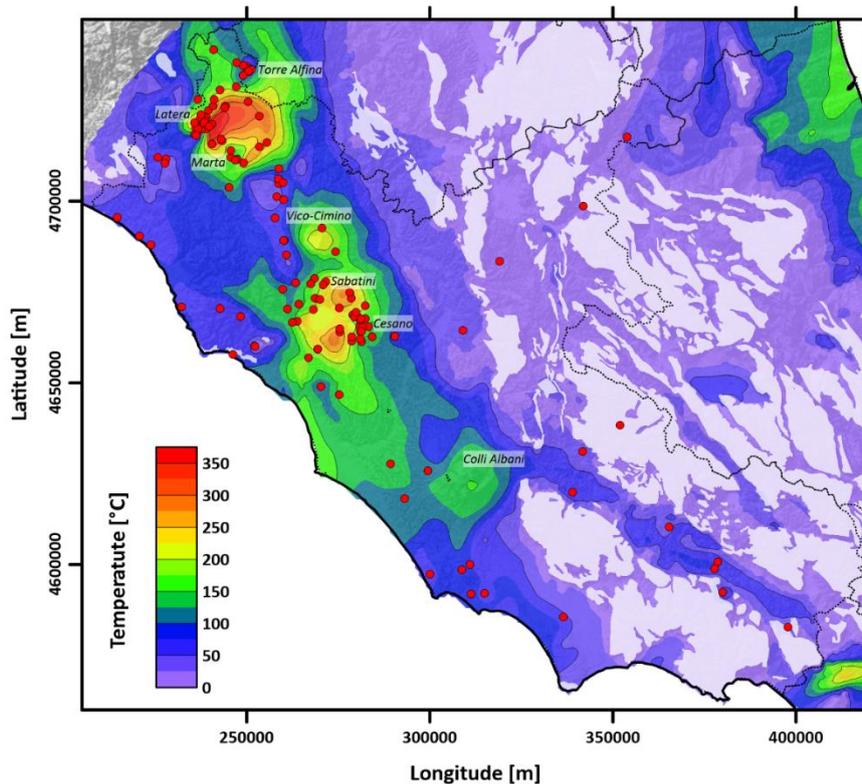


Figure 38. Temperature distribution at the top of the geothermal reservoir.

The integrated analysis of different interdisciplinary surveys led to the determination of the potentially favourable areas, generally located near the volcanic centres around which a widespread thermal anomaly is present. Thermal anomalies were reconstructed in the framework of the Geothermal Atlas of Southern Italy, led by IGG-CNR, based on more than 150 wells drilled to depths between 100 m and 3000 m. The distribution of the temperature at the top of the geothermal reservoir is shown in Figure 37. A geothermal gradient anomaly of about 55 – 65 °C/km (about twice the average geothermal gradient in Italy) delimits the area including all the Latium volcanoes. Limited but more intense anomalies involve the alkaline-potassic volcanic centres and are associated with magmatic bodies located in the upper crust, at depths of 2 – 6 km. The highest temperature values (higher than 200°C) were recorded in the areas of most recent volcanic activity. The peripheral sectors display medium-low temperatures (around 100 °C) and decrease eastward due to many interactions with cold surface hydro-geological circulation. The hydrological recharge area is identified in the Apennine chain where the carbonate reservoir units' outcrop.

5.1.3. The Cesano-Sabatini case study

The results of the optimized thermal model of the Cesano-Sabatini geothermal field are shown in Figure 39 (a). Boreholes are employed as control points in order to verify the site-specific simulated thermal profiles and the measured temperatures, their position is shown in Figure 39 (b). The thermal anomaly in the sector of the volcanic centre is well fitted highlighting the occurrence of a high-temperature geothermal reservoir characterized by temperatures higher than 200 – 250 °C at depths greater than 2 km. The thermal anomaly decreases as one moves away from the volcanic centre to the East, where the thermal field is influenced by cold lateral fluid flow from recharge areas in the Apennine region.

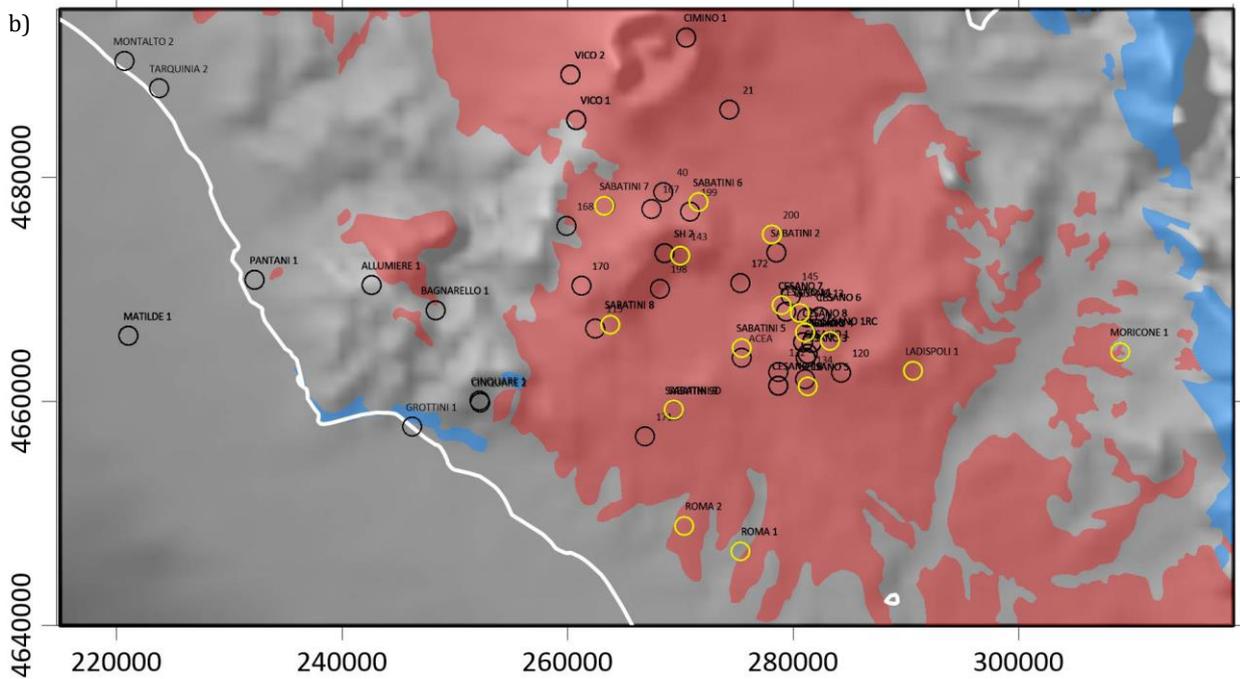
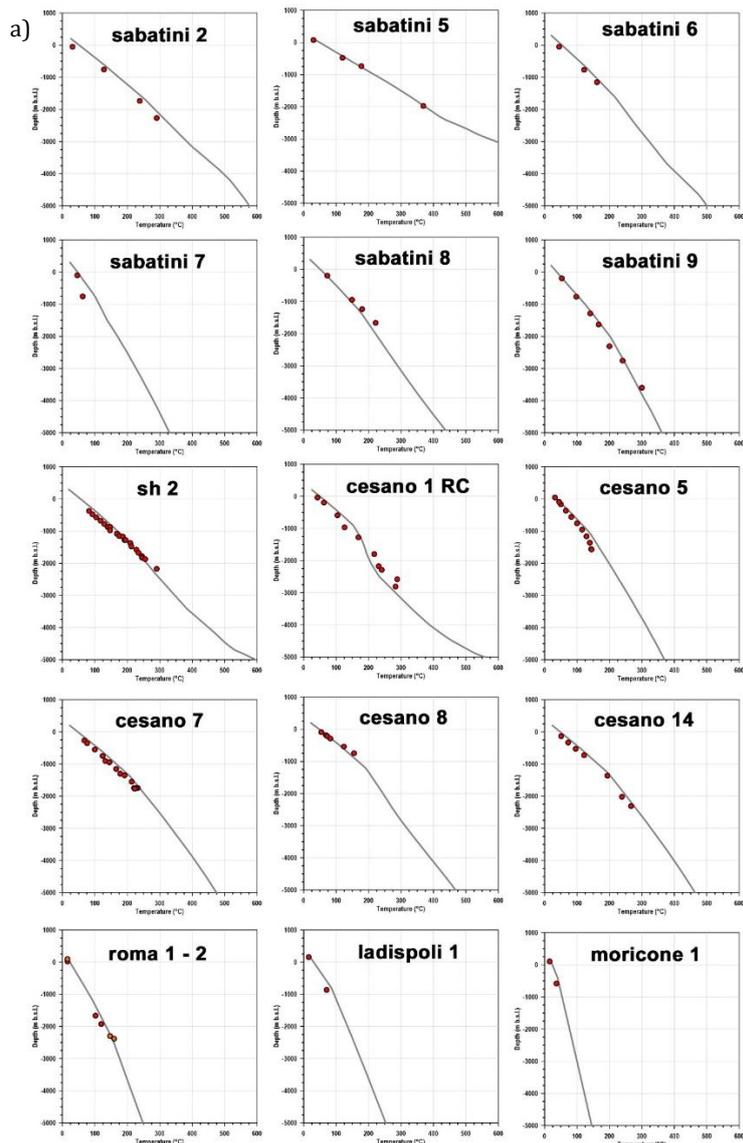


Figure 39. Temperature profiles (a) and position (b) of the boreholes

5.2. Temperature and pressure distribution for code's simulation

Following the methods outlined in chapters 4.1.1 and 4.1.3, it is possible to find the temperature matrix and then the pressure matrix for the studied area. A mask is applied to the pressure matrix to indicate the location of the reservoir. Furthermore, the values at the desired depth are extracted and used for the evaluation of the production temperature, pressure and mass flow rate. The temperature and the pressure at this depth are assumed to be constant over time, as is the mass flow rate. In contrast, production temperature and pressure change during the plant's lifetime. This happens because the production temperature evaluated through Ramey's model varies through the years, and then also production pressure.

Inside the code, there is a control on the value of the pressure at the wellhead, since it indicates whether there is a reservoir or not. So, the evaluations are done only in case the production pressure is higher than zero, because sometimes could happen that the production pressure inside the reservoir is lower than the hydrostatic one, resulting in a negative value. Moreover, if the pressure exceeds 35 bar, it is capped at this threshold, as geothermal power plants usually do not operate at excessively high pressure. However, in Figure 40 is possible to see the behaviour of both temperature and pressure at production wells. They both increase faster in the first seven years of operation compared to the remaining ones, because initially the temperature difference is higher, and hence the fluid exchanges more heat with the surrounding rocks. These increments in their value make the power output higher at the end of the life of the power plant, so the geothermal potential reported refers to the 30th year. This trend of the power is a consequence of the previous assumption of keeping constant the pressure and temperature in the reservoir. If they decrease in time as normally happens, this constant increment does not verify.

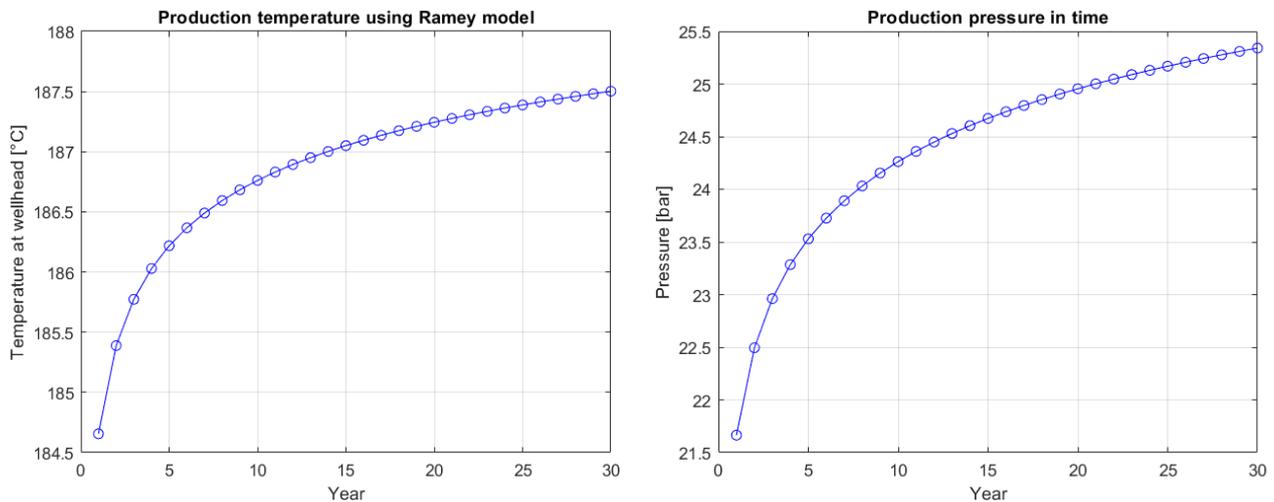


Figure 40. Production temperature and pressure during the plant's lifetime. Author's own elaboration

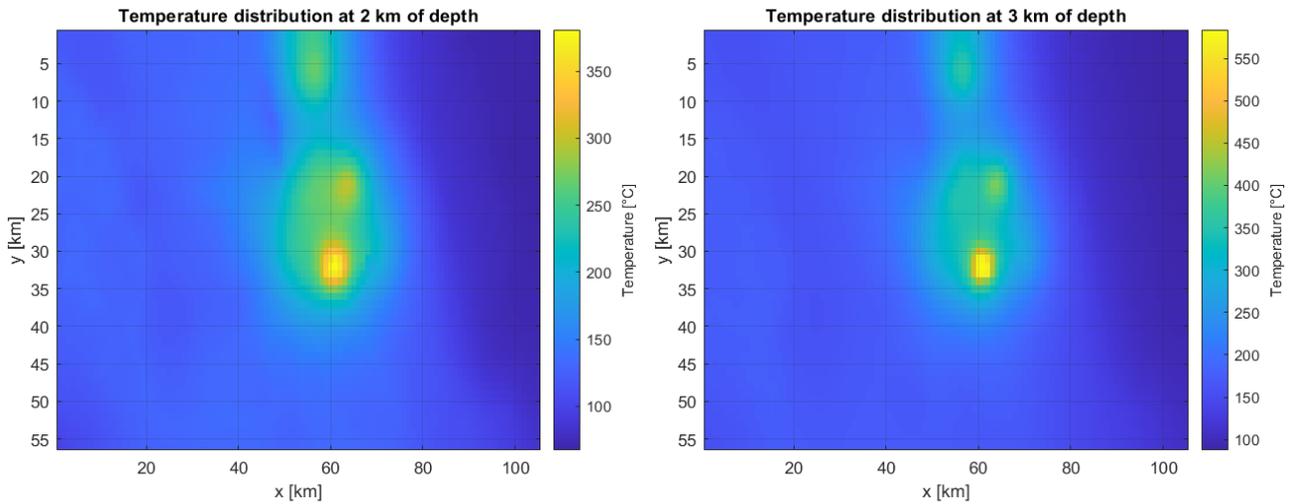


Figure 41. Temperature distribution at 2 and 3 kilometers depth. Author's own elaboration

The hot spot is located in the upper-central part of the Figure 41, and characterizes the temperature distribution of the area, since there are some magmatic intrusions which increase the temperature, which at 3 km depth results to be higher than 550 °C in the middle of the hot spot. Around that zone, the temperature is higher and more favourable to exploitation. Then on the right of the hotspot, there are much lower temperatures since there are the Apennines Mountain chains, while moving to the left and to the bottom the temperature gradually decreases.

The pressure distribution underlain that the presence of the hotspot reduces considerably the pressure, and this is more accentuated at higher depths. Then on the right, where there are the Apennines Mountain chains, the pressure is significantly higher.

Moreover, the pressure distribution is also indicative of the reservoir presence, and how it is possible to see from Figure 42 it is larger at 3 kilometres. The region where it does not exist corresponds to zero values of pressure.

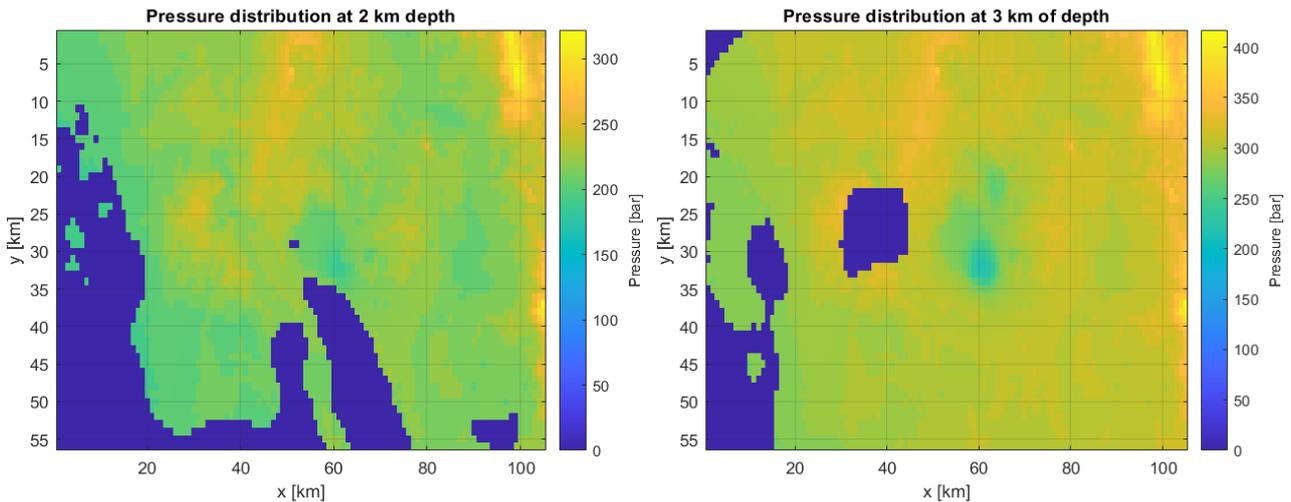


Figure 42. Pressure distribution at 2 and 3 kilometers depth. Author's own elaboration

5.3. Results

The first type of power plant analysed is the dry steam power plant, due to the high temperature highlighted by the temperature distributions at the two selected depths.

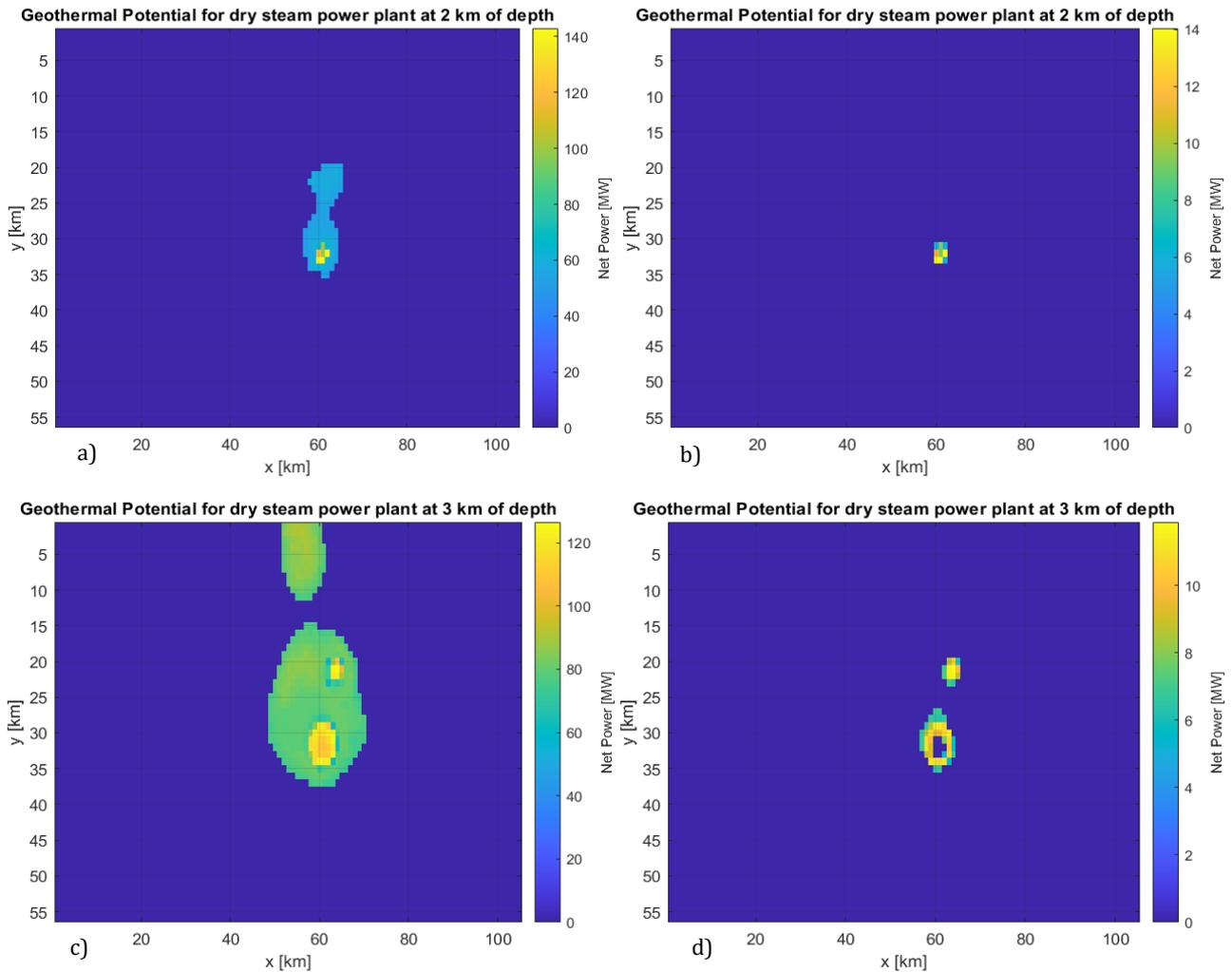


Figure 43. Geothermal potential of dry steam power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 . Author's own elaboration

The geothermal potential for this plant is limited to the zone around the hot spots, particularly at 3 km of depth and with high permeability, as it is possible to see in Figure 43 (c). With low permeability the zone from which is possible to produce power is limited Figure 43 (d). The same happens at 2 km of depth, but the region is further restricted, down to the minimum one reported in Figure 43 (b). From a depth of 2 km the maximum power output results to be higher, this depends on the amount of heat exchanged during the ascent of the geothermal fluid, which is lower starting from that depth, resulting in a higher temperature at the production wells.

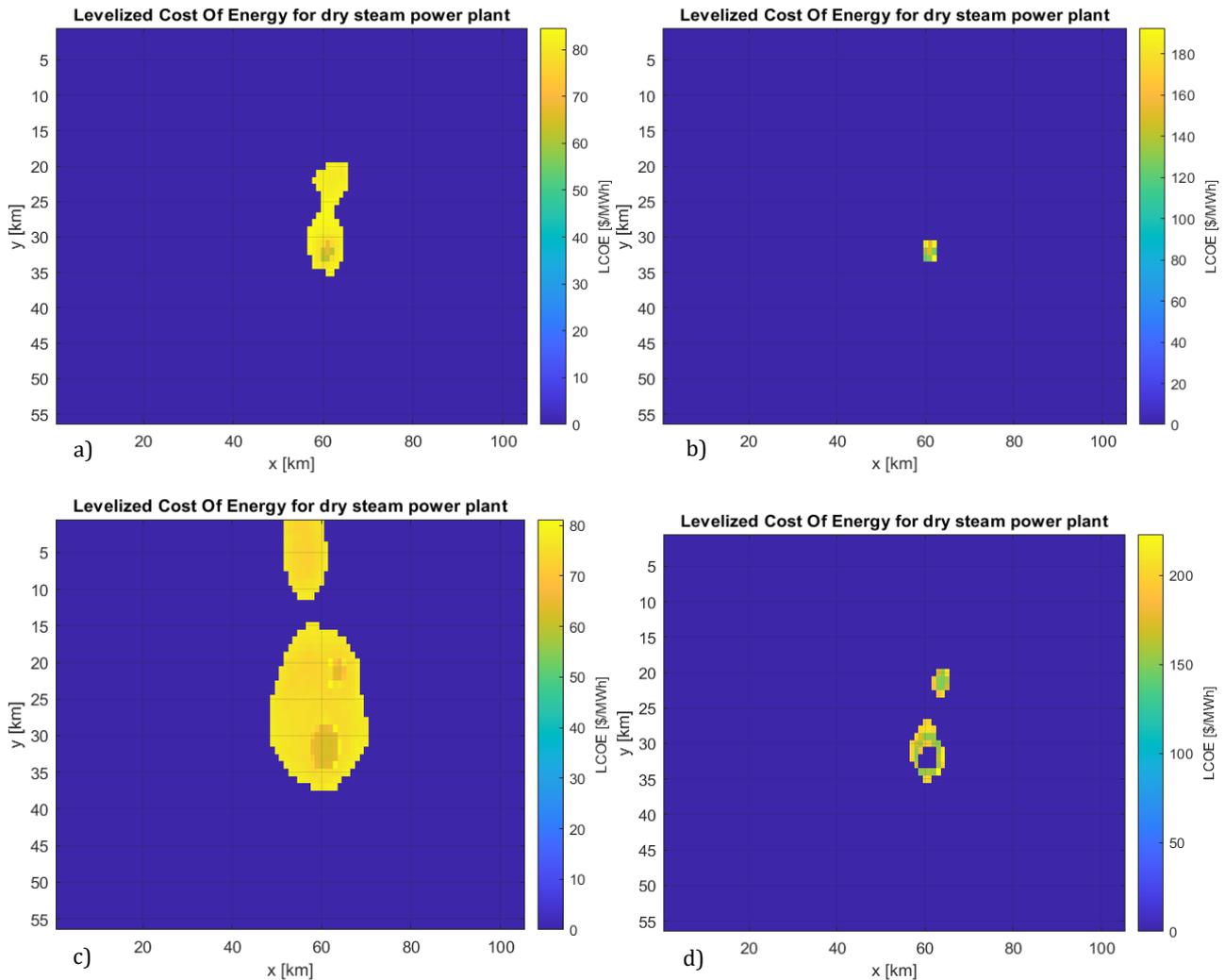


Figure 44. Levelized Cost of Energy of dry steam power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 . Author's own elaboration

The LCOE follow the distribution of the geothermal potential, and obviously, it is lower where it is possible to produce more power. Anyway, how it is possible to read in Table 10, the range covered in case of high permeability at both depths is in agreement with the value reported by IRENA (2020) [78], in which the weighted world average LCOE is $71 \left[\frac{\$}{MWh} \right]$, but considering all the different types of plants. For the low value of permeability, the LCOE is more than doubled, which makes clear the low profitability of the power plant with that condition. This is also better understood in Figure 45 (b-d), in which with low permeability the power plant after 30 years of operation does not have a payback time, and so the cash flow is still negative. On the contrary, the dry steam technology results to be profitable in case of high permeability at both depths, Figure 45 (a-c), with high revenues at 2 km, thanks to the higher power produced. However, it must be noticed that the incomes at 2 km are lower around the hot spot, with only a few spots of high profit, while at 3 km the profitability is more spread and constant, and obviously with peaks near the central hot spot.

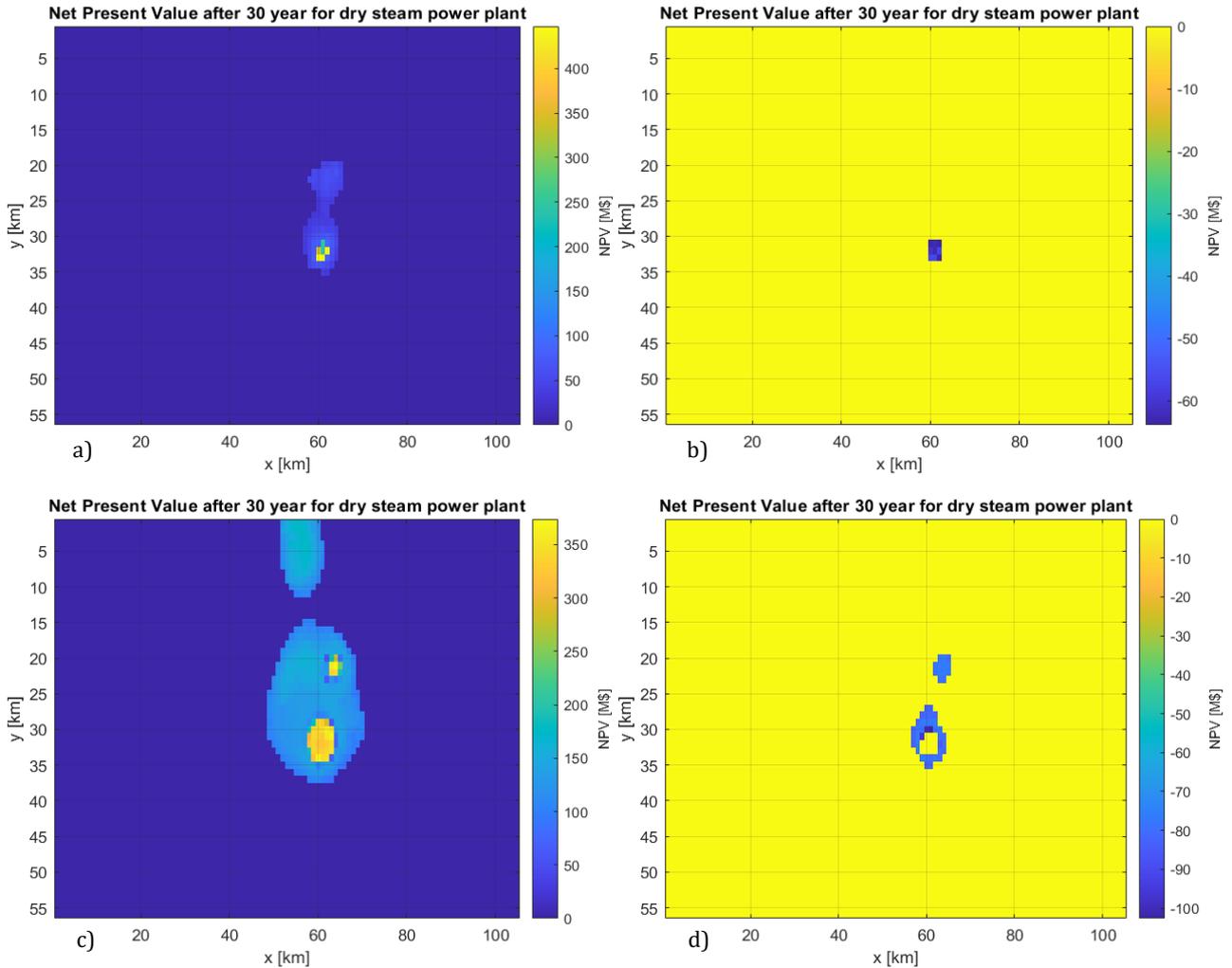


Figure 45. Net Present Value at end of plant's lifetime of dry steam power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m². Author's own elaboration

Table 10 reports the range of the LCOE and the performance parameters in the studied cases. The specific steam consumption and the utilization efficiency result similar at 2 km, while at 3 km the maximum SSC is notably higher, therefore the efficiency decreases. Similarly, the minimum SSC at that depth is a little bit lower, resulting in a higher maximum efficiency.

Z [m]	K [m ²]	LCOE [\$/MWh]	SSC [kg/kWh]	Efficiency [-]
2000	10^{-14}	61.68÷84.56	4.99÷5.88	0.595÷0.608
2000	10^{-15}	126.41÷192.56	5.16÷5.74	0.597÷0.605
3000	10^{-14}	62.29÷81.16	4.04÷5.88	0.595÷0.627
3000	10^{-15}	148.43÷223.16	4.98÷12.15	0.423÷0.608

Table 10. Dry steam power plant results

Continuing the analysis with the single flash power plant, the geothermal potential results are still high in some cells around the central hot spot, but lower than the one evaluated using the dry steam technology, but it is more spread within the area. Particularly, in Figure 46 (c) it is possible to see that where there is the geothermal fluid, it is possible to produce between 5 to 15 MW of power, while in Figure 46 (a) these range drop to 2.5 to 10 MW.

In the simulation of this type of plant, there is a check on the vapour phase after the production well, before the geothermal fluid enters the separator. If the vapour fraction is higher than 0.5 the power is set at zero, considering the usual range of vapour fraction reported by DiPippo (1980) [51]. This causes

the difference in the middle of the potential maps at 3 km of depth, Figure 46 (c-d) since with low permeability the mass flow rate is lower and exchanges more heat with the surrounding rocks during the ascent. Then the temperature at the wellhead is much lower than the one in case of high permeability.

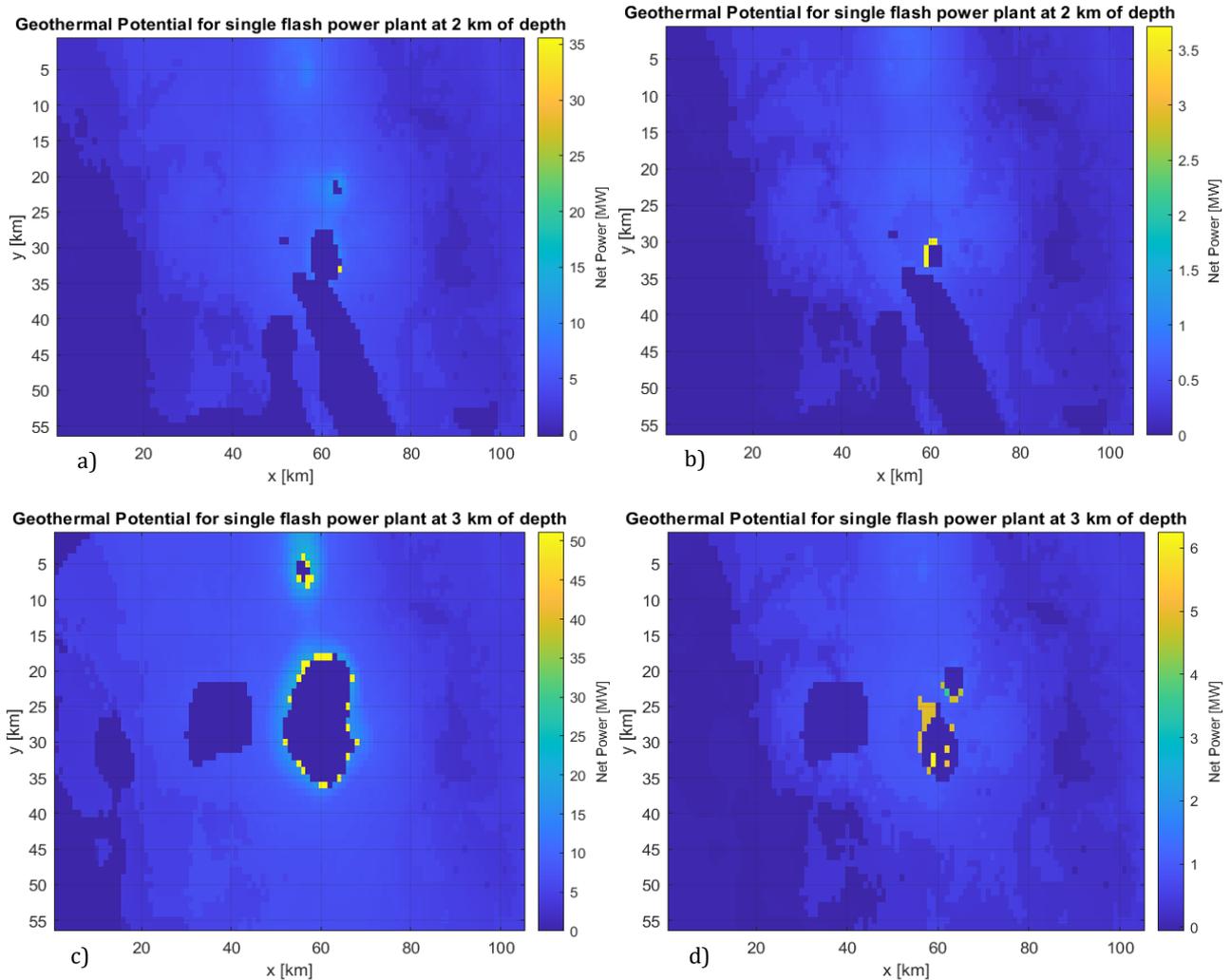


Figure 46. Geothermal potential of single flash power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 . Author's own elaboration

For what concern the LCOE, has been chosen to set a limit up to $500 \left[\frac{\$}{\text{MWh}} \right]$, to make more readable the maps, since some values were really high, as it is possible to see from Table 11. However, the locations where the LCOE has a low value are around the hot spot, while in the rest of the area, it quickly increases as the power produced decreases.

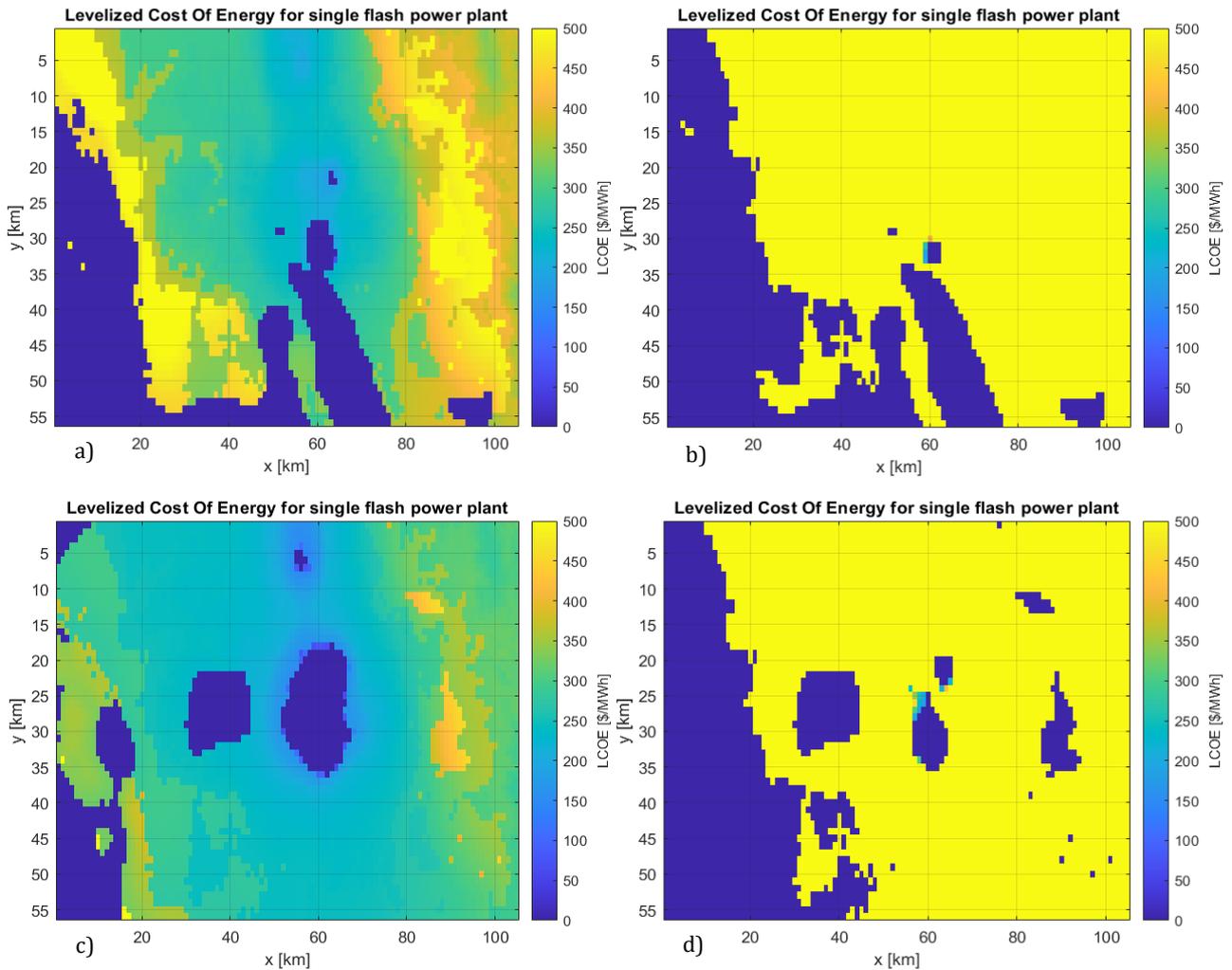
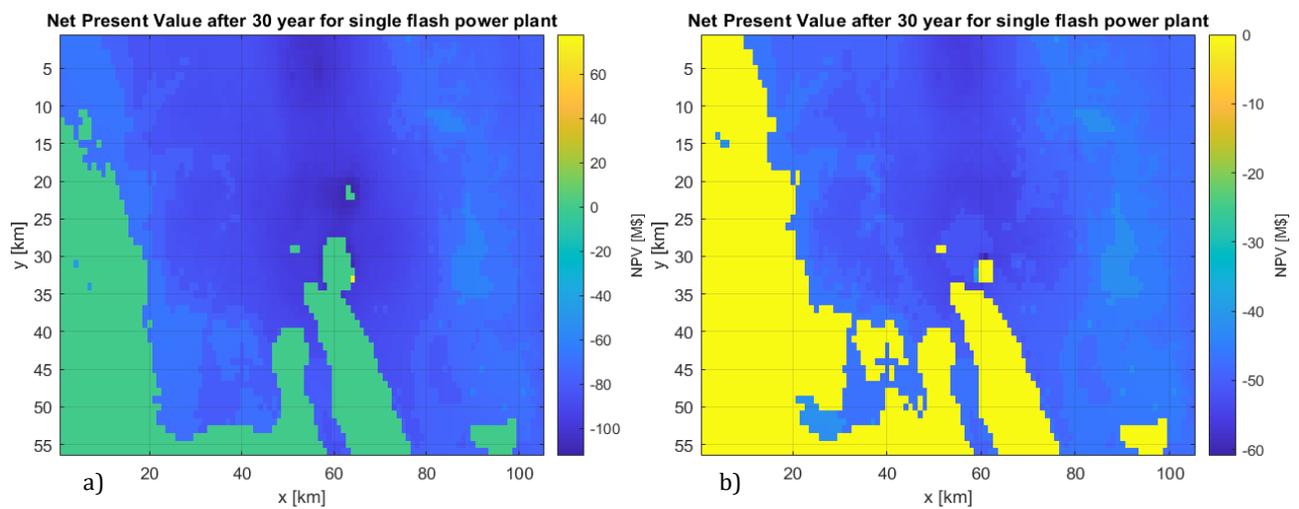


Figure 47. Levelized Cost of Energy of single flash power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 .
 Author's own elaboration

The NPV evaluation makes it more clear that the single flash power plant as a solution is not too feasible, also if it is possible to produce from a larger area. In fact, only a few locations have positive cash flow at the end of the plant's lifetime, and only in case of high permeability.



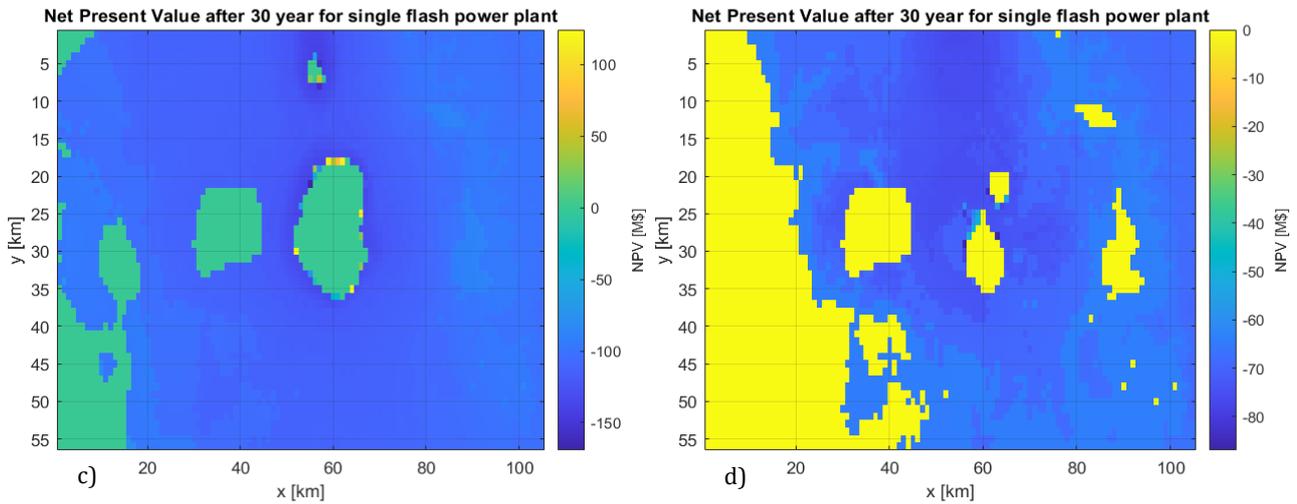


Figure 48. Net Present Value of single flash power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m². Author's own elaboration

Table 11 reports the ranges of performance parameters and LCOE, and it is possible to see that differently from dry steam power plants they are wider. The LCOE is in general higher, but there are some cells in which it is comparable with the one in dry steam plant. The specific steam consumption results are constant in the area evaluated in all of the cases. The utilization efficiency is very low in some locations, the same where the LCOE is high, but it is still quite high in the other ones, also if lower than dry steam efficiencies.

Z [m]	K [m ²]	LCOE [\$/MWh]	SSC [kg/kWh]	Efficiency [-]
2000	10^{-14}	68.18÷4'397.12	8.688	0.002÷0.4189
2000	10^{-15}	191.03÷20'404	8.688	0.0004÷0.419
3000	10^{-14}	66.38÷936.22	11.986	0.0772÷0.419
3000	10^{-15}	187.52÷3'153.26	8.688	0.111÷0.432

Table 11. Single Flash power plant results

The last technology analysed is the binary power plant. Also for this analysis, there is a constraint on the vapour fraction at the production well, which must be zero, to ensure only liquid enters the heat exchanger. Other checks are done inside the code to ensure the correctness of the cycle, with particular attention to the heat exchange process.

Considering the temperature and pressure distributions, Figure 41 and Figure 42 respectively, it is obvious that this plant could not perform well in the studied area. However, it seems to be possible to produce power from a wide area, at both depths. Moreover, when the permeability has the lowest value, the productive area is not too small, and it has the same values of net power. Comparing Figure 49 (b-d) with Figure 43 (b-d) and Figure 46 (b-d), it is noticeable that the maximum power output of binary technology is only a third of the one in case of high permeability, while for dry steam and single flash power plants, it is a tenth.

This underlines a better use of the resource by the binary plant when there is low permeability and therefore low mass flow rate.

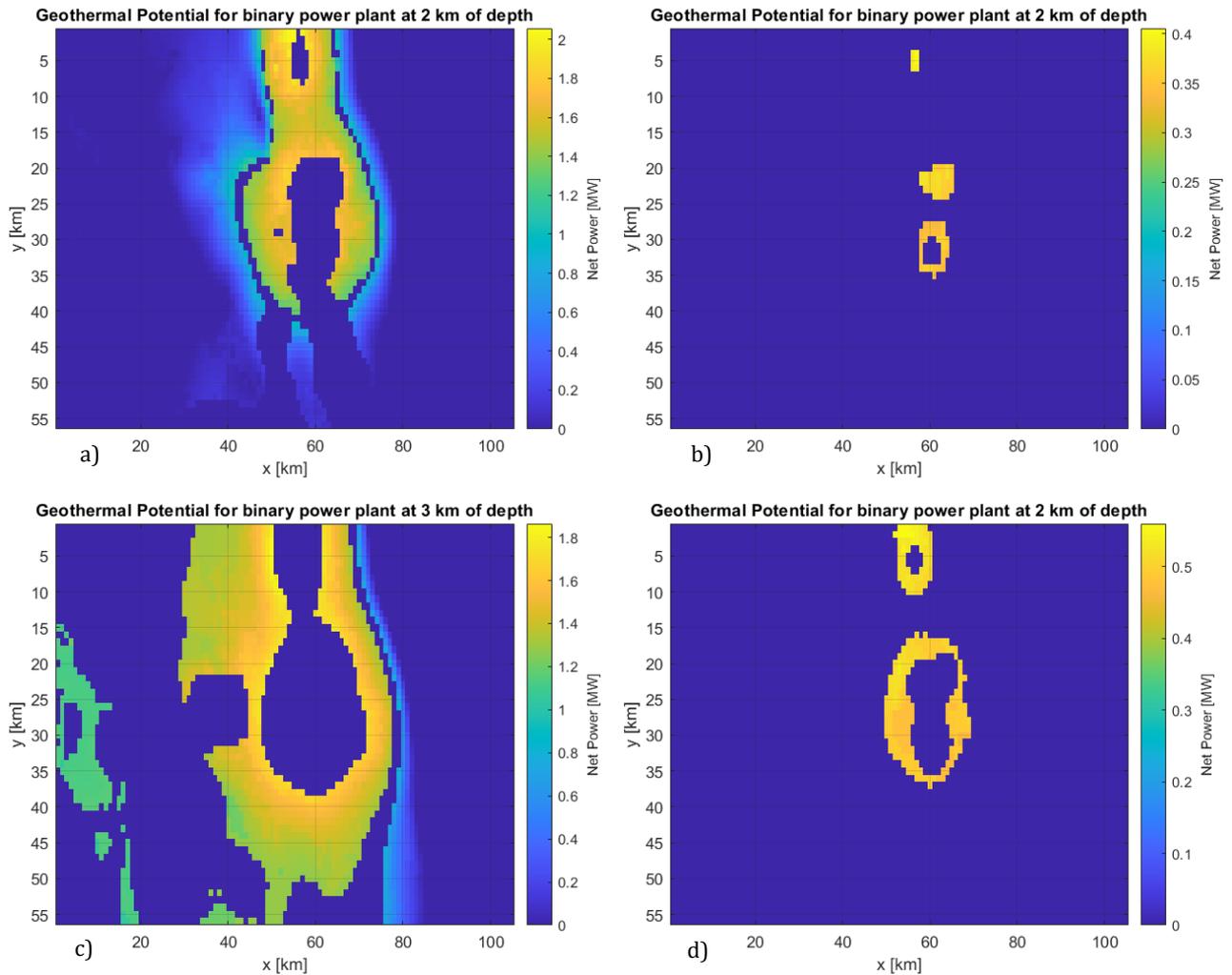


Figure 49. Geothermal potential of binary power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 . Author's own elaboration

Figure 50 reports the LCOE distribution, and as previously done a limit is set up to $500 \left[\frac{\$}{MWh} \right]$. It is relatively low only along the central zone, at both depths and only for the higher value of permeability, but the calculated value is high in any case.

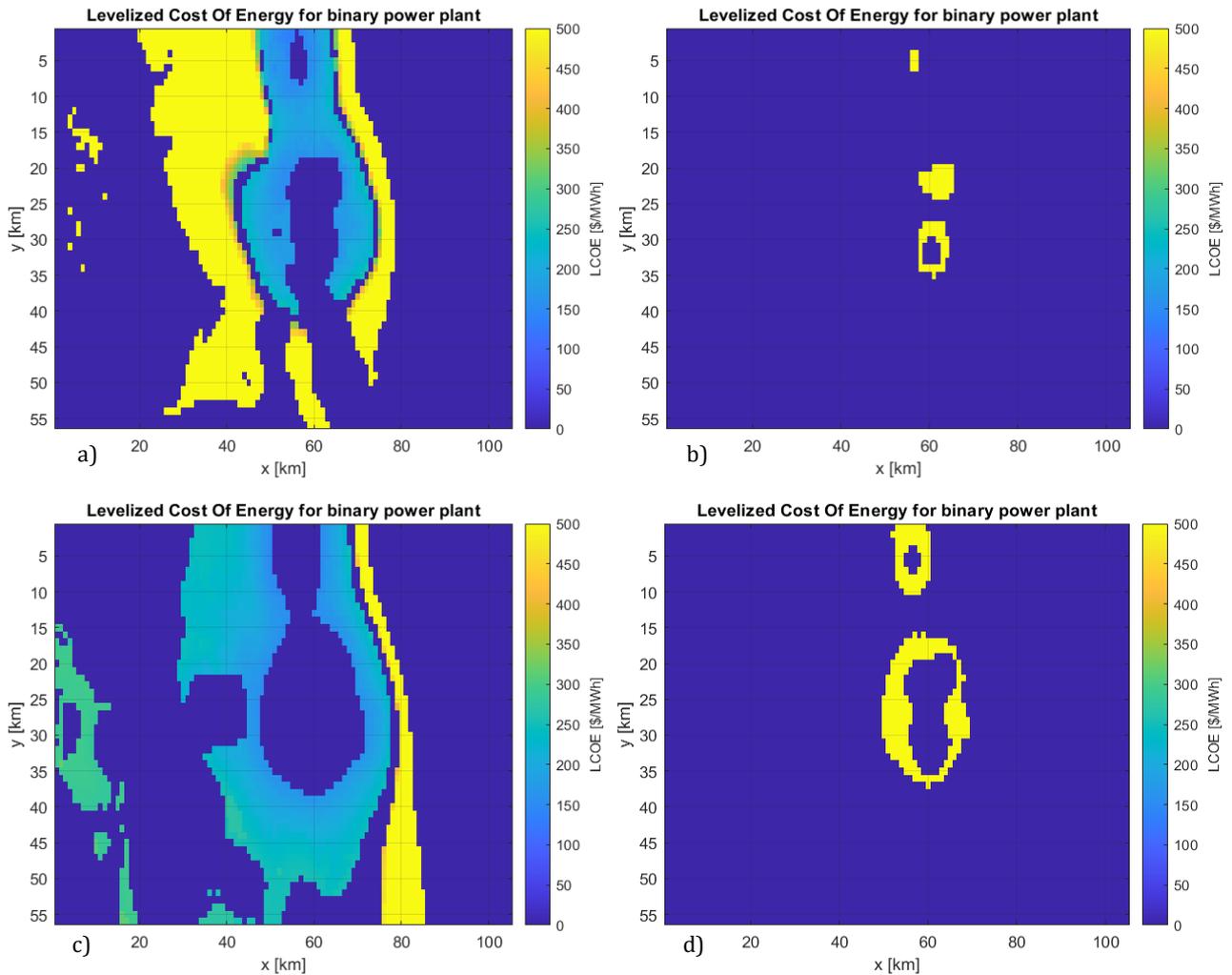
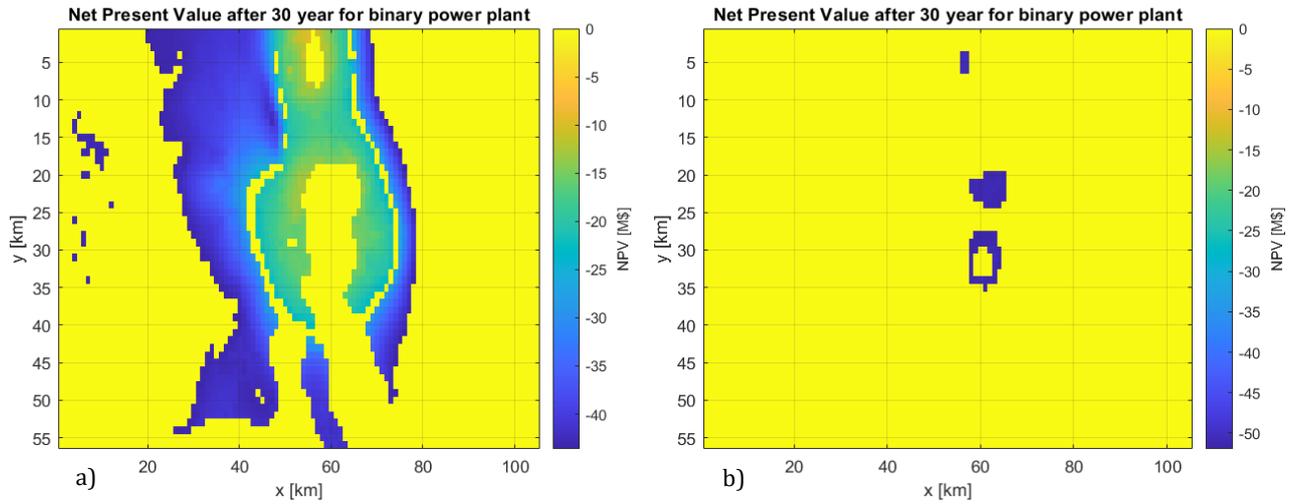


Figure 50. Levelized Cost of Energy of binary power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m². Author's own elaboration

Figure 51 shows the net present value and it is always negative in the studied situations, as expected due to the low power output, also if it is constant in time, which brings low incomes.



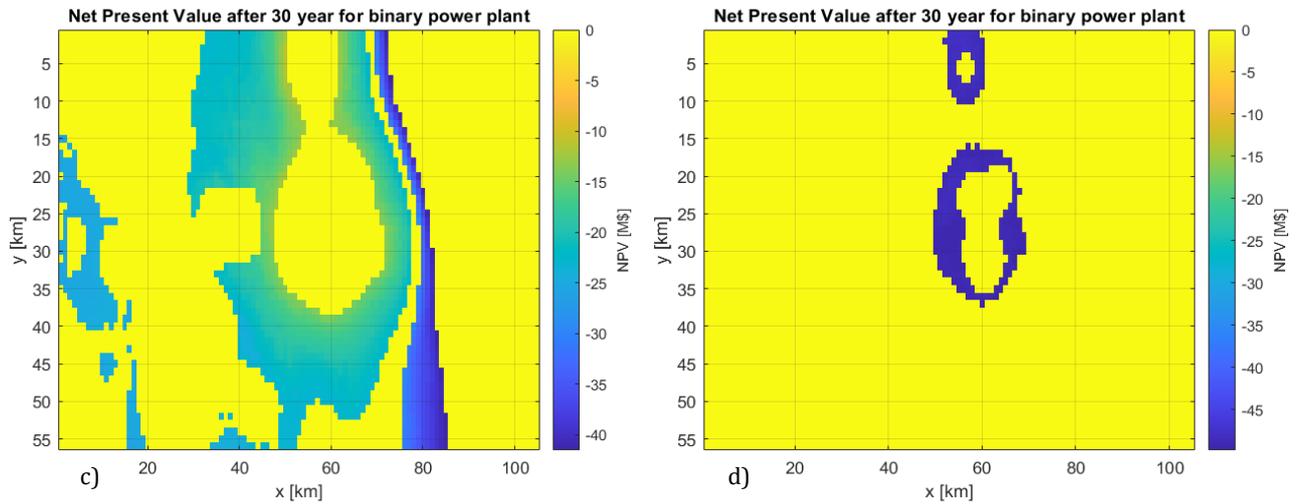


Figure 51. Net Present Value of binary power plant at 2 and 3 kilometers, with permeability of 10^{-14} and 10^{-15} m^2 . Author's own elaboration

Table 12 reports all the minimum and maximum values for the performance parameters and the LCOE since it is limited in Figure 50. The performance parameters for low permeability have a lower range since the power produced is inside a small area, and the value in each cell differs less.

Z [m]	K [m ²]	LCOE [\$/MWh]	SFC [kg/kWh]	Efficiency [-]
2000	10^{-14}	131.44÷312'254	88.83÷21'960	0.00036÷0.086
2000	10^{-15}	1'218.6÷1'500.87	89.09÷92.33	0.0782÷0.0811
3000	10^{-14}	144.94÷55'446	123.91÷6'804.1	0.0011÷0.0616
3000	10^{-15}	860.9÷1'088.82	89.12÷95.25	0.0758÷0.0811

Table 12. Binary power plant results

Considering all the simulations done with the three types of plants, it is possible to affirm that the studied area is feasible to be exploited using dry steam and partially single flash technologies. The first one is already economically viable when there is high permeability, and it could become viable for low permeability considering incentives. On the contrary, single flash plants are feasible only in a few locations around the central hot spot, and only for high permeability. Considering the incentives maybe some other locations could become feasible with that condition, but difficultly in case of low permeability, it becomes a usable technology. It must be underlined that the net power strongly depends on the separation pressure, which for the case study is set at 6.2 bar as suggested by Zarrouk (2014)[123]. So, with different values maybe it is possible to increase the power produced and make feasible this type of plant. For what concern binary power plants, they are not feasible under the studied conditions.

6. Conclusion and future developments

The geothermal potential evaluation is fundamental for assessing the feasibility of installing a geothermal power plant in a specific location. In this context, the proposed code wants to be the starting point for a new method to evaluate the geothermal potential, combining time-variable soil conditions with the thermodynamic cycle of a power plant. It aims to provide not only the net power produced over a year but also the potential economic return after the plant's lifetime and the levelized cost of energy. Currently, the variability of the subsoil conditions is not fully implemented, although the three proposed thermodynamic cycles have been completed and validated, as reported in chapter 4.6. Additionally, other types of technologies could enhance the efficiencies of flash and binary cycles, such as increasing the number of flashing processes or introducing regenerative heat exchange.

There are also opportunities to combine different types of plants. For instance, a binary cycle can be utilized after a dry steam or flash cycle, using geothermal fluid as a heat source post-turbine expansion. The application of the code to the case study of the Cesano-Sabatini area confirms its potential for evaluating geothermal potential. Preliminary observations of temperature and pressure at different depths suggested that a dry steam plant would be the most favourable option. This consideration was validated by calculation using the code, which also reports a favourable economic return and competitive levelized cost of energy for this technology.

In conclusion, further developments of the code will generate a powerful tool to evaluate the geothermal potential, aiding in the selection of optimal location for extracting geothermal fluid and reducing uncertainty related to the geothermal power plant projects.

Dedication

Questa tesi rappresenta la conclusione del mio percorso universitario, che è stato problematico, stancante, stressante e pesante, ma altrettanto bello e divertente, e più di tutto è stato costruttivo. Mi ha insegnato ad affrontare problemi che ogni tanto sembrano irrisolvibili, a non fermarmi alle prime difficoltà, e a cambiare punto di vista per capire meglio le cose.

Tutto questo non sarebbe stato possibile senza i miei genitori, che voglio ringraziare per il supporto che mi hanno dato durante questa carriera universitaria, dandomi la possibilità di studiare senza troppe preoccupazioni.

Voglio ringraziare le mie sorelle. Grazie Giulia per avermi corretto le relazioni, per avermi sistemato i PowerPoint, per avermi ascoltato nell'esporsi, e grazie per avermi corretto le e-mail anche mentre eri a lavoro. Grazie Gaia per ascoltare anche tu le mie presentazioni, e per correggermi nella pronuncia dell'inglese, anche se ogni tanto lo sai meno di me. Grazie ad entrambe per esserci quando ne ho bisogno, e grazie per essere semplicemente le mie sorelle.

Ringrazio tutti i miei amici con cui ho condiviso questi anni, ed in particolare voglio ringraziare i miei compagni di università, senza i quali questo percorso sarebbe stato completamente diverso. Grazie a chi c'è stato dal primo anno e che mi ha accompagnato fino all'ultimo, Angela (wing) e Matte, e grazie a chi mi ha accompagnato in questi ultimi due anni e mezzo, in particolare grazie Ale, Matti, Andre, e Fabio, con cui ho passato infinite ore a fare progetti e ridere. Senza di voi sarebbe stato tutto più brutto.

Grazie Poli, è stato bello.

Bibliography

- [1] ENERGIT, "Quali forme di inquinamento sono causate dal petrolio?" Accessed: Sep. 06, 2024. [Online]. Available: <https://energit.it/quali-forme-di-inquinamento-sono-causate-dal-petrolio/>
- [2] W. By, A. Kagel, B. Bates, and K. Gawell, "A Guide to Geothermal Energy and the Environment," 2005.
- [3] J. Shere, *Renewable : the world-changing power of alternative energy*. New York, 2013.
- [4] "Renewable energy highlights Electricity generation by energy source Fossil fuels Nuclear Other non-renewables Pumped storage Renewables % Renewables % Variable Renewables," 2024.
- [5] IEA, "Geothermal power generation in the Net Zero Scenario, 2000-2030."
- [6] "Technology Roadmap: Geothermal Heat and Power." [Online]. Available: www.iea.org/about/copyright.asp
- [7] A. Toth, P. Szucs, E. Bobok, A. N. Toth, and Z. Fejes, "Methodology for Determining Geothermal Potential," 2014, doi: 10.13140/2.1.4679.5204.
- [8] E. Barbier, "Geothermal energy technology and current status an overview," *Renewable and Sustainable Energy Reviews*, vol. 6, no. 1, pp. 3–65, 2002, doi: [https://doi.org/10.1016/S1364-0321\(02\)00002-3](https://doi.org/10.1016/S1364-0321(02)00002-3).
- [9] H. N. Pollack, S. J. Hurter, and J. R. Johnson, "HEAT FLOW FROM THE EARTH'S INTERIOR: ANALYSIS OF THE GLOBAL DATA SET," 1993.
- [10] G. R. (Graeme R. Beardsmore and J. P. (James P. Cull, *Crustal heat flow : a guide to measurement and modelling*. Cambridge University Press, 2001.
- [11] International Renewable Energy Agency. and International Geothermal Association., *Global geothermal : market and technology assessment*.
- [12] P. Frankl, "Renewable Energy Essentials: Geothermal." [Online]. Available: www.iea.org
- [13] International Renewable Energy Agency, *RENEWABLE ENERGY STATISTICS 2024*. 2024. [Online]. Available: www.irena.org
- [14] K. Dekkers *et al.*, "Resource Assessment: Estimating the Potential of a Geothermal Reservoir."
- [15] A. E. Ciriaco, S. J. Zarrouk, and G. Zakeri, "Geothermal resource and reserve assessment methodology: Overview, analysis and future directions," Mar. 01, 2020, *Elsevier Ltd*. doi: 10.1016/j.rser.2019.109515.
- [16] M. J. O'Sullivan and J. P. O'Sullivan, "7 - Reservoir modeling and simulation for geothermal resource characterization and evaluation," in *Geothermal Power Generation*, R. DiPippo, Ed., Woodhead Publishing, 2016, pp. 165–199. doi: <https://doi.org/10.1016/B978-0-08-100337-4.00007-3>.
- [17] K. Wisian, D. Blackwell, and M. Richards, "Correlation of heat loss and total energy production for geothermal systems," *Geothermal Resources Council Transactions*, vol. 25, Aug. 2001.
- [18] P. Muffler and R. Cataldi, "Methods for regional assessment of geothermal resources," *Geothermics*, vol. 7, no. 2, 1978, doi: [https://doi.org/10.1016/0375-6505\(78\)90002-0](https://doi.org/10.1016/0375-6505(78)90002-0).
- [19] M. A. Grant, "Geothermal resource proving criteria," pp. 2581–2584.
- [20] S. Sanyal, R. Henneberger, C. Klein, and R. Decker, "A Methodology for Assessment of Geothermal Energy Reserves Associated with Volcanic Systems," *Transactions - Geothermal Resources Council*, Jan. 2002.
- [21] C. A. Brook, R. H. Mariner, D. R. Mabey, J. R. Swanson, M. Guffanti, and L. J. P. Muffler, "Hydrothermal convection systems with reservoir temperatures > 90 C," *Assessment of geothermal resources of the United States-1978: US Geological Survey Circular*, vol. 790, pp. 18–85.
- [22] R. James, "Power potential of geothermal fields," University of Auckland, Geothermal Institute, p. 97.
- [23] M. Wilmarth and J. Stimac, "Power density in geothermal fields," *Power*, vol. 19, no. 2015, p. 25.
- [24] D. E. White and D. L. Williams, *Assessment of geothermal resources of the United States, 1975*. US Department of the Interior, Geological Survey.
- [25] J. J. Quinao and S. J. Zarrouk, "A Review of the Volumetric Stored-Heat Resource Assessment: One Method , Different Result," pp. 24–26.
- [26] AGRCC, "Australian Code for Reporting of Exploration Results, Geothermal Resources and Geothermal Reserves," 2010.
- [27] AGRCC, "Geothermal Lexicon For Resources and Reserves Definition and Reporting".
- [28] S. J. Zarrouk and F. Simiyu, "A review of geothermal resource estimation methodology,"
- [29] M. Parini and K. Riedel, "COMBINING PROBABILISTIC VOLUMETRIC AND NUMERICAL SIMULATION APPROACHES TO IMPROVE ESTIMATES OF GEOTHERMAL RESOURCE CAPACITY."
- [30] R. C. M. Malate and M. J. O'sullivan, "Modelling of chemical and thermal changes in well pn-26 palinpinon geothermal field, Philippines," *Geothermics*, vol. 20, no. 5, pp. 291–318, 1991, doi: [https://doi.org/10.1016/0375-6505\(91\)90022-N](https://doi.org/10.1016/0375-6505(91)90022-N).

- [31] G. Gaoxuan, G. Axelsson, Y. Chao, and X. Baodong, "Assessment of the hofsstadir geothermal field, W-Iceland, by lumped parameter modelling, Monte Carlo simulation and tracer test analysis," *Geothermal training programme. Reports. Orkustofnun, Grensásvegur*, vol. 9, no. 18, pp. 247–279.
- [32] R. G. Orizonte, A. E. Amistoso, and R. C. M. Malate, "Optimizing the Palinpinon-2 geothermal reservoir, Southern Negros geothermal production field, Philippines," *Proc. World geotherm. Congr. Antalya, Turkey*.
- [33] M. J. O'Sullivan, K. Pruess, and M. J. Lippmann, "State of the art of geothermal reservoir simulation," *Geothermics*, vol. 30, no. 4, pp. 395–429, 2001, doi: [https://doi.org/10.1016/S0375-6505\(01\)00005-0](https://doi.org/10.1016/S0375-6505(01)00005-0).
- [34] J. W. Pritchett, "STAR: A geothermal reservoir simulation system," pp. 2959–2963.
- [35] K. Pruess, "TOUGH2-A general-purpose numerical simulator for multiphase fluid and heat flow".
- [36] P. K. W. Vinsome and G. M. Shook, "Multi-purpose simulation," *J Pet Sci Eng*, vol. 9, no. 1, pp. 29–38, 1993, doi: [https://doi.org/10.1016/0920-4105\(93\)90026-B](https://doi.org/10.1016/0920-4105(93)90026-B).
- [37] K. Zhang, B.-H. Lee, L. Ling, T.-R. Guo, C.-H. Liu, and S. Ouyang, "Modeling studies for production potential of Chingshui geothermal reservoir," *Renew Energy*, vol. 94, pp. 568–578, 2016, doi: <https://doi.org/10.1016/j.renene.2016.03.099>.
- [38] B. M. Adams and P. Schädle, "genGEO: coupled reservoir-electricity-cost geothermal simulator," Dec. 2020, *Zenodo*. doi: 10.5281/zenodo.4383139.
- [39] K. Beckers, K. McCabe, and U. O. of E. E. and R. E. (EERE), "GEOPHIRES v2.0," 2020, *United States*. doi: 10.11578/dc.20200217.1.
- [40] E. Cassano, R. Fichera, and F. Rota, "Rilievo aeromagnetico d'Italia. Alcuni risultati interpretativi," *Rilievo Aeromagnetico d'Italia. Alcuni Risultati Interpretativi*, vol. 2, pp. 939–962, Jan. 1986.
- [41] V. Pasquale, G. Gola, P. Chiozzi, and M. Verdoya, "Thermophysical properties of the Po Basin rocks," *Geophys J Int*, vol. 186, no. 1, pp. 69–81, Jul. 2011, doi: 10.1111/j.1365-246X.2011.05040.x.
- [42] K. Sekiguchi, "A method for determining terrestrial heat flow in oil basinal areas," *Tectonophysics*, vol. 103, no. 1–4, pp. 67–79, Mar. 1984, doi: 10.1016/0040-1951(84)90075-1.
- [43] A. M. Hofmeister, "Thermodynamic and optical thickness corrections to diffusive radiative transfer formulations with application to planetary interiors," *Geophys Res Lett*, vol. 41, no. 9, pp. 3074–3080, May 2014, doi: 10.1002/2014GL059833.
- [44] A. H. Lachenbruch, "Crustal temperature and heat production: Implications of the linear heat-flow relation," *J Geophys Res*, vol. 75, no. 17, pp. 3291–3300, Jun. 1970, doi: 10.1029/JB075i017p03291.
- [45] V. Pasquale, M. Verdoya, and P. Chiozzi, "Radioactive heat generation and its thermal effects in the Alps–Apennines boundary zone," *Tectonophysics*, vol. 331, no. 3, pp. 269–283, Feb. 2001, doi: 10.1016/S0040-1951(00)00294-8.
- [46] H. J. Ramey, "Wellbore Heat Transmission," *Journal of Petroleum Technology*, vol. 14, no. 04, pp. 427–435, Apr. 1962, doi: 10.2118/96-PA.
- [47] L. P. Dake, *Fundamentals of Reservoir Engineering*, 1st ed., vol. 8. 1983.
- [48] Magnus Holmgren, "X Steam, Thermodynamic properties of water and steam," 2024, *MATLAB Central File Exchange*. Accessed: Sep. 04, 2024. [Online]. Available: <https://it.mathworks.com/matlabcentral/fileexchange/9817-x-steam-thermodynamic-properties-of-water-and-steam>
- [49] R. J. Volino, "A MATLAB Based Set of Functions for Finding Thermodynamic Properties and Solving Gas Turbine and Other Thermodynamics Problems," in *Volume 5: Education; Electric Power; Fans and Blowers*, American Society of Mechanical Engineers, Jun. 2022. doi: 10.1115/GT2022-82041.
- [50] K. Rafferty, "GEOHERMAL POWER GENERATION A PRIMER ON LOW-TEMPERATURE, SMALL-SCALE APPLICATIONS," 2000.
- [51] R. DiPippo, "GEOHERMAL ENERGY AS A SOURCE OF ELECTRICITY A Worldwide Survey of the Design and Operation of Geothermal Power Plants," 1980.
- [52] R. Bonciani, A. Lenzi, F. Luperini, and F. Sabatelli, "EGC 2013 Geothermal power plants in Italy: increasing the environmental compliance," 2013.
- [53] P. Ediatmaja, P. S. Darmanto, and D. T. Maulana, "Performance Evaluation of Geothermal Power Production Using EES (Case Study Ulumbu Geothermal Power Production Unit 4 East Nusa Tenggara, Indonesia)," in *IOP Conference Series: Earth and Environmental Science*, IOP Publishing Ltd, Apr. 2021. doi: 10.1088/1755-1315/732/1/012017.
- [54] C. Mulyana, R. Adiprana, A. H. Saad, M. H. Ridwan, and F. Muhammad, "The thermodynamic cycle models for geothermal power plants by considering the working fluid characteristic," in *AIP Conference Proceedings*, American Institute of Physics Inc., Feb. 2016. doi: 10.1063/1.4941863.

- [55] R. DiPippo, *Geothermal power plants: principles, applications, case studies and environmental impact*. Butterworth-Heinemann.
- [56] M. Andrés and T. Millachine, "Guidelines for Optimum Gas Extraction System Selection," 2011.
- [57] M. Yari, "Exergetic analysis of various types of geothermal power plants," *Renew Energy*, vol. 35, no. 1, pp. 112–121, Jan. 2010, doi: 10.1016/j.renene.2009.07.023.
- [58] P. J. Yekoladio, T. Bello-Ochende, and J. P. Meyer, "Thermodynamic analysis and performance optimization of organic rankine cycles for the conversion of low-to-moderate grade geothermal heat," *Int J Energy Res*, vol. 39, no. 9, pp. 1256–1271, Jul. 2015, doi: 10.1002/er.3326.
- [59] C. Augustine, R. Field, R. DiPippo, G. Gigliucci, I. Fastelli, and J. Tester, "Modeling and analysis of sub-and supercritical binary Rankine cycles for low-to mid-temperature geothermal resources," *GRC Transactions*, vol. 33, pp. 689–693.
- [60] M. Kanoglu and A. Bolatturk, "Performance and parametric investigation of a binary geothermal power plant by exergy," *Renew Energy*, vol. 33, no. 11, pp. 2366–2374, Nov. 2008, doi: 10.1016/j.renene.2008.01.017.
- [61] T. Sharmin, N. R. Khan, M. S. Akram, and M. M. Ehsan, "A State-of-the-Art Review on Geothermal Energy Extraction, Utilization, and Improvement Strategies: Conventional, Hybridized, and Enhanced Geothermal Systems," May 01, 2023, *Elsevier B.V.* doi: 10.1016/j.ijft.2023.100323.
- [62] D. Leith and D. Mehta, "CYCLONE PERFORMANCE AND DESIGN," 1973.
- [63] S. J. Zarrouk and H. Moon, "Efficiency of geothermal power plants: A worldwide review," Jul. 2014. doi: 10.1016/j.geothermics.2013.11.001.
- [64] F. Sabatelli, M. Mannari, and R. Parri, "Hydrogen sulfide and mercury abatement: Development and successful operation of AMIS technology," *Transactions - Geothermal Resources Council*, vol. 33, pp. 308–312, Sep. 2009.
- [65] C. Osnaghi, *Teoria delle Turbomacchine*, 3rd ed. 2020.
- [66] "The Engineering ToolBox." Accessed: Sep. 04, 2024. [Online]. Available: https://www.engineeringtoolbox.com/pumps-power-d_505.html
- [67] "Introducing Aspen Plus," in *Aspen Plus®*, John Wiley & Sons, Ltd, 2016, ch. 1, pp. 1–47. doi: <https://doi.org/10.1002/9781119293644.ch1>.
- [68] H. Kazemi and M. A. Ehyaei, "Energy, exergy, and economic analysis of a geothermal power plant," *Advances in Geo-Energy Research*, vol. 2, no. 2, pp. 190–209, Jun. 2018, doi: 10.26804/ager.2018.02.07.
- [69] ASSOCLIMA, "TORRI DI RAFFREDDAMENTO," 2017.
- [70] Y. A. Çengel, *Termodinamica e trasmissione del calore*. Milano: McGraw-Hill libri Italia, 1998.
- [71] M. S. Peters, K. D. Timmerhaus, and R. E. West, *Plant design and economics for chemical engineers*, 5. ed. Boston: McGraw-Hill, 2003.
- [72] C. Ezgi, "Basic Design Methods of Heat Exchanger," 2017. doi: 10.5772/67888.
- [73] V. Verda and E. Guelpa, *METODI TERMODINAMICI PER L'USO EFFICIENTE DELLE RISORSE ENERGETICHE*, 2nd ed. 2015.
- [74] J. Wang, J. Wang, Y. Dai, and P. Zhao, "Thermodynamic analysis and optimization of a flash-binary geothermal power generation system," *Geothermics*, vol. 55, pp. 69–77, May 2015, doi: 10.1016/j.geothermics.2015.01.012.
- [75] L. d'Anterrosches and M. Bruno, "Céondo." Accessed: Sep. 13, 2024. [Online]. Available: <https://www.chemeo.com/cid/68-840-4/Isobutane>
- [76] S. Park, A. Langat, K. Lee, and Y. Yoon, "Measuring the impact of risk on LCOE (levelized cost of energy) in geothermal technology," *Geothermal Energy*, vol. 9, no. 1, p. 27, Dec. 2021, doi: 10.1186/s40517-021-00209-9.
- [77] A. Gallo, "A Refresher on Net Present Value," 2014. [Online]. Available: www.business-literacy.com.
- [78] IRENA, "Renewable Power Generation Costs in 2020," Abu Dabhi, 2021. Accessed: Sep. 25, 2024. [Online]. Available: [/-/media/Files/IRENA/Agency/Publication/2021/Jun/IRENA_Power_Generation_Costs_2020.pdf](https://media/Files/IRENA/Agency/Publication/2021/Jun/IRENA_Power_Generation_Costs_2020.pdf)
- [79] National Renewable Energy Laboratory (NREL), "Annual Technology Baseline." Accessed: Oct. 02, 2024. [Online]. Available: <https://atb.nrel.gov/electricity/2024/index>
- [80] EMBER, "European wholesale electricity price data," 2024. Accessed: Oct. 02, 2024. [Online]. Available: <https://ember-climate.org/data-catalogue/european-wholesale-electricity-price-data/>
- [81] "Google Finance." Accessed: Sep. 27, 2024. [Online]. Available: <https://www.google.com/finance/quote/EUR-USD?hl=it>

- [82] U.S. Bureau of Labor Statistics, "U.S. Bureau of Labor Statistics ." Accessed: Sep. 25, 2024. [Online]. Available: <https://data.bls.gov/series-report?redirect=true>
- [83] C. Maxwell, "Cost Indices - Towering Skill." Accessed: Sep. 26, 2024. [Online]. Available: <https://toweringskills.com/financial-analysis/cost-indices/>
- [84] G. L. Mines, "GETEM User Manual," 2016.
- [85] D. R. Woods, *Rules of Thumb in Engineering Practice*. Wiley, 2007. doi: 10.1002/9783527611119.
- [86] "D2.3 GEOELEC web-service database on resource assessment Content."
- [87] J. F. Dewey, M. L. Helman, S. D. Knott, E. Turco, and D. H. W. Hutton, "Kinematics of the western Mediterranean," *Geological Society, London, Special Publications*, vol. 45, no. 1, pp. 265–283, Jan. 1989, doi: 10.1144/GSL.SP.1989.045.01.15.
- [88] A. Malinverno and W. B. F. Ryan, "Extension in the Tyrrhenian Sea and shortening in the Apennines as result of arc migration driven by sinking of the lithosphere," *Tectonics*, vol. 5, no. 2, pp. 227–245, Apr. 1986, doi: 10.1029/TC005i002p00227.
- [89] C. Doglioni, E. Gueguen, P. Harabaglia, and F. Mongelli, "On the origin of west-directed subduction zones and applications to the western Mediterranean," *Geological Society, London, Special Publications*, vol. 156, no. 1, pp. 541–561, Jan. 1999, doi: 10.1144/GSL.SP.1999.156.01.24.
- [90] C. Faccenna, C. Piromallo, A. Crespo-Blanc, L. Jolivet, and F. Rossetti, "Lateral slab deformation and the origin of the western Mediterranean arcs," *Tectonics*, vol. 23, no. 1, Feb. 2004, doi: 10.1029/2002TC001488.
- [91] G. Rosenbaum and G. S. Lister, "Neogene and Quaternary rollback evolution of the Tyrrhenian Sea, the Apennines, and the Sicilian Maghrebides," *Tectonics*, vol. 23, no. 1, Feb. 2004, doi: 10.1029/2003TC001518.
- [92] P. Scandone, "Origin of the Tyrrhenian Sea and Calabrian Arc," *Bollettino Della Societa Geologica Italiana*, vol. 98, no. 1, pp. 27–34, 1990.
- [93] L. Royden, E. Patacca, and P. Scandone, "Segmentation and configuration of subducted lithosphere in Italy: An important control on thrust-belt and foredeep-basin evolution," *Geology*, vol. 15, no. 8, p. 714, 1987, doi: 10.1130/0091-7613(1987)15<714:SACOSL>2.0.CO;2.
- [94] E. Patacca, S. R., and P. Scandone, "Tyrrhenian Basin and Apenninic Arcs: Kinematic relations since late Tortonian times," *Bollettino- Societa Geologica Italiana*, vol. 45, pp. 425–451, Jan. 1990.
- [95] C. Doglioni, "A proposal for the kinematic modelling of W-dipping subductions - possible applications to the Tyrrhenian-Apennines system," *Terra Nova*, vol. 3, no. 4, pp. 423–434, Jul. 1991, doi: 10.1111/j.1365-3121.1991.tb00172.x.
- [96] L. Jolivet and C. Faccenna, "Mediterranean extension and the Africa-Eurasia collision," *Tectonics*, vol. 19, no. 6, pp. 1095–1106, Dec. 2000, doi: 10.1029/2000TC900018.
- [97] G. Lavecchia, "The tyrrhenian-apennines system: structural setting and seismotectogenesis," *Tectonophysics*, vol. 147, no. 3–4, pp. 263–296, Apr. 1988, doi: 10.1016/0040-1951(88)90190-4.
- [98] L. E. and R. Nicolich, "Geodinamica del Tirreno e dell'Appennino Centro-Meridionale: la nuova carta della Moho," *Mem. Soc. Geol. It.*, vol. 41, pp. 121–140, Jan. 1988.
- [99] G. Lavecchia, C. Federico, F. Stoppa, and G. D. Karner, "LA DISTENSIONE TOSCO-TIRRENICA COME POSSIBILE MOTORE DELLA COMPRESSIONE APPENNINICA," *Studi geologici camerati*, pp. 489–497, 1995.
- [100] Liotta, Cernobori, and Nicolich, "Restricted rifting and its coexistence with compressional structures: results from the CROP 3 traverse (Northern Apennines, Italy)," *Terra Nova*, vol. 10, no. 1, pp. 16–20, Jan. 1998, doi: 10.1046/j.1365-3121.1998.00157.x.
- [101] G. Lavecchia, P. Boncio, N. Creati, and F. Brozzetti, "Some Aspects Of The Italian Geology Not Fitting With A Subduction Scenario," *Journal of the Virtual Explorer*, vol. 10, pp. 1–14, 2003, [Online]. Available: www.unich.it
- [102] R. Catalano, C. Doglioni, and S. Merlini, "On the Mesozoic Ionian Basin," *Geophys J Int*, vol. 144, no. 1, pp. 49–64, Jan. 2001, doi: 10.1046/j.0956-540X.2000.01287.x.
- [103] G. Ciarapica and L. Passeri, "The palaeogeographic duplicity of the Apennines," *Boll. Soc. Geol. It.*, vol. 1, pp. 67–75, Jan. 2002.
- [104] P. Casero *et al.*, "Neogene Geodynamic Evolution of the Southern Apennines," *Mem. Soc. Geol. It.*, vol. 41, pp. 109–120, Jan. 1992.
- [105] R. Fantoni and R. Franciosi, "Tectono-sedimentary setting of the Po Plain and Adriatic foreland," *RENDICONTI LINCEI*, vol. 21, no. S1, pp. 197–209, Dec. 2010, doi: 10.1007/s12210-010-0102-4.

- [106] E. Patacca and P. Scandone, "THE PLIO-PLEISTOCENE THRUST BELT – FOREDEEP SYSTEM IN THE SOUTHERN APENNINES AND SICILY (ITALY)," *Italian Geological Society for the IGC*, vol. 32, pp. 93–129, 2004.
- [107] D. Scrocca, E. Carminati, C. Doglioni, and D. Procaccianti, "Tyrrhenian Sea," in *Regional Geology and Tectonics: Phanerozoic Passive Margins, Cratonic Basins and Global Tectonic Maps*, Elsevier, 2012, pp. 472–485. doi: 10.1016/B978-0-444-56357-6.00012-3.
- [108] L. Chiaraluce, C. Chiarabba, C. Collettini, D. Piccinini, and M. Cocco, "Architecture and mechanics of an active low-angle normal fault: Alto Tiberina Fault, northern Apennines, Italy," *J Geophys Res Solid Earth*, vol. 112, no. B10, Oct. 2007, doi: 10.1029/2007JB005015.
- [109] F. Brozzetti, "The Campania-Lucania Extensional Fault System, southern Italy: A suggestion for a uniform model of active extension in the Italian Apennines," *Tectonics*, vol. 30, no. 5, Oct. 2011, doi: 10.1029/2010TC002794.
- [110] G. Lavecchia *et al.*, "Multidisciplinary inferences on a newly recognized active east-dipping extensional system in Central Italy," *Terra Nova*, vol. 29, no. 1, pp. 77–89, Feb. 2017, doi: 10.1111/ter.12251.
- [111] L. Carmignani, P. Conti, G. Cornamusini, and M. Meccheri, "The internal Northern Apennines, the Northern Tyrrhenian Sea and the Sardinia-Corsica Block," in *Geology of Italy*, 2004, pp. 59–77.
- [112] G. Calcagnile and G. F. Panza, "The main characteristics of the lithosphere-asthenosphere system in Italy and surrounding regions," *Pure Appl Geophys*, vol. 119, no. 4, pp. 865–879, Jul. 1980, doi: 10.1007/BF01131263.
- [113] H. Anderson and J. Jackson, "The deep seismicity of the Tyrrhenian Sea," *Geophys J Int*, vol. 91, no. 3, pp. 613–637, Dec. 1987, doi: 10.1111/j.1365-246X.1987.tb01661.x.
- [114] G. F. Panza, R. B. Raykova, E. Carminati, and C. Doglioni, "Upper mantle flow in the western Mediterranean," *Earth Planet Sci Lett*, vol. 257, no. 1–2, pp. 200–214, May 2007, doi: 10.1016/j.epsl.2007.02.032.
- [115] G. F. Panza and R. B. Raykova, "Structure and rheology of lithosphere in Italy and surrounding," *Terra Nova*, vol. 20, no. 3, pp. 194–199, Jun. 2008, doi: 10.1111/j.1365-3121.2008.00805.x.
- [116] E. Trumpy and A. Manzella, "Geothopica and the interactive analysis and visualization of the updated Italian National Geothermal Database," *International Journal of Applied Earth Observation and Geoinformation*, vol. 54, pp. 28–37, Feb. 2017, doi: 10.1016/j.jag.2016.09.004.
- [117] F. Mongelli, G. Zito, B. Della Vedova, G. Pellis, P. Squarci, and L. Taffi, "Geothermal Regime of Italy and Surrounding Seas," 1991, pp. 381–394. doi: 10.1007/978-3-642-75582-8_18.
- [118] R. Cataldi, F. Mongelli, P. Squarci, L. Taffi, G. Zito, and C. Calore, "Geothermal ranking of Italian territory," *Geothermics*, vol. 24, no. 1, pp. 115–129.
- [119] B. Della Vedova, S. Bellani, G. Pellis, and P. Squarci, "Deep temperatures and surface heat flow distribution," in *Anatomy of an Orogen: the Apennines and Adjacent Mediterranean Basins*, Dordrecht: Springer Netherlands, 2001, pp. 65–76. doi: 10.1007/978-94-015-9829-3_7.
- [120] S. Bellani, A. Brogi, A. Lazzarotto, D. Liotta, and G. Ranalli, "Heat flow, deep temperatures and extensional structures in the Larderello Geothermal Field (Italy): constraints on geothermal fluid flow," *Journal of Volcanology and Geothermal Research*, vol. 132, no. 1, pp. 15–29, Apr. 2004, doi: 10.1016/S0377-0273(03)00418-9.
- [121] ENEL, ENI-AGIP, CNR, and ENEA, "Inventario delle risorse geotermiche nazionali," Sep. 1988.
- [122] ENEL, ENI-AGIP, CNR, and ENEA, "Inventario delle risorse geotermiche nazionali," Jun. 1993.
- [123] S. J. Zarrouk and H. Moon, "Efficiency of geothermal power plants: A worldwide review," Jul. 2014. doi: 10.1016/j.geothermics.2013.11.001.

Appendix

In this section the most important codes are reported. The title of each subchapter corresponds to the name given to the code.

I. *Askinput_binary*

```
%% Ambient condition
choiceAmbient=input('Want to insert set ambient condition? 1=yes, 0=no: ');
if choiceAmbient==1
    T0=input('set ambient temperature [°C]: ');
    p0=input('set ambient pressure [bar]: ');
else
    T0=20;
    p0=1;
    fprintf('The ambient temperature is %d °C and the ambient pressure is %d bar\n', T0, p0);
end

choicelifetimeValue=input('Want to set the lifetime of the plant? 1=yes, 0=no: ');
if choicelifetimeValue==1
    lifetime=input('set the lifetime of the plant: ');
    fprintf('The lifetime is %d years\n', lifetime);
else
    lifetime=30;
    fprintf('The lifetime is %d years\n', lifetime);
end

choicetimestepperyearValue=input('Want to set the time step per year? 1=yes, 0=no: ');
if choicetimestepperyearValue==1
    time_step_per_year=input('set the time step per year: ');
    fprintf('The number of time step per year is %d \n', time_step_per_year);
else
    time_step_per_year=1;
    fprintf('The number of time step per year is %d \n', time_step_per_year);
end

%% Production and Injection parameters
choiceDprodValue=input('Want to set the diameter of production well? 1=yes, 0=no: ');
if choiceDprodValue==1
    d_prod=input('The diameter must be between 0.0254 and 0.7620. Set production well diameter: ');
    while d_prod/0.0254<1 || d_prod/0.0254>30
        d_prod=input('The diameter must be between 0.0254 and 0.7620. Set production well diameter: ');
    end
    fprintf('The production well diameter is %f [m]\n', d_prod);
else
    d_prod=8*0.0254;
    fprintf('The production well diameter is %f [m]\n', d_prod);
end

choiceNprodValue=input('Want to set the number of production well? 1=yes, 0=no: ');
if choiceNprodValue==1
    n_prod=input('Set number of production wells: ');
    fprintf('The number of production wells is %d [-]\n', n_prod);
else
    n_prod=2;
    fprintf('The number of production wells is %d [-]\n', n_prod);
end

L_prod=0;
while L_prod==0
    L_prod=input('set the depth of the production well: ');
    fprintf('The production well depth is %.2f\n', L_prod);
    % Check if the input is valid (non-empty and a number)
    if isempty(L_prod) || ~isnumeric(L_prod)
        disp('Invalid input. Please enter a number. ');
        L_prod = 0; % Reset the value to stay in the loop
    else
        fprintf('The production well depth is %.2f\n', L_prod);
    end
end

%% Injection condition
choiceNinjValue=input('Want to set the number of injection well? 1=yes, 0=no: ');
```

```

if choiceNinjValue==1
    n_inj=input('Set number of injection wells: ');
    fprintf('The number of injection wells is %d [-]\n', n_inj);
else
    n_inj=1;
    fprintf('The number of injection wells is %d [-]\n', n_inj);
end

choiceDinjValue=input('Want to set the diameter of injection well? 1=yes, 0=no: ');
if choiceDinjValue==1
    d_inj=input('The diameter must be between 0.0254 and 0.7620. Set injection well diameter: ');
    while d_inj/0.0254<1 || d_inj/0.0254>30
        d_inj=input('The diameter must be between 0.0254 and 0.7620. Set injection well diameter: ');
    end
    fprintf('The injection well diameter is %f [m]\n', d_inj);
else
    d_inj=8*0.0254;
    fprintf('The injection well diameter is %f [m]\n', d_inj);
end

% Filter's diameter
choiceDfilter=input('Want to set the diameter of the filter? 1=yes, 0=no: ');
if choiceDfilter==1
    D_filter=input('The diameter is worth to be between 0.1 and 0.5. Set filter diameter: ');
    fprintf('The diameter of the filter is %.2f [m]\n',D_filter);
else
    D_filter=0.4;
    fprintf('The diameter of the filter is %.2f [m]\n',D_filter);
end

% Efficiency of machinaries
choiceEtaTurbValue=input('Want to set turbine efficiency? 1=yes, 0=no: ');
if choiceEtaTurbValue==1
    eta_turb=input('set turbine efficiency: ');
    while eta_turb<=0 || eta_turb>=1
        eta_turb=input('The efficiency must be between 0 and 1. Set turbine efficiency: ');
    end
    fprintf('The turbine efficiency is %f [-]\n', eta_turb);
else
    eta_turb=0.85;
    fprintf('The turbine efficiency is %f [-]\n', eta_turb);
end

choiceXvaplimValue=input('Want to set the admissible fraction of vapour expanding in turbine? 1=yes, 0=no: ');
if choiceXvaplimValue==1
    xvap_lim=input('set the admissible fraction of vapour: ');
    if xvap_lim>1
        xvap_lim=1;
        fprintf('The admissible fraction of vapour is %f [-]\n', xvap_lim);
    end
    if xvap_lim<0.8
        xvap_lim=0.85;
        fprintf('The admissible fraction of vapour is %f [-]\n', xvap_lim);
    end
else
    xvap_lim=0.95;
    fprintf('The admissible fraction of vapour is %f [-]\n', xvap_lim);
end

choiceEtaCondPumpValue=input('Want to set the efficiency of condensation pump? 1=yes, 0=no: ');
if choiceEtaCondPumpValue==1
    eta_condpump=input('set condensation pump efficiency: ');
    while eta_condpump<=0 || eta_condpump>=1
        eta_condpump=input('The efficiency must be between 0 and 1. Set condensation pump efficiency: ');
    end
    fprintf('The condensation pump efficiency is %f [-]\n', eta_condpump);
else
    eta_condpump=0.85;
    fprintf('The condensation pump efficiency is %f [-]\n', eta_condpump);
end

choiceEtaBDPumpValue=input('Want to set the efficiency of blowdown pump? 1=yes, 0=no: ');
if choiceEtaBDPumpValue==1

```

```

    eta_bdpump=input('set blowdown pump efficiency: ');
    while eta_bdpump<=0 || eta_bdpump>=1
        eta_bdpump=input('The efficiency must be between 0 and 1. Set blowdown pump efficiency: ');
    end
    fprintf('The blowdown pump efficiency is %f [-]\n', eta_bdpump);
else
    eta_bdpump=0.85;
    fprintf('The blowdown pump efficiency is %f [-]\n', eta_bdpump);
end

choiceEtaCircPumpValue=input('Want to set the efficiency of circulation pump? 1=yes, 0=no: ');
if choiceEtaCircPumpValue==1
    eta_circpump=input('set circulation pump efficiency: ');
    while eta_circpump<=0 || eta_circpump>=1
        eta_circpump=input('The efficiency must be between 0 and 1. Set circulation pump efficiency: ');
    end
    fprintf('The circulation pump efficiency is %f [-]\n', eta_circpump);
else
    eta_circpump=0.85;
    fprintf('The circulation pump efficiency is %f [-]\n', eta_circpump);
end

choiceEtaGenerator=input('Want to set the efficiency of generator? 1=yes, 0=no: ');
if choiceEtaGenerator==1
    eta_gen=input('set circulation pump efficiency: ');
    while eta_gen<=0 || eta_gen>=1
        eta_gen=input('The efficiency must be between 0 and 1. Set circulation pump efficiency: ');
    end
    fprintf('The circulation pump efficiency is %f [-]\n', eta_gen);
else
    eta_gen=0.95;
    fprintf('The circulation pump efficiency is %f [-]\n', eta_gen);
end

%% Global heat transfer coefficient HX
choiceGlobalHeatTransferCoefficientHX=input('Want to set the global heat transfer coefficient water-isobutane
(U)? 1=yes, 0=no: ');
if choiceGlobalHeatTransferCoefficientHX==1
    U_hx=input('set the global heat transfer coefficient water-isobutane (U), between 500 and 1000: ');
    fprintf('The global heat transfer coefficient water-isobutane (U) is %.2f [W/m2K]\n', U_hx);
else
    U_hx=500;
    fprintf('The global heat transfer coefficient water-isobutane (U) is %.2f [W/m2K]\n', U_hx);
end

%% Global heat transfer coefficient condenser
choiceGlobalHeatTransferCoefficientCond=input('Want to set the global heat transfer coefficient water-
isobutane (U)? 1=yes, 0=no: ');
if choiceGlobalHeatTransferCoefficientCond==1
    U_cond=input('set the global heat transfer coefficient water-isobutane (U), between 375-750: ');
    fprintf('The global heat transfer coefficient water-isobutane (U) is %.2f [W/m2K]\n', U_cond);
else
    U_cond=500;
    fprintf('The global heat transfer coefficient water-isobutane (U) is %.2f [W/m2K]\n', U_cond);
end

%% cooling tower heat/electricity ratio
choiceHeatToElectricityRatio=input('Want to set the heat to electricity ratio for cooling tower? 1=yes, 0=no:
');
if choiceHeatToElectricityRatio==1
    kwt_to_kWe=input('This value can be equal to 40 or 80. Set heat to electricity ratio: ');
    fprintf('The heat to electricity ratio is %.1f [-]\n',kwt_to_kWe);
else
    kwt_to_kWe=40;
    fprintf('The heat to electricity ratio is %.1f [-]\n',kwt_to_kWe);
end

%% ORC fluid pressure
choiceORCpressure=input('Want to set the pressure of organic fluid after the condensation pump? 1=yes, 0=no:
');
if choiceORCpressure==1
    pressure_orc_fluid=input('set the pressure of organic fluid after condensation pump: ');
    fprintf('The pressure of organic fluid is %.2f [bar]\n', pressure_orc_fluid);
else
    pressure_orc_fluid=25;

```

```

    fprintf('The pressure of organic fluid is %.2f [bar]\n', pressure_orc_fluid);
end

%% ORC fluid temperature
choiceORCtemperature=input('Want to set the ORC fluid initial temperature? 1=yes,0=no: ');
if choiceORCtemperature==1
    temperature_orc_fluid=input('set the temperature of the ORC fluid: ');
    fprintf('The temperature of the ORC fluid is %.2f [°C]\n', temperature_orc_fluid);
else
    temperature_orc_fluid=15;
    fprintf('The temperature of the ORC fluid is %.2f [°C]\n', temperature_orc_fluid);
end

%% pressure turbine outlet
choicePressureTurbineOutlet=input('Want to set the pressure at turbine outlet? 1=yes, 0=no: ');
if choicePressureTurbineOutlet==1
    pressure_out_turbine=input('set the pressure at turbine outlet: ');
    fprintf('The pressure at turbine outlet is %.2f [bar]\n', pressure_out_turbine);
else
    pressure_out_turbine=4;
    fprintf('The pressure at turbine outlet is %.2f [bar]\n', pressure_out_turbine);
end

%% pressure in cooling tower loop
choicePressureCoolingTower=input('Want to set the pressure in cooling tower loop? 1=yes, 0=no: ');
if choicePressureCoolingTower==1
    pressure_water_inlet_cond=input('set the pressure in cooling tower loop: ');
    fprintf('The pressure in cooling tower loop is %.2f [bar]\n', pressure_water_inlet_cond);
else
    pressure_water_inlet_cond=2;
    fprintf('The pressure in cooling tower loop is %.2f [bar]\n', pressure_water_inlet_cond);
end

%% Temperature difference between cooling tower and ORC fluid
choiceTemperatureDifferenceInletCondenser=input('Want to set the temperature difference between ORC fluid
lower temperature and water temperature at inlet condenser? 1=yes,0=no: ');
if choiceTemperatureDifferenceInletCondenser==1
    dT_cooling_water=input('set the temperature difference: ');
    fprintf('The temperature of the ORC fluid is %.2f [°C]\n', dT_cooling_water);
else
    dT_cooling_water=5;
    fprintf('The temperature difference is %.2f [°C]\n', dT_cooling_water);
end

%% economic parameter
choiceGeologicalFormation=input('Want to set the geological formation? 1=yes, 0=no: ');
if choiceGeologicalFormation==1
    sourcetype=input('Set the type of geological formation. 1=sedimentary, 2=fractured, 3=basement: ');
    if sourcetype==1
        Succrate=0.95;
        fprintf('The geological formation is sedimentary, and the success rate in drilling is %.2f [-
]\n',Succrate);
    elseif sourcetype==2
        Succrate=0.75;
        fprintf('The geological formation is fractured, and the success rate in drilling is %.2f [-
]\n',Succrate);
    else
        Succrate=0.9;
        fprintf('The geological formation is basement, and the success rate in drilling is %.2f [-
]\n',Succrate);
    end
else
    Succrate=0.9;
    fprintf('The geological formation is basement, and the success rate in drilling is %.2f [-]\n',Succrate);
end

choiceFieldType=input('Want to choose the field type? 1=yes, 0=no: ');
if choiceFieldType==1
    fieldtype=input('Set 1 for brownfield and 2 for greenfield: ');
    if fieldtype==1
        fprintf('The fieldtype chosen is brownfield [-]\n');
    else
        fprintf('The fieldtype chosen is greenfield [-]\n');
    end
else
end

```

```

        fieldtype=2;
        fprintf('The fieldtype chosen is greenfield [-]\n');
end

choiceEvaporator=input('Want to chose the type of evaporator? 1=yes, 0=no: ');
if choiceEvaporator==1
    typeHX=input('Choose the type of evaporator. 1=Shell and tube, 2=Plate: ');
    if typeHX==1
        fprintf('The evaporator is shell and tube type\n');
    else if typeHX==2
        fprintf('The evaporator is plate type\n');
    else
        fprintf('The evaporator is shell and tube type\n');
    end
end
else
    typeHX=1;
    fprintf('The evaporator is shell and tube type\n');
end

choiceCondenser=input('Want to choose the type of condenser? 1=yes, 0=no: ');
if choiceCondenser==1
    typeCOND=input('Choose the type of condenser. 1=Shell and tube, 2=Plate: ');
    if typeCOND==1
        fprintf('The condenser is shell and tube type\n');
    else if typeCOND==2
        fprintf('The condenser is plate type\n');
    else
        fprintf('The condenser is shell and tube type\n');
    end
end
else
    typeCOND=1;
    fprintf('The condenser is shell and tube type\n');
end

choiceCapacityFactor=input('Want to set the capacity factor? 1=yes, 0=no: ');
if choiceCapacityFactor==1
    CF=input('set the capacity factor (suggested between 0.7 and 0.9): ');
    fprintf('the capacity factor is %.2f [-]\n', CF);
else
    CF=0.82;
    fprintf('the capacity factor is %.2f [-]\n', CF);
end

choiceElectricityPrice=input('Want to set the electricity selling price?: 1=yes, 0=no: ');
if choiceElectricityPrice==1
    eleprice=input('set the electricity selling price [$/kWh]: ');
    fprintf('The electricity selling price is %.3f [$/kWh]\n', eleprice);
else
    eleprice=0.09;
    fprintf('The electricity selling price is %.3f [$/kWh]\n', eleprice);
end

choiceCAPEXincentives=input('Want to consider incentives for CAPEX? 1=yes, 0=no: ');
if choiceCAPEXincentives==1
    choiceCAPEXincentivesValue=input('Want set incentives for CAPEX? 1=yes, 0=no: ');
    if choiceCAPEXincentivesValue==1
        CAPEXincentives=input('set capex incentives as percentage of the capex (all possible incentives
together). Value between 0 and 1: ');
        fprintf('the incentives are the %.2f of CAPEX\n',CAPEXincentives);
    else
        CAPEXincentives=0.2;
        fprintf('the incentives are the %.2f of CAPEX\n',CAPEXincentives);
    end
else
    CAPEXincentives=0;
    fprintf('the incentives are the %.2f of CAPEX\n',CAPEXincentives);
end

choiceOPEXincentives=input('Want to consider incentives for OPEX? 1=yes, 0=no: ');
if choiceOPEXincentives==1
    choiceOPEXincentivesValue=input('Want set incentives for OPEX? 1=yes, 0=no: ');
    if choiceOPEXincentivesValue==1

```

```

        OPEXincentives=input('set opex incentives as percentage of the opex (all possible incentives
together). Value between 0 and 1: ');
        fprintf('the incentives are the %.2f of OPEX\n',OPEXincentives);
    else
        OPEXincentives=0.1;
        fprintf('the incentives are the %.2f of OPEX\n',OPEXincentives);
    end
else
    OPEXincentives=0;
    fprintf('the incentives are the %.2f of OPEX\n',OPEXincentives);
end

choiceREVENUEincentives=input('Want to consider incentives for revenue? 1=yes, 0=no: ');
if choiceREVENUEincentives==1
    choiceREVENUEincentivesValue=input('Want set incentives for revenue? 1=yes, 0=no: ');
    if choiceREVENUEincentivesValue==1
        REVENUEincentives=input('set incentives as percentage of the revenue (all possible incentives
together). Value between 0 and 1: ');
        fprintf('the incentives are the %.2f of revenue\n',REVENUEincentives);
    else
        REVENUEincentives=0.3;
        fprintf('the incentives are the %.2f of revenue\n',REVENUEincentives);
    end
else
    REVENUEincentives=0;
    fprintf('the incentives are the %.2f of revenue\n',REVENUEincentives);
end

```

II. *Askinput_dry_steam*

```

%% Ambient condition
choiceAmbient=input('Want to insert set ambient condition? 1=yes, 0=no: ');
if choiceAmbient==1
    T0=input('set ambient temperature [°C]: ');
    p0=input('set ambient pressure [bar]: ');
else
    T0=20;
    p0=1;
    fprintf('The ambient temperature is %d °C and the ambient pressure is %d bar\n', T0, p0);
end

choiceInjectioTemperature=input('Want to insert set injection temperature? 1=yes, 0=no: ');
if choiceInjectioTemperature==1
    Tinj=input('Be warned that it must be lower than the temperature at turbine outlet. Set ambient
temperature [°C]: ');
    fprintf('The injection temperature is %d °C\n', Tinj);
else
    Tinj=25;
    fprintf('The injection temperature is %d °C\n', Tinj);
end

%% Plant lifetime and time step per year
choiceLifetimeValue=input('Want to set the lifetime of the plant? 1=yes, 0=no: ');
if choiceLifetimeValue==1
    lifetime=input('set the lifetime of the plant: ');
    fprintf('The lifetime is %d years\n', lifetime);
else
    lifetime=30;
    fprintf('The lifetime is %d years\n', lifetime);
end

choicetimestepperyearValue=input('Want to set the time step per year? 1=yes, 0=no: ');
if choicetimestepperyearValue==1
    time_step_per_year=input('set the time step per year: ');
    fprintf('The number of time step per year is %d \n', time_step_per_year);
else
    time_step_per_year=1;
    fprintf('The number of time step per year is %d \n', time_step_per_year);
end

%% Production and Injection parameters
choiceDprodValue=input('Want to set the diameter of production well? 1=yes, 0=no: ');
if choiceDprodValue==1
    d_prod=input('The diameter must be between 0.0254 and 0.7620. Set production well diameter: ');

```

```

while d_prod/0.0254<1 || d_prod/0.0254>30
    d_prod=input('The diameter must be between 0.0254 and 0.7620. Set production well diameter: ');
end
fprintf('The production well diameter is %f [m]\n', d_prod);
else
    d_prod=8*0.0254;
    fprintf('The production well diameter is %f [m]\n', d_prod);
end

choiceNprodValue=input('Want to set the number of production well? 1=yes, 0=no: ');
if choiceNprodValue==1
    n_prod=input('Set number of production wells: ');
    fprintf('The number of production wells is %d [-]\n', n_prod);
else
    n_prod=2;
    fprintf('The number of production wells is %d [-]\n', n_prod);
end

L_prod=0;
while L_prod==0
    L_prod=input('set the depth of the production well: ');
    fprintf('The production well depth is %.2f\n', L_prod);
    % Check if the input is valid (non-empty and a number)
    if isempty(L_prod) || ~isnumeric(L_prod)
        disp('Invalid input. Please enter a number. ');
        L_prod = 0; % Reset the value to stay in the loop
    else
        fprintf('The production well depth is %.2f\n', L_prod);
    end
end

choiceDinjValue=input('Want to set the diameter of injection well? 1=yes, 0=no: ');
if choiceDinjValue==1
    d_inj=input('The diameter must be between 0.0254 and 0.7620. Set injection well diameter: ');
    while d_inj/0.0254<1 || d_inj/0.0254>30
        d_inj=input('The diameter must be between 0.0254 and 0.7620. Set injection well diameter: ');
    end
    fprintf('The injection well diameter is %f [m]\n', d_inj);
else
    d_inj=8*0.0254;
    fprintf('The injection well diameter is %f [m]\n', d_inj);
end

choiceNinjValue=input('Want to set the number of injection well? 1=yes, 0=no: ');
if choiceNinjValue==1
    n_inj=input('Set number of injection wells: ');
    fprintf('The number of injection wells is %d [-]\n', n_inj);
else
    n_inj=1;
    fprintf('The number of injection wells is %d [-]\n', n_inj);
end

%% Filter's diameter
choiceDfilter=input('Want to set the diameter of the filter? 1=yes, 0=no: ');
if choiceDfilter==1
    D_filter=input('The diameter is worth to be between 0.1 and 0.5. Set filter diameter: ');
    fprintf('The diameter of the filter is %.2f [m]\n',D_filter);
else
    D_filter=0.4;
    fprintf('The diameter of the filter is %.2f [m]\n',D_filter);
end

%% Turbine outlet pressure
choiceTurbinePressureOutlet=input('Want to set the outlet pressure of turbine (p2)? 1=yes, 0=no: ');
if choiceTurbinePressureOutlet==1
    pressure_turb_outlet=input('set the outlet pressure of turbine (p2): ');
    fprintf('The outlet pressure of turbine is %f [bar]\n', pressure_turb_outlet);
else
    pressure_turb_outlet=0.12;
    fprintf('The outlet pressure of turbine is %f [bar]\n', pressure_turb_outlet);
end

choiceCompressorPressureOutlet=input('Want to set the outlet pressure of turbine? 1=yes, 0=no: ');
if choiceCompressorPressureOutlet==1

```

```

    pressure_compr_outlet=input('set the outlet pressure of turbine: ');
    fprintf('The outlet pressure of turbine is %f [bar]\n', pressure_compr_outlet);
else
    pressure_compr_outlet=p0;
    fprintf('The outlet pressure of turbine is %f [bar]\n', pressure_compr_outlet);
end

%% Efficiency of machinery
choiceEtaTurbValue=input('Want to set turbine efficiency? 1=yes, 0=no: ');
if choiceEtaTurbValue==1
    eta_turb=input('set turbine efficiency: ');
    while eta_turb<=0 || eta_turb>=1
        eta_turb=input('The efficiency must be between 0 and 1. Set turbine efficiency: ');
    end
    fprintf('The turbine efficiency is %f [-]\n', eta_turb);
else
    eta_turb=0.85;
    fprintf('The turbine efficiency is %f [-]\n', eta_turb);
end

choiceXvaplimValue=input('Want to set the admissible fraction of vapour expanding in turbine? 1=yes, 0=no: ');
if choiceXvaplimValue==1
    xvap_lim=input('set the admissible fraction of vapour : ');
    if xvap_lim>1
        xvap_lim=1;
        fprintf('The admissible fraction of vapour is %f [-]\n', xvap_lim);
    end
    if xvap_lim<0.8
        xvap_lim=0.8;
        fprintf('The admissible fraction of vapour is %f [-]\n', xvap_lim);
    end
else
    xvap_lim=0.8;
    fprintf('The admissible fraction of vapour is %f [-]\n', xvap_lim);
end

choiceEtaCondPumpValue=input('Want to set the efficiency of condensation pump? 1=yes, 0=no: ');
if choiceEtaCondPumpValue==1
    eta_condpump=input('set condensation pump efficiency: ');
    while eta_condpump<=0 || eta_condpump>=1
        eta_condpump=input('The efficiency must be between 0 and 1. Set condensation pump efficiency: ');
    end
    fprintf('The condensation pump efficiency is %f [-]\n', eta_condpump);
else
    eta_condpump=0.85;
    fprintf('The condensation pump efficiency is %f [-]\n', eta_condpump);
end

choiceEtaBDPumpValue=input('Want to set the efficiency of blowdown pump? 1=yes, 0=no: ');
if choiceEtaBDPumpValue==1
    eta_bdpump=input('set blowdown pump efficiency: ');
    while eta_bdpump<=0 || eta_bdpump>=1
        eta_bdpump=input('The efficiency must be between 0 and 1. Set blowdown pump efficiency: ');
    end
    fprintf('The blowdown pump efficiency is %f [-]\n', eta_bdpump);
else
    eta_bdpump=0.85;
    fprintf('The blowdown pump efficiency is %f [-]\n', eta_bdpump);
end

choiceEtaCircPumpValue=input('Want to set the efficiency of circulation pump? 1=yes, 0=no: ');
if choiceEtaCircPumpValue==1
    eta_circpump=input('set circulation pump efficiency: ');
    while eta_circpump<=0 || eta_circpump>=1
        eta_circpump=input('The efficiency must be between 0 and 1. Set circulation pump efficiency: ');
    end
    fprintf('The circulation pump efficiency is %f [-]\n', eta_circpump);
else
    eta_circpump=0.85;
    fprintf('The circulation pump efficiency is %f [-]\n', eta_circpump);
end

choiceEtaGenerator=input('Want to set the efficiency of generator? 1=yes, 0=no: ');
if choiceEtaGenerator==1

```

```

eta_gen=input('set circulation pump efficiency: ');
while eta_gen<=0 || eta_gen>=1
    eta_gen=input('The efficiency must be between 0 and 1. Set circulation pump efficiency: ');
end
fprintf('The circulation pump efficiency is %f [-]\n', eta_gen);
else
    eta_gen=0.95;
    fprintf('The circulation pump efficiency is %f [-]\n', eta_gen);
end

choiceEtaCompressor=input('Want to set the isentropic efficiency of compressor? 1=yes, 0=no: ');
if choiceEtaCompressor==1
    eta_compr=input('set circulation pump efficiency: ');
    while eta_compr<=0 || eta_compr>=1
        eta_compr=input('The efficiency must be between 0 and 1. Set circulation pump efficiency: ');
    end
    fprintf('The circulation pump efficiency is %f [-]\n', eta_compr);
else
    eta_compr=0.85;
    fprintf('The circulation pump efficiency is %f [-]\n', eta_compr);
end

%% Non-condensable gases
choiceXnonCondGas=input('Want to set the percentage of non-condensable gases at turbine outlet? 1=yes, 0=no: ');
if choiceXnonCondGas==1
    x_ncg=input('set percentage of non-condensable gases: ');
    fprintf('The percentage of non-condensable gases is %.3f [-]\n',x_ncg);
else
    x_ncg=0.05;
    fprintf('The percentage of non-condensable gases is %.3f [-]\n',x_ncg);
end

choiceCompositionUncondensedGas=input('Want to set the composition of uncondensed gas (CO2,H2S,Hg)? The sum must be equal to 1! 1=yes, 0=no: ');
if choiceCompositionUncondensedGas==1
    check_sum_uncondensed_gas=0;
    while check_sum_uncondensed_gas>1 || check_sum_uncondensed_gas<1
        xCO2=input('set the percentage of CO2: ');
        fprintf('The percentage of CO2 is %.3f [-]\n',xCO2);
        xH2S=input('set the percentage of H2S: ');
        fprintf('The percentage of H2S is %.3f [-]\n',xH2S);
        xHg=input('set the percentage of Hg: ');
        fprintf('The percentage of Hg is %.3f [-]\n',xHg);
        check_sum_uncondensed_gas=xHg+xH2S+xCO2;
    end
else
    xCO2=0.95;
    fprintf('The percentage of CO2 is %.3f [-]\n',xCO2);
    xH2S=0.04;
    fprintf('The percentage of H2S is %.3f [-]\n',xH2S);
    xHg=0.01;
    fprintf('The percentage of Hg+ is %.3f [-]\n',xHg);
end

%% cooling tower heat/electricity ratio
choiceHeatToElectricityRatio=input('Want to set the heat to electricity ratio for cooling tower? 1=yes, 0=no: ');
if choiceHeatToElectricityRatio==1
    kWt_to_kWe=input('This value can be equal to 40 or 80. Set heat to electricity ratio: ');
    fprintf('The heat to electricity ratio is %.1f [-]\n',kWt_to_kWe);
else
    kWt_to_kWe=40;
    fprintf('The heat to electricity ratio is %.1f [-]\n',kWt_to_kWe);
end

%% economic parameter
choiceGeologicalFormation=input('Want to set the geological formation? 1=yes, 0=no: ');
if choiceGeologicalFormation==1
    sourcetype=input('Set the type of geological formation. 1=sedimentary, 2=fractured, 3=basement: ');
    if sourcetype==1
        Succrate=0.95;
        fprintf('The geological formation is sedimentary, and the success rate in drilling is %.2f [-]\n',Succrate);
    elseif sourcetype==2

```

```

        Succrate=0.75;
        fprintf('The geological formation is fractured, and the success rate in drilling is %.2f [-
]\n',Succrate);
    else
        Succrate=0.9;
        fprintf('The geological formation is basement, and the success rate in drilling is %.2f [-
]\n',Succrate);
    end
else
    Succrate=0.9;
    fprintf('The geological formation is basement, and the success rate in drilling is %.2f [-]\n',Succrate);
end

choiceFieldType=input('Want to choose the field type? 1=yes, 0=no: ');
if choiceFieldType==1
    fieldtype=input('Set 1 for brownfield and 2 for greenfield: ');
    if fieldtype==1
        fprintf('The fieldtype chosen is brownfield [-]\n');
    else
        fprintf('The fieldtype chosen is greenfield [-]\n');
    end
else
    fieldtype=2;
    fprintf('The fieldtype chosen is greenfield [-]\n');
end

choiceCapacityFactor=input('Want to set the capacity factor? 1=yes, 0=no: ');
if choiceCapacityFactor==1
    CF=input('set the capacity factor (suggested between 0.7 and 0.9):');
    fprintf('the capacity factor is %.2f [-]\n', CF);
else
    CF=0.82;
    fprintf('the capacity factor is %.2f [-]\n', CF);
end

choiceElectricityPrice=input('Want to set the electricity selling price?: 1=yes, 0=no: ');
if choiceElectricityPrice==1
    eleprice=input('set the electricity selling price [$/kWh]: ');
    fprintf('The electricity selling price is %.3f [$/kWh]\n', eleprice);
else
    eleprice=0.09;
    fprintf('The electricity selling price is %.3f [$/kWh]\n', eleprice);
end

choiceCAPEXincentives=input('Want to consider incentives for CAPEX? 1=yes, 0=no: ');
if choiceCAPEXincentives==1
    choiceCAPEXincentivesValue=input('Want set incentives for CAPEX? 1=yes, 0=no: ');
    if choiceCAPEXincentivesValue==1
        CAPEXincentives=input('set capex incentives as percentage of the capex (all possible incentives
together). Value between 0 and 1: ');
        fprintf('the incentives are the %.2f of CAPEX\n',CAPEXincentives);
    else
        CAPEXincentives=0.2;
        fprintf('the incentives are the %.2f of CAPEX\n',CAPEXincentives);
    end
else
    CAPEXincentives=0;
    fprintf('the incentives are the %.2f of CAPEX\n',CAPEXincentives);
end

choiceOPEXincentives=input('Want to consider incentives for OPEX? 1=yes, 0=no: ');
if choiceOPEXincentives==1
    choiceOPEXincentivesValue=input('Want set incentives for OPEX? 1=yes, 0=no: ');
    if choiceOPEXincentivesValue==1
        OPEXincentives=input('set opex incentives as percentage of the opex (all possible incentives
together). Value between 0 and 1: ');
        fprintf('the incentives are the %.2f of OPEX\n',OPEXincentives);
    else
        OPEXincentives=0.1;
        fprintf('the incentives are the %.2f of OPEX\n',OPEXincentives);
    end
else
    OPEXincentives=0;
    fprintf('the incentives are the %.2f of OPEX\n',OPEXincentives);
end

```

```

choiceREVENUEincentives=input('Want to consider incentives for revenue? 1=yes, 0=no: ');
if choiceREVENUEincentives==1
    choiceREVENUEincentivesValue=input('Want set incentives for revenue? 1=yes, 0=no: ');
    if choiceREVENUEincentivesValue==1
        REVENUEincentives=input('set incentives as percentage of the revenue (all possible incentives
together). Value between 0 and 1: ');
        fprintf('the incentives are the %.2f of revenue\n',REVENUEincentives);
    else
        REVENUEincentives=0.3;
        fprintf('the incentives are the %.2f of revenue\n',REVENUEincentives);
    end
end
else
    REVENUEincentives=0;
    fprintf('the incentives are the %.2f of revenue\n',REVENUEincentives);
end
end

```

III. *Asinput_dry_steam_top_spillation*

```

%% Ambient condition
choiceAmbient=input('Want to insert set ambient condition? 1=yes, 0=no: ');
if choiceAmbient==1
    T0=input('set ambient temperature [°C]: ');
    p0=input('set ambient pressure [bar]: ');
else
    T0=20;
    p0=1;
    fprintf('The ambient temperature is %d °C and the ambient pressure is %d bar\n', T0, p0);
end

%% Plant lifetime and time step per year
choiceLifetimeValue=input('Want to set the lifetime of the plant? 1=yes, 0=no: ');
if choiceLifetimeValue==1
    lifetime=input('set the lifetime of the plant: ');
    fprintf('The lifetime is %d years\n', lifetime);
else
    lifetime=30;
    fprintf('The lifetime is %d years\n', lifetime);
end

choicetimestepperyearValue=input('Want to set the time step per year? 1=yes, 0=no: ');
if choicetimestepperyearValue==1
    timestepperyear=input('set the time step per year: ');
    fprintf('The number of time step per year is %d \n', timestepperyear);
else
    timestepperyear=1;
    fprintf('The number of time step per year is %d \n', timestepperyear);
end

%% Production and injection parameters
choiceDprodValue=input('Want to set the diameter of production well? 1=yes, 0=no: ');
if choiceDprodValue==1
    d_prod=input('The diameter must be between 0.0254 and 0.7620. Set production well diameter: ');
    while d_prod/0.0254<1 || d_prod/0.0254>30
        d_prod=input('The diameter must be between 0.0254 and 0.7620. Set production well diameter: ');
    end
    fprintf('The production well diameter is %f [m]\n', d_prod);
else
    d_prod=8*0.0254;
    fprintf('The production well diameter is %f [m]\n', d_prod);
end

choiceNprodValue=input('Want to set the number of production well? 1=yes, 0=no: ');
if choiceNprodValue==1
    n_prod=input('Set number of production wells: ');
    fprintf('The number of production wells is %d [-]\n', n_prod);
else
    n_prod=2;
    fprintf('The number of production wells is %d [-]\n', n_prod);
end

L_prod=0;
while L_prod==0
    L_prod=input('set the depth of the production well: ');
end

```

```

fprintf('The production well depth is %.2f\n', L_prod);
% Check if the input is valid (non-empty and a number)
if isempty(L_prod) || ~isnumeric(L_prod)
    disp('Invalid input. Please enter a number. ');
    L_prod = 0; % Reset the value to stay in the loop
else
    fprintf('The production well depth is %.2f\n', L_prod);
end
end

choiceDinjValue=input('Want to set the diameter of injection well? 1=yes, 0=no: ');
if choiceDinjValue==1
    d_inj=input('The diameter must be between 0.0254 and 0.7620. Set injection well diameter: ');
    while d_inj/0.0254<1 || d_inj/0.0254>30
        d_inj=input('The diameter must be between 0.0254 and 0.7620. Set injection well diameter: ');
    end
    fprintf('The injection well diameter is %f [m]\n', d_inj);
else
    d_inj=8*0.0254;
    fprintf('The injection well diameter is %f [m]\n', d_inj);
end

choiceNinjValue=input('Want to set the number of injection well? 1=yes, 0=no: ');
if choiceNinjValue==1
    n_inj=input('Set number of injection wells: ');
    fprintf('The number of injection wells is %d [-]\n', n_inj);
else
    n_inj=1;
    fprintf('The number of injection wells is %d [-]\n', n_inj);
end

%% Filter's diameter
choiceDfilter=input('Want to set the diameter of the filter? 1=yes, 0=no: ');
if choiceDfilter==1
    D_filter=input('The diameter is worth to be between 0.1 and 0.5. Set filter diameter: ');
    fprintf('The diameter of the filter is %.2f [m]\n',D_filter);
else
    D_filter=0.4;
    fprintf('The diameter of the filter is %.2f [m]\n',D_filter);
end

%% Spallation fraction
choiceXspillation=input('Want to set the spilled fraction of vapour? 1=yes, 0=no: ');
if choiceXspillation==1
    x_enduse=input('Insert a number between 0 and 1: ');
    while x_enduse<=0 || x_enduse>1
        x_enduse=input('Insert a number between 0 and 1: ');
    end
else
    x_enduse=0.3;
end

%% Turbine outlet pressure
choiceTurbinePressureOutlet=input('Want to set the outlet pressure of turbine?: 1=yes, 0=no: ');
if choiceTurbinePressureOutlet==1
    pressure_turb_outlet=input('set the outlet pressure of turbine: ');
    fprintf('The outlet pressure of turbine is %f [bar]\n', pressure_turb_outlet);
else
    pressure_turb_outlet=0.12;
    fprintf('The outlet pressure of turbine is %f [bar]\n', pressure_turb_outlet);
end

choiceCompressorPressureOutlet=input('Want to set the outlet pressure of turbine?: 1=yes, 0=no: ');
if choiceCompressorPressureOutlet==1
    pressure_compr_outlet=input('set the outlet pressure of compressor: ');
    fprintf('The outlet pressure of compressor is %f [bar]\n', pressure_compr_outlet);
else
    pressure_compr_outlet=p0;
    fprintf('The outlet pressure of compressor is %f [bar]\n', pressure_compr_outlet);
end

%% Efficiency of machinery
choiceEtaTurbValue=input('Want to set turbine efficiency? 1=yes, 0=no: ');
if choiceEtaTurbValue==1
    eta_turb=input('set turbine efficiency: ');
    while eta_turb<=0 || eta_turb>=1

```

```

        eta_turb=input('The efficiency must be between 0 and 1. Set turbine efficiency: ');
    end
    fprintf('The turbine efficiency is %f [-]\n', eta_turb);
else
    eta_turb=0.85;
    fprintf('The turbine efficiency is %f [-]\n', eta_turb);
end

choiceXvaplimValue=input('Want to set the admissible fraction of vapour expanding in turbine? 1=yes, 0=no: ');
if choiceXvaplimValue==1
    xvap_lim=input('set the admissible fraction of vapour: ');
    if xvap_lim>1
        xvap_lim=1;
        fprintf('The admissible fraction of vapour is %f [-]\n', xvap_lim);
    end
    if xvap_lim<0.8
        xvap_lim=0.85;
        fprintf('The admissible fraction of vapour is %f [-]\n', xvap_lim);
    end
else
    xvap_lim=0.8;
    fprintf('The admissible fraction of vapour is %f [-]\n', xvap_lim);
end

choiceEtaCondPumpValue=input('Want to set the efficiency of condensation pump? 1=yes, 0=no: ');
if choiceEtaCondPumpValue==1
    eta_condpump=input('set condensation pump efficiency: ');
    while eta_condpump<=0 || eta_condpump>=1
        eta_condpump=input('The efficiency must be between 0 and 1. Set condensation pump efficiency: ');
    end
    fprintf('The condensation pump efficiency is %f [-]\n', eta_condpump);
else
    eta_condpump=0.85;
    fprintf('The condensation pump efficiency is %f [-]\n', eta_condpump);
end

choiceEtaBDPumpValue=input('Want to set the efficiency of blowdown pump? 1=yes, 0=no: ');
if choiceEtaBDPumpValue==1
    eta_bdpump=input('set blowdown pump efficiency: ');
    while eta_bdpump<=0 || eta_bdpump>=1
        eta_bdpump=input('The efficiency must be between 0 and 1. Set blowdown pump efficiency: ');
    end
    fprintf('The blowdown pump efficiency is %f [-]\n', eta_bdpump);
else
    eta_bdpump=0.85;
    fprintf('The blowdown pump efficiency is %f [-]\n', eta_bdpump);
end

choiceEtaCircPumpValue=input('Want to set the efficiency of circulation pump? 1=yes, 0=no: ');
if choiceEtaCircPumpValue==1
    eta_circpump=input('set circulation pump efficiency: ');
    while eta_circpump<=0 || eta_circpump>=1
        eta_circpump=input('The efficiency must be between 0 and 1. Set circulation pump efficiency: ');
    end
    fprintf('The circulation pump efficiency is %f [-]\n', eta_circpump);
else
    eta_circpump=0.85;
    fprintf('The circulation pump efficiency is %f [-]\n', eta_circpump);
end

choiceEtaEnduse=input('Want to set the end-use efficiency? 1=yes, 0=no: ');
if choiceEtaEnduse==1
    eta_enduse=input('set end-use efficiency: ');
    while eta_enduse<=0 || eta_enduse>=1
        eta_enduse=input('The efficiency must be between 0 and 1. Set end-use efficiency: ');
    end
    fprintf('The end-use efficiency is %f [-]\n', eta_enduse);
else
    eta_enduse=0.9;
    fprintf('The end-use efficiency is %f [-]\n', eta_enduse);
end

choiceEtaGenerator=input('Want to set the efficiency of generator? 1=yes, 0=no: ');

```

```

if choiceEtaGenerator==1
    eta_gen=input('set circulation pump efficiency: ');
    while eta_gen<=0 || eta_gen>=1
        eta_gen=input('The efficiency must be between 0 and 1. Set circulation pump efficiency: ');
    end
    fprintf('The circulation pump efficiency is %f [-]\n', eta_gen);
else
    eta_gen=0.95;
    fprintf('The circulation pump efficiency is %f [-]\n', eta_gen);
end

choiceEtaCompressor=input('Want to set the isentropic efficiency of compressor? 1=yes, 0=no: ');
if choiceEtaCompressor==1
    eta_compr=input('set circulation pump efficiency: ');
    while eta_compr<=0 || eta_compr>=1
        eta_compr=input('The efficiency must be between 0 and 1. Set circulation pump efficiency: ');
    end
    fprintf('The circulation pump efficiency is %f [-]\n', eta_compr);
else
    eta_compr=0.85;
    fprintf('The circulation pump efficiency is %f [-]\n', eta_compr);
end

%% Non-condensable gases
choiceXnonCondGas=input('Want to set the percentage of non-condensable gases at turbine outlet? 1=yes, 0=no: ');
if choiceXnonCondGas==1
    x_ncg=input('set percentage of non-condensable gases: ');
    fprintf('The percentage of non-condensable gases is %.3f [-]\n',x_ncg);
else
    x_ncg=0.05;
    fprintf('The percentage of non-condensable gases is %.3f [-]\n',x_ncg);
end

choiceCompositionUncondensedGas=input('Want to set the composition of uncondensed gas (CO2,H2S,Hg)? The sum must be equal to 1! 1=yes, 0=no: ');
if choiceCompositionUncondensedGas==1
    check_sum_uncondensed_gas=0;
    while check_sum_uncondensed_gas>1 || check_sum_uncondensed_gas<1
        xCO2=input('set the percentage of CO2: ');
        fprintf('The percentage of CO2 is %.3f [-]\n',xCO2);
        xH2S=input('set the percentage of H2S: ');
        fprintf('The percentage of H2S is %.3f [-]\n',xH2S);
        xHg=input('set the percentage of Hg: ');
        fprintf('The percentage of Hg+ is %.3f [-]\n',xHg);
        check_sum_uncondensed_gas=xHg+xH2S+xCO2;
    end
else
    xCO2=0.95;
    fprintf('The percentage of CO2 is %.3f [-]\n',xCO2);
    xH2S=0.04;
    fprintf('The percentage of H2S is %.3f [-]\n',xH2S);
    xHg=0.01;
    fprintf('The percentage of Hg+ is %.3f [-]\n',xHg);
end

%% cooling tower heat/electricity ratio
choiceHeatToElectricityRatio=input('Want to set the heat to electricity ratio for cooling tower? 1=yes, 0=no: ');
if choiceHeatToElectricityRatio==1
    kWt_to_kWe=input('This value can be equal to 40 or 80. Set heat to electricity ratio: ');
    fprintf('The heat to electricity ratio is %.1f [-]\n',kWt_to_kWe);
else
    kWt_to_kWe=40;
    fprintf('The heat to electricity ratio is %.1f [-]\n',kWt_to_kWe);
end

%% economic parameter
choiceGeologicalFormation=input('Want to set the geological formation? 1=yes, 0=no: ');
if choiceGeologicalFormation==1
    sourcetype=input('Set the type of geological formation. 1=sedimentary, 2=fractured, 3=basement: ');
    if sourcetype==1
        Succrate=0.95;
    end
end

```

```

        fprintf('The geological formation is sedimentary, and the success rate in drilling is %.2f [-
]\n',Succrate);
    elseif sourcetype==2
        Succrate=0.75;
        fprintf('The geological formation is fractured, and the success rate in drilling is %.2f [-
]\n',Succrate);
    else
        Succrate=0.9;
        fprintf('The geological formation is basement, and the success rate in drilling is %.2f [-
]\n',Succrate);
    end
else
    Succrate=0.9;
    fprintf('The geological formation is basement, and the success rate in drilling is %.2f [-]\n',Succrate);
end

choiceFieldType=input('Want to choose the field type? 1=yes, 0=no: ');
if choiceFieldType==1
    fieldtype=input('Set 1 for brownfield and 2 for greenfield: ');
    if fieldtype==1
        fprintf('The fieldtype chosen is brownfield [-]\n');
    else
        fprintf('The fieldtype chosen is greenfield [-]\n');
    end
else
    fieldtype=2;
    fprintf('The fieldtype chosen is greenfield [-]\n');
end

choiceCapacityFactor=input('Want to set the capacity factor? 1=yes, 0=no: ');
if choiceCapacityFactor==1
    CF=input('set the capacity factor (suggested between 0.7 and 0.9): ');
    fprintf('the capacity factor is %.2f [-]\n', CF);
else
    CF=0.82;
    fprintf('the capacity factor is %.2f [-]\n', CF);
end

choiceElectricityPrice=input('Want to set the electricity selling price?: 1=yes, 0=no: ');
if choiceElectricityPrice==1
    eleprice=input('set the electricity selling price [$/kWh]: ');
    fprintf('The electricity selling price is %.3f [$/kWh]\n', eleprice);
else
    eleprice=0.09;
    fprintf('The electricity selling price is %.3f [$/kWh]\n', eleprice);
end

choiceHeatPrice=input('Want to set the heat selling price?: 1=yes, 0=no: ');
if choiceHeatPrice==1
    heatprice=input('set the heat selling price [$/kWh]: ');
    fprintf('The heat selling price is %.3f [$/kWh]\n', heatprice);
else
    heatprice=0.02;
    fprintf('The heat selling price is %.3f [$/kWh]\n', heatprice);
end

choiceCAPEXincentives=input('Want to consider incentives for CAPEX? 1=yes, 0=no: ');
if choiceCAPEXincentives==1
    choiceCAPEXincentivesValue=input('Want set incentives for CAPEX? 1=yes, 0=no: ');
    if choiceCAPEXincentivesValue==1
        CAPEXincentives=input('set capex incentives as percentage of the capex (all possible incentives
together). Value between 0 and 1: ');
        fprintf('the incentives are the %.2f of CAPEX\n',CAPEXincentives);
    else
        CAPEXincentives=0.2;
        fprintf('the incentives are the %.2f of CAPEX\n',CAPEXincentives);
    end
else
    CAPEXincentives=0;
    fprintf('the incentives are the %.2f of CAPEX\n',CAPEXincentives);
end

choiceOPEXincentives=input('Want to consider incentives for OPEX? 1=yes, 0=no: ');
if choiceOPEXincentives==1
    choiceOPEXincentivesValue=input('Want set incentives for OPEX? 1=yes, 0=no: ');

```

```

    if choiceOPEXincentivesValue==1
        OPEXincentives=input('set opex incentives as percentage of the opex (all possible incentives
together). Value between 0 and 1: ');
        fprintf('the incentives are the %.2f of OPEX\n',OPEXincentives);
    else
        OPEXincentives=0.1;
        fprintf('the incentives are the %.2f of OPEX\n',OPEXincentives);
    end
else
    OPEXincentives=0;
    fprintf('the incentives are the %.2f of OPEX\n',OPEXincentives);
end

```

```

choiceREVENUEincentives=input('Want to consider incentives for revenue? 1=yes, 0=no: ');
if choiceREVENUEincentives==1
    choiceREVENUEincentivesValue=input('Want set incentives for revenue? 1=yes, 0=no: ');
    if choiceREVENUEincentivesValue==1
        REVENUEincentives=input('set incentives as percentage of the revenue (all possible incentives
together). Value between 0 and 1: ');
        fprintf('the incentives are the %.2f of revenue\n',REVENUEincentives);
    else
        REVENUEincentives=0.3;
        fprintf('the incentives are the %.2f of revenue\n',REVENUEincentives);
    end
else
    REVENUEincentives=0;
    fprintf('the incentives are the %.2f of revenue\n',REVENUEincentives);
end

```

IV. *Askinput_single_flash*

```

%% Ambient condition
choiceAmbient=input('Want to insert set ambient condition? 1=yes, 0=no: ');
if choiceAmbient==1
    T0=input('set ambient temperature [°C]: ');
    p0=input('set ambient pressure [bar]: ');
else
    T0=20;
    p0=1;
    fprintf('The ambient temperature is %d °C and the ambient pressure is %d bar\n', T0, p0);
end

%% Plant lifetime and time step per year
choicelifetimeValue=input('Want to set the lifetime of the plant? 1=yes, 0=no: ');
if choiceLifetimeValue==1
    lifetime=input('set the lifetime of the plant: ');
    fprintf('The lifetime is %d years\n', lifetime);
else
    lifetime=30;
    fprintf('The lifetime is %d years\n', lifetime);
end

choicetimestepperyearValue=input('Want to set the time step per year? 1=yes, 0=no: ');
if choicetimestepperyearValue==1
    timestepperyear=input('set the time step per year: ');
    fprintf('The number of time step per year is %d \n', timestepperyear);
else
    timestepperyear=1;
    fprintf('The number of time step per year is %d \n', timestepperyear);
end

%% Production and injection parameters
choiceDprodValue=input('Want to set the diameter of production well? 1=yes, 0=no: ');
if choiceDprodValue==1
    d_prod=input('The diameter must be between 0.0254 and 0.7620. Set production well diameter: ');
    while d_prod/0.0254<1 || d_prod/0.0254>30
        d_prod=input('The diameter must be between 0.0254 and 0.7620. Set production well diameter: ');
    end
    fprintf('The production well diameter is %f [m]\n', d_prod);
else
    d_prod=8*0.0254;
    fprintf('The production well diameter is %f [m]\n', d_prod);
end

choiceNprodValue=input('Want to set the number of production well? 1=yes, 0=no: ');

```

```

if choiceNprodValue==1
    n_prod=input('Set number of production wells: ');
    fprintf('The number of production wells is %d [-]\n', n_prod);
else
    n_prod=2;
    fprintf('The number of production wells is %d [-]\n', n_prod);
end

L_prod=0;
while L_prod==0
    L_prod=input('set the depth of the production well: ');
    fprintf('The production well depth is %.2f\n', L_prod);
    % Check if the input is valid (non-empty and a number)
    if isempty(L_prod) || ~isnumeric(L_prod)
        disp('Invalid input. Please enter a number. ');
        L_prod = 0; % Reset the value to stay in the loop
    else
        fprintf('The production well depth is %.2f\n', L_prod);
    end
end

choiceDinjValue=input('Want to set the diameter of injection well? 1=yes, 0=no: ');
if choiceDinjValue==1
    d_inj=input('The diameter must be between 0.0254 and 0.7620. Set injection well diameter: ');
    while d_inj/0.0254<1 || d_inj/0.0254>30
        d_inj=input('The diameter must be between 0.0254 and 0.7620. Set injection well diameter: ');
    end
    fprintf('The injection well diameter is %f [m]\n', d_inj);
else
    d_inj=8*0.0254;
    fprintf('The injection well diameter is %f [m]\n', d_inj);
end

choiceNinjValue=input('Want to set the number of injection well? 1=yes, 0=no: ');
if choiceNinjValue==1
    n_inj=input('Set number of injection wells: ');
    fprintf('The number of injection wells is %d [-]\n', n_inj);
else
    n_inj=1;
    fprintf('The number of injection wells is %d [-]\n', n_inj);
end

%% Filter's diameter
choiceDfilter=input('Want to set the diameter of the filter? 1=yes, 0=no: ');
if choiceDfilter==1
    D_filter=input('The diameter is worth to be between 0.1 and 0.5. Set filter diameter: ');
    fprintf('The diameter of the filter is %.2f [m]\n',D_filter);
else
    D_filter=0.4;
    fprintf('The diameter of the filter is %.2f [m]\n',D_filter);
end

%% Turbine outlet pressure
choiceTurbinePressureOutlet=input('Want to set the outlet pressure of turbine (p2)? 1=yes, 0=no: ');
if choiceTurbinePressureOutlet==1
    pressure_turb_outlet=input('set the outlet pressure of turbine (p2): ');
    fprintf('The outlet pressure of turbine is %f [bar]\n', pressure_turb_outlet);
else
    pressure_turb_outlet=0.12;
    fprintf('The outlet pressure of turbine is %f [bar]\n', pressure_turb_outlet);
end

choiceCompressorPressureOutlet=input('Want to set the outlet pressure of compressor? 1=yes, 0=no: ');
if choiceCompressorPressureOutlet==1
    pressure_compr_outlet=input('set the outlet pressure of compressor: ');
    fprintf('The outlet pressure of compressor is %f [bar]\n', pressure_compr_outlet);
else
    pressure_compr_outlet=p0;
    fprintf('The outlet pressure of compressor is %f [bar]\n', pressure_compr_outlet);
end

%% Efficiency of machinery
choiceEtaTurbValue=input('Want to set turbine efficiency? 1=yes, 0=no: ');
if choiceEtaTurbValue==1
    eta_turb=input('set turbine efficiency: ');
    while eta_turb<=0 || eta_turb>=1

```

```

        eta_turb=input('The efficiency must be between 0 and 1. Set turbine efficiency: ');
    end
    fprintf('The turbine efficiency is %f [-]\n', eta_turb);
else
    eta_turb=0.85;
    fprintf('The turbine efficiency is %f [-]\n', eta_turb);
end

choiceXvaplimValue=input('Want to set the admissible fraction of vapour expanding in turbine? 1=yes, 0=no: ');
if choiceXvaplimValue==1
    xvap_lim=input('set the admissible fraction of vapour : ');
    if xvap_lim>1
        xvap_lim=1;
        fprintf('The admissible fraction of vapour is %f [-]\n', xvap_lim);
    end
    if xvap_lim<0.8
        xvap_lim=0.85;
        fprintf('The admissible fraction of vapour is %f [-]\n', xvap_lim);
    end
else
    xvap_lim=0.8;
    fprintf('The admissible fraction of vapour is %f [-]\n', xvap_lim);
end

choiceEtaCondPumpValue=input('Want to set the efficiency of condensation pump? 1=yes, 0=no: ');
if choiceEtaCondPumpValue==1
    eta_condpump=input('set condensation pump efficiency: ');
    while eta_condpump<=0 || eta_condpump>=1
        eta_condpump=input('The efficiency must be between 0 and 1. Set condensation pump efficiency: ');
    end
    fprintf('The condensation pump efficiency is %f [-]\n', eta_condpump);
else
    eta_condpump=0.85;
    fprintf('The condensation pump efficiency is %f [-]\n', eta_condpump);
end

choiceEtaBDPumpValue=input('Want to set the efficiency of blowdown pump? 1=yes, 0=no: ');
if choiceEtaBDPumpValue==1
    eta_bdpump=input('set blowdown pump efficiency: ');
    while eta_bdpump<=0 || eta_bdpump>=1
        eta_bdpump=input('The efficiency must be between 0 and 1. Set blowdown pump efficiency: ');
    end
    fprintf('The blowdown pump efficiency is %f [-]\n', eta_bdpump);
else
    eta_bdpump=0.85;
    fprintf('The blowdown pump efficiency is %f [-]\n', eta_bdpump);
end

choiceEtaCircPumpValue=input('Want to set the efficiency of circulation pump? 1=yes, 0=no: ');
if choiceEtaCircPumpValue==1
    eta_circpump=input('set circulation pump efficiency: ');
    while eta_circpump<=0 || eta_circpump>=1
        eta_circpump=input('The efficiency must be between 0 and 1. Set circulation pump efficiency: ');
    end
    fprintf('The circulation pump efficiency is %f [-]\n', eta_circpump);
else
    eta_circpump=0.85;
    fprintf('The circulation pump efficiency is %f [-]\n', eta_circpump);
end

choiceEtaGenerator=input('Want to set the efficiency of generator? 1=yes, 0=no: ');
if choiceEtaGenerator==1
    eta_gen=input('set circulation pump efficiency: ');
    while eta_gen<=0 || eta_gen>=1
        eta_gen=input('The efficiency must be between 0 and 1. Set circulation pump efficiency: ');
    end
    fprintf('The circulation pump efficiency is %f [-]\n', eta_gen);
else
    eta_gen=0.95;
    fprintf('The circulation pump efficiency is %f [-]\n', eta_gen);
end

choiceEtaCompressor=input('Want to set the isentropic efficiency of compressor? 1=yes, 0=no: ');

```

```

if choiceEtaCompressor==1
    eta_compr=input('set circulation pump efficiency: ');
    while eta_compr<=0 || eta_compr>=1
        eta_compr=input('The efficiency must be between 0 and 1. Set circulation pump efficiency: ');
    end
    fprintf('The circulation pump efficiency is %f [-]\n', eta_compr);
else
    eta_compr=0.85;
    fprintf('The circulation pump efficiency is %f [-]\n', eta_compr);
end

%% Non-condensable gases
choiceXnonCondGas=input('Want to set the percentage of non-condensable gases at turbine outlet? 1=yes, 0=no: ');
if choiceXnonCondGas==1
    x_ncg=input('set percentage of non-condensable gases: ');
    fprintf('The percentage of non-condensable gases is %.3f [-]\n',x_ncg);
else
    x_ncg=0.05;
    fprintf('The percentage of non-condensable gases is %.3f [-]\n',x_ncg);
end

choiceCompositionUncondensedGas=input('Want to set the composition of uncondensed gas (CO2,H2S,Hg+)? The sum must be equal to 1! 1=yes, 0=no: ');
if choiceCompositionUncondensedGas==1
    check_sum_uncondensed_gas=0;
    while check_sum_uncondensed_gas>1 || check_sum_uncondensed_gas<1
        xCO2=input('set the percentage of CO2: ');
        fprintf('The percentage of CO2 is %.3f [-]\n',xCO2);
        xH2S=input('set the percentage of H2S: ');
        fprintf('The percentage of H2S is %.3f [-]\n',xH2S);
        xHg=input('set the percentage of Hg+: ');
        fprintf('The percentage of Hg+ is %.3f [-]\n',xHg);
        check_sum_uncondensed_gas=xHg+xH2S+xCO2;
    end
else
    xCO2=0.95;
    fprintf('The percentage of CO2 is %.3f [-]\n',xCO2);
    xH2S=0.04;
    fprintf('The percentage of H2S is %.3f [-]\n',xH2S);
    xHg=0.01;
    fprintf('The percentage of Hg+ is %.3f [-]\n',xHg);
end

%% cooling tower heat/electricity ratio
choiceHeatToElectricityRatio=input('Want to set the heat to electricity ratio for cooling tower? 1=yes, 0=no: ');
if choiceHeatToElectricityRatio==1
    kwt_to_kWe=input('This value can be equal to 40 or 80. Set heat to electricity ratio: ');
    fprintf('The heat to electricity ratio is %.1f [-]\n',kwt_to_kWe);
else
    kwt_to_kWe=40;
    fprintf('The heat to electricity ratio is %.1f [-]\n',kwt_to_kWe);
end

%% economic parameter
choiceGeologicalallFormation=input('Want to set the geological formation? 1=yes, 0=no: ');
if choiceGeologicalallFormation==1
    sourcetype=input('Set the type of geological formation. 1=sedimentary, 2=fractured, 3=basement: ');
    if sourcetype==1
        Succrate=0.95;
        fprintf('The geological formation is sedimentary, and the success rate in drilling is %.2f [-]\n',Succrate);
    elseif sourcetype==2
        Succrate=0.75;
        fprintf('The geological formation is fractured, and the success rate in drilling is %.2f [-]\n',Succrate);
    else
        Succrate=0.9;
        fprintf('The geological formation is basement, and the success rate in drilling is %.2f [-]\n',Succrate);
    end
else
    Succrate=0.9;
end

```

```

    fprintf('The geological formation is basement, and the success rate in drilling is %.2f [-]\n',Succrate);
end

choiceFieldType=input('Want to choose the field type? 1=yes, 0=no: ');
if choiceFieldType==1
    fieldtype=input('Set 1 for brownfield and 2 for greenfield: ');
    if fieldtype==1
        fprintf('The fieldtype chosen is brownfield [-]\n');
    else
        fprintf('The fieldtype chosen is greenfield [-]\n');
    end
else
    fieldtype=2;
    fprintf('The fieldtype chosen is greenfield [-]\n');
end

choiceCapacityFactor=input('Want to set the capacity factor? 1=yes, 0=no: ');
if choiceCapacityFactor==1
    CF=input('set the capacity factor (suggested between 0.7 and 0.9): ');
    fprintf('the capacity factor is %.2f [-]\n', CF);
else
    CF=0.82;
    fprintf('the capacity factor is %.2f [-]\n', CF);
end

choiceElectricityPrice=input('Want to set the electricity selling price?: 1=yes, 0=no: ');
if choiceElectricityPrice==1
    eleprice=input('set the electricity selling price [$/kWh]: ');
    fprintf('The electricity selling price is %.3f [$/kWh]\n', eleprice);
else
    eleprice=0.09;
    fprintf('The electricity selling price is %.3f [$/kWh]\n', eleprice);
end

choiceCAPEXincentives=input('Want to consider incentives for CAPEX? 1=yes, 0=no: ');
if choiceCAPEXincentives==1
    choiceCAPEXincentivesValue=input('Want set incentives for CAPEX? 1=yes, 0=no: ');
    if choiceCAPEXincentivesValue==1
        CAPEXincentives=input('set capex incentives as percentage of the capex (all possible incentives
together). Value between 0 and 1: ');
        fprintf('the incentives are the %.2f of CAPEX\n',CAPEXincentives);
    else
        CAPEXincentives=0.2;
        fprintf('the incentives are the %.2f of CAPEX\n',CAPEXincentives);
    end
else
    CAPEXincentives=0;
    fprintf('the incentives are the %.2f of CAPEX\n',CAPEXincentives);
end

choiceOPEXincentives=input('Want to consider incentives for OPEX? 1=yes, 0=no: ');
if choiceOPEXincentives==1
    choiceOPEXincentivesValue=input('Want set incentives for OPEX? 1=yes, 0=no: ');
    if choiceOPEXincentivesValue==1
        OPEXincentives=input('set opex incentives as percentage of the opex (all possible incentives
together). Value between 0 and 1: ');
        fprintf('the incentives are the %.2f of OPEX\n',OPEXincentives);
    else
        OPEXincentives=0.1;
        fprintf('the incentives are the %.2f of OPEX\n',OPEXincentives);
    end
else
    OPEXincentives=0;
    fprintf('the incentives are the %.2f of OPEX\n',OPEXincentives);
end

choiceREVENUEincentives=input('Want to consider incentives for revenue? 1=yes, 0=no: ');
if choiceREVENUEincentives==1
    choiceREVENUEincentivesValue=input('Want set incentives for revenue? 1=yes, 0=no: ');
    if choiceREVENUEincentivesValue==1
        REVENUEincentives=input('set incentives as percentage of the revenue (all possible incentives
together). Value between 0 and 1: ');
        fprintf('the incentives are the %.2f of revenue\n',REVENUEincentives);
    else
        REVENUEincentives=0.3;
    end
end

```

```

        fprintf('the incentives are the %.2f of revenue\n',REVENUEincentives);
    end
else
    REVENUEincentives=0;
    fprintf('the incentives are the %.2f of revenue\n',REVENUEincentives);
end

```

V. Binary

```

year=(1:1/time_step_per_year:lifetime);

p_prod=p_time;
T_prod=T_time;
m_prod=m_time;

p_limit=35;
for i=1:length(year)
    h_prod(i)=XSteam('h_pT',p_prod(i),T_prod(i));
    if p_prod(i)>p_limit
        p_prod(i)=p_limit;
    end
    x_prod(i)=XSteam('x_ph',p_prod(i),h_prod(i));
    s_prod(i)=XSteam('s_ph',p_prod(i),h_prod(i));
    rho_prod(i)=XSteam('rho_ph',p_prod(i),h_prod(i));
    rhoL_prod(i)=XSteam('rhoL_p',p_prod(i));
    rhoV_prod(i)=XSteam('rhoV_p',p_prod(i));
end
if any(x_prod(:)>0)
    W_net=zeros(1,length(year));
    return;
end

% cyclone filter
A_filter=pi*D_filter^2/4;
deltaH=10;
g=9.81;

vol_flow_rate=m_prod./rho_prod;
velocity=vol_flow_rate/A_filter;
deltaP_cs=0.5*rhoV_prod./rhoL_prod.*velocity.^2*(deltaH/g)/1e5; % [bar]

m1=m_prod; %[kg/s] production well flow rate
T1=T_prod; %[°C] production well temperature
p1=p_prod-deltaP_cs; %[bar] production well pressure

%% vector containing all the state parameters(1=pressure,2=Temperature,3=mass flow rate,4=specific heat at
constant pressure,5=density,6=enthalpy,7=entropy,8=vapour mass fraction,9=specific exergy)
state1=zeros(9,lifetime); % inlet HX (geoth.fluid side) after filter unit
state2=zeros(9,lifetime); % outlet HX (geoth.fluid side) and inlet blowdown pump
state3=zeros(9,lifetime); % injection state after blowdown pump
state4=zeros(9,lifetime); % outlet condensation pump and inlet heat exchanger (R600a side)
state5=zeros(9,lifetime); % outlet HX (R600a side) and turbine inlet
state6=zeros(9,lifetime); % turbine outlet and inlet condenser
state7=zeros(9,lifetime); % outlet condenser and inlet condensation pump
state8=zeros(9,lifetime); % inlet condenser (water side) and outlet circulation pump
state9=zeros(9,lifetime); % outlet condenser (water side) and inlet cooling tower
state10=zeros(9,lifetime); % outlet cooling tower and inlet circulation pump

%% enthalpy and entropy at ambient condition
% water
h0=XSteam('h_pT',p0,T0); % [kJ/kg]
s0=XSteam('s_pT',p0,T0); % [kJ/kg/°C]
cp0=XSteam('Cp_pT',p0,T0); % [kJ/kg/°C]

% isobutane
h0_r600a=realprop('h','si','bar','C','r600a','t',T0,'p',p0); % [kJ/kg]
s0_r600a=realprop('s','si','bar','C','r600a','t',T0,'p',p0); % [kJ/kg/°C]

%% state 1: Inlet HX (geothermal fluid side) after filter unit
% evaluation of unknown parameter
for i=1:length(year)
    h1(i)=XSteam('h_pT',p1(i),T1(i)); % [kJ/kg]
    cp1(i)=XSteam('Cp_ph',p1(i),h1(i)); % [kJ/kg/°C]
    rho1(i)=XSteam('rho_ph',p1(i),h1(i)); % [kg/m^3]

```

```

s1(i)=XSteam('s_ph',p1(i),h1(i));    % [kJ/kg/°C]
xvap1(i)=XSteam('x_ph',p1(i),h1(i)); % [%]
e1(i)=(h1(i)-h0)-(T0+273.15)*(s1(i)-s0);

state1(1,i)=p1(i);
state1(2,i)=T1(i);
state1(3,i)=m1(i);
state1(4,i)=cp1(i);
state1(5,i)=rho1(i);
state1(6,i)=h1(i);
state1(7,i)=s1(i);
state1(8,i)=xvap1(i);
state1(9,i)=e1(i);
end

if any(xvap1(:)>0)
    W_net=zeros(1,length(year));
    return;
end

%% state 4: outlet condensation pump and inlet heat exchanger (R600a side)
%% assumption: all the working fluid parameter in the initial state are
%% constant in time
p4=pressure_orc_fluid*ones(1,lifetime); % [bar]
T4=temperature_orc_fluid*ones(1,lifetime); % [°C]
dTpp=5;
Tevap=realprop('t','si','bar','C','r600a','p',p4(1),'x',0);
hVout=realprop('h','si','bar','C','r600a','x',1,'p',p4(1));
hVin=realprop('h','si','bar','C','r600a','x',0,'p',p4(1));
flow_rate_orc_fluid=m1(end)*cp1(end)*(T1(end)-(Tevap+dTpp))/(hVout-hVin);
if flow_rate_orc_fluid>m1(end)
    flow_rate_orc_fluid=m1(end);
    m4=flow_rate_orc_fluid*ones(1,lifetime); % [kg/s]
else
    if flow_rate_orc_fluid>0
        m4=flow_rate_orc_fluid*ones(1,lifetime); % [kg/s]
    else
        W_net=zeros(1,length(year));
        return;
    end
end

if any(T1(:)<Tevap)
    W_net=zeros(1,lifetime);
    return;
end

for i=1:length(year)
    h4(i)=realprop('h','si','bar','C','r600a','t',T4(i),'p',p4(i)); % [kJ/kg]
    cp4(i)=realprop('cp','si','bar','C','r600a','h',h4(i),'p',p4(i)); % [kJ/kg/°C]
    v4(i)=realprop('v','si','bar','C','r600a','h',h4(i),'p',p4(i)); % [m^3/kg]
    rho4(i)=1/v4(i); % [kg/m^3]
    s4(i)=realprop('s','si','bar','C','r600a','h',h4(i),'p',p4(i)); % [kJ/kg/°C]
    xvap4(i)=realprop('x','si','bar','C','r600a','h',h4(i),'p',p4(i)); % [%]
    e4(i)=(h4(i)-h0_r600a)-(T0+273.15)*(s4(i)-s0_r600a);

    state4(1,i)=p4(i);
    state4(2,i)=T4(i);
    state4(3,i)=m4(i);
    state4(4,i)=cp4(i);
    state4(5,i)=rho4(i);
    state4(6,i)=h4(i);
    state4(7,i)=s4(i);
    state4(8,i)=xvap4(i);
    state4(9,i)=e4(i);
end

%% HX geothermal fluid/isobutane (R600a)
%U_hx=500; % [W/m2K] global heat transfer coefficient organic fluid-water
increment=20; % discretization of the heat exchanger

T5=(1:1/time_step_per_year:lifetime)*0;
T2=(1:1/time_step_per_year:lifetime)*0;
dT_lmtd=(1:1/time_step_per_year:lifetime)*0;
A_hx=(1:1/time_step_per_year:lifetime)*0;

```

```

Cmin=min(max(m1.*cp1),max(m4.*cp4)); %[kW/°C]
Cmax=max(min(m1.*cp1),min(m4.*cp4)); %[kW/°C]
Cratio=Cmin/Cmax;
if Cratio<0.25
    eps_hx=0.97;
else if Cratio>=0.25 && Cratio<0.5
    eps_hx=0.95;
else if Cratio>=0.5 && Cratio<0.75
    eps_hx=0.90;
else
    eps_hx=0.80;
end
end
end

hV=realprop('h','si','bar','C','r600a','x',1,'p',p4(1));
Q_evap=m4.*((hV+0.01)-h4);
dQ=Q_evap/increment;
Q_exch=Q_evap/eps_hx;

for tt=1:length(year)

    T1_in(1)=T1(tt);
    h1_in(1)=h1(tt);
    T4_in(1)=T4(tt);
    h4_in(1)=h4(tt);

    Q1(1)=0;
    for i=2:increment+1
        h1_in(i)=h1_in(i-1)-dQ(tt)/eps_hx/m1(tt);
        h4_in(i)=h4_in(i-1)+dQ(tt)/m4(tt);
        T1_in(i)=XSteam('T_ph',p1(1),h1_in(i));
        T4_in(i)=realprop('t','si','bar','C','r600a','h',h4_in(i),'p',p4(tt));
        Q1(i)=Q1(i-1)+dQ(tt)/eps_hx;
    end
    dT_HX=flip(T1_in)-T4_in;

    if any(dT_HX(:)<0)
        W_net=zeros(1,lifetime);
        return;
    end

    T2(tt)=T1_in(end);
    T5(tt)=T4_in(end);
    h5(tt)=h4_in(end);
    h2(tt)=h1_in(end);
    dT_lmtd(tt)=((T1(tt)-T5(tt))-(T2(tt)-T4(tt)))/log((T1(tt)-T5(tt))/(T2(tt)-T4(tt)));
    A_hx(tt)=Q_exch(tt)*1000/U_hx/dT_lmtd(tt);
end

if dT_lmtd(:)<0
    W_net=zeros(1,lifetime);
    return;
end

%% state 2: outlet HX (geoth.fluid side) and inlet blowdown pump
p2=p1;
m2=m1;
for i=1:length(year)
    cp2(i)=XSteam('Cp_ph',p2(i),h2(i)); % [kJ/kg/°C]
    rho2(i)=XSteam('rho_ph',p2(i),h2(i)); % [kg/m^3]
    s2(i)=XSteam('s_ph',p2(i),h2(i)); % [kJ/kg/°C]
    xvap2(i)=XSteam('x_ph',p2(i),h2(i)); % [%]
    e2(i)=(h2(i)-h0)-(T0+273.15)*(s2(i)-s0);

    state2(1,i)=p2(i);
    state2(2,i)=T2(i);
    state2(3,i)=m2(i);
    state2(4,i)=cp2(i);
    state2(5,i)=rho2(i);
    state2(6,i)=h2(i);
    state2(7,i)=s2(i);
    state2(8,i)=xvap2(i);
    state2(9,i)=e2(i);
end

```

```

end

if any(xvap2(:)>0)
    W_net=zeros(1,length(year));
    return;
end

%% state3: injection state after blowdown pump
L_inj=1500;
pressure_injection=rho2(1)*9.81*L_inj/1e5;
p3=ones(1,lifetime)*pressure_injection;
T3=T2;
m3=m2;

% pump power
for i=1:length(year)
    h3(i)=XSteam('h_pT',p3(i),T3(i));
    cp3(i)=XSteam('Cp_ph',p3(i),h3(i)); % [kJ/kg/°C]
    rho3(i)=XSteam('rho_ph',p3(i),h3(i)); % [kg/m^3]
    s3(i)=XSteam('s_ph',p3(i),h3(i)); % [kJ/kg/°C]
    xvap3(i)=XSteam('x_ph',p3(i),h3(i)); % [%]
    e3(i)=(h3(i)-h0)-(T0+273.15)*(s3(i)-s0);

    state3(1,i)=p3(i);
    state3(2,i)=T3(i);
    state3(3,i)=m3(i);
    state3(4,i)=cp3(i);
    state3(5,i)=rho3(i);
    state3(6,i)=h3(i);
    state3(7,i)=s3(i);
    state3(8,i)=xvap3(i);
    state3(9,i)=e3(i);

    % blow down pump power consumption
    W_bdpump(i)=m3(i)/rho3(i)*3600*(p3(i)-p2(i))*10^5/3.6E6/eta_bdpump/1000; %[MW]
end

%% state 5: outlet HX (R600a side) and turbine inlet
p5=p4; % [bar]
m5=m4; % [kg/s]
for i=1:length(year)
    cp5(i)=realprop('cp','si','bar','C','r600a','h',h5(i),'p',p5(i)); % [kJ/kg/°C]
    v5(i)=realprop('v','si','bar','C','r600a','h',h5(i),'p',p5(i)); % [m^3/kg]
    rho5(i)=1/v5(i); % [kg/m^3]
    s5(i)=realprop('s','si','bar','C','r600a','h',h5(i),'p',p5(i)); % [kJ/kg/°C]
    xvap5(i)=realprop('x','si','bar','C','r600a','h',h5(i),'p',p5(i)); % [%]
    e5(i)=(h5(i)-h0_r600a)-(T0+273.15)*(s5(i)-s0_r600a);

    state5(1,i)=p5(i);
    state5(2,i)=T5(i);
    state5(3,i)=m5(i);
    state5(4,i)=cp5(i);
    state5(5,i)=rho5(i);
    state5(6,i)=h5(i);
    state5(7,i)=s5(i);
    state5(8,i)=xvap5(i);
    state5(9,i)=e5(i);
end

%% state 6: turbine outlet and inlet condenser
m6=m5;
p6=ones(1,lifetime)*pressure_out_turbine;

for i=1:length(year)
    state6(3,i)=m6(i);
    state6(1,i)=p6(i);

    % pressure outlet at turbine is known
    h6s(i)=realprop('h','si','bar','C','r600a','s',s5(i),'p',p6(i)); % [kJ/kg]
    h6(i)=h5(i)-eta_turb*(h5(i)-h6s(i)); % [kJ/kg]
    T6(i)=realprop('t','si','bar','C','r600a','h',h6(i),'p',p6(i)); % [°C]
    cp6(i)=realprop('cp','si','bar','C','r600a','h',h6(i),'p',p6(i)); % [kJ/kg/°C]
    v6(i)=realprop('v','si','bar','C','r600a','h',h6(i),'p',p6(i)); % [kg/m^3]
    rho6(i)=1/v6(i);
    s6(i)=realprop('s','si','bar','C','r600a','h',h6(i),'p',p6(i)); % [kJ/kg/°C]
end

```

```

e6(i)=(h6(i)-h0_r600a)-(T0+273.15)*(s6(i)-s0_r600a);

% check if fraction of vapour is high enough to not damage the turbine
xvap6(i)=realprop('x','si','bar','C','r600a','h',h6(i),'p',p6(i));
if xvap6(i)/100<xvap_lim
    W_net=zeros(1,lifetime);
    return; %The fluid could be liquid, or between a liquid and vapour phase. Reduce the pressure p6 to
ensure only vapour phase at the exit
end

state6(2,i)=T6(i);
state6(3,i)=m6(i);
state6(4,i)=cp6(i);
state6(5,i)=rho6(i);
state6(6,i)=h6(i);
state6(7,i)=s6(i);
state6(8,i)=xvap6(i);
state6(9,i)=e6(i);

% evaluation of the turbine's specific work
w_turb(i)=h5(i)-h6(i);
W_turb(i)=m6(i)*w_turb(i)/1000; % [MWe]
W_gen(i)=eta_gen*W_turb(i); % [MWe]
end

%% Condenser isobutane(R600a)/water
%U_cond=500; %[W/m2K] global heat transfer coefficient organic fluid-water
p8=pressure_water_inlet_cond*ones(1,lifetime);
T8=T4-dT_cooling_water;
h8=XSteam('h_pT',p8(1),T8(1))*ones(1,lifetime);
cp8=XSteam('cp_ph',p8(1),h8(1))*ones(1,lifetime);

eps_cond=0.99;
hcond=realprop('h','si','bar','C','r600a','t',T4(1),'p',p6(1))*ones(1,lifetime);
Q_cond=m6.*(h6-hcond);
dQ_cond=Q_cond/increment;
m8=Q_cond./(cp8.*dT_cooling_water);

for tt=1:length(year)
    T6_in(1)=T6(tt);
    h6_in(1)=h6(tt);
    T8_in(1)=T8(tt);
    h8_in(1)=h8(tt);

    Q2(1)=0;
    for i=2:increment+1
        h6_in(i)=h6_in(i-1)-dQ_cond(tt)/m6(tt);
        T6_in(i)=realprop('t','si','bar','C','r600a','h',h6_in(i),'p',p6(1));
        h8_in(i)=h8_in(i-1)+dQ_cond(tt)*eps_cond/m8(tt);
        T8_in(i)=XSteam('T_ph',p8(tt),h8_in(i));
        Q2(i)=Q2(i-1)+dQ_cond(tt);
    end

    dT_cond=flip(T6_in)-T8_in;
    if any(dT_cond(:)<0)
        W_net=zeros(1,length(year));
        return;
    end
    T7(tt)=T6_in(end);
    T9(tt)=T8_in(end);
    h7(tt)=h6_in(end);
    h9(tt)=h8_in(end);
    dT_lmtcond_cond(tt)=((T6(tt)-T9(tt))-(T7(tt)-T8(tt)))/log((T6(tt)-T9(tt))/(T7(tt)-T8(tt)));
    A_cond(tt)=Q_cond(tt)*1000/U_cond/dT_lmtcond_cond(tt);
end

%% state 7: outlet condenser and inlet condensation pump
p7=p6;
m7=m6;
for i=1:length(year)
    cp7(i)=realprop('cp','si','bar','C','r600a','h',h7(i),'p',p7(i)); % [kJ/kg/°C]
    v7(i)=realprop('v','si','bar','C','r600a','h',h7(i),'p',p7(i)); % [m^3/kg]
    rho7(i)=1/v7(i); % [kg/m^3]
    s7(i)=realprop('s','si','bar','C','r600a','h',h7(i),'p',p7(i)); % [kJ/kg/°C]
    xvap7(i)=realprop('x','si','bar','C','r600a','h',h7(i),'p',p7(i));

```

```

e7(i)=(h7(i)-h0_r600a)-(T0+273.15)*(s7(i)-s0_r600a); % [kJ/kg]

state7(1,i)=p7(i);
state7(2,i)=T7(i);
state7(3,i)=m7(i);
state7(4,i)=cp7(i);
state7(5,i)=rho7(i);
state7(6,i)=h7(i);
state7(7,i)=s7(i);
state7(8,i)=xvap7(i);
state7(9,i)=e7(i);

% condensation pump power consumption
W_condpump(i)=m7(i)/rho7(i)*3600*(p4(i)-p7(i))*10^5/3.6E6/eta_condpump/1000; % [MWe]
End

%% state 8: inlet condenser (water side) and outlet circulation pump
for i=1:length(year)
rho8(i)=XSteam('rho_ph',p8(i),h8(i));
s8(i)=XSteam('s_ph',p8(i),h8(i));
xvap8(i)=XSteam('x_ph',p8(i),h8(i));
e8(i)=(h9(i)-h0)-(T0+273.15)*(s8(i)-s0);

state8(1,i)=p8(i);
state8(2,i)=T8(i);
state8(3,i)=m8(i);
state8(4,i)=cp8(i);
state8(5,i)=rho8(i);
state8(6,i)=h8(i);
state8(7,i)=s8(i);
state8(8,i)=xvap8(i);
state8(9,i)=e8(i);
end

%% state 9: outlet condenser (water side) and inlet cooling tower
p9=p8;
m9=m8;
for i=1:length(year)
cp9(i)=XSteam('cp_ph',p9(i),h9(i));
rho9(i)=XSteam('rho_ph',p9(i),h9(i));
s9(i)=XSteam('s_ph',p9(i),h9(i));
xvap9(i)=XSteam('x_ph',p9(i),h9(i));
e9(i)=(h9(i)-h0)-(T0+273.15)*(s9(i)-s0);

state9(1,i)=p9(i);
state9(2,i)=T9(i);
state9(3,i)=m9(i);
state9(4,i)=cp9(i);
state9(5,i)=rho9(i);
state9(6,i)=h9(i);
state9(7,i)=s9(i);
state9(8,i)=xvap9(i);
state9(9,i)=e9(i);
end

%% state 10: outlet cooling tower and inlet circulation pump
p10=p9-0.5; % assumption of concentrated pressure loss(?)
T10=T8;
m10=m9;
for i=1:length(year)
h10(i)=XSteam('h_pT',p10(i),T10(i));
cp10(i)=XSteam('Cp_ph',p10(i),h10(i));
rho10(i)=XSteam('rho_ph',p10(i),h10(i));
s10(i)=XSteam('s_ph',p10(i),h10(i));
xvap10(i)=XSteam('x_ph',p10(i),h10(i));
e10(i)=(h10(i)-h0)-(T0+273.15)*(s10(i)-s0);

state10(1,i)=p10(i);
state10(2,i)=T10(i);
state10(3,i)=m10(i);
state10(4,i)=cp10(i);
state10(5,i)=rho10(i);
state10(6,i)=h10(i);
state10(7,i)=s10(i);
state10(8,i)=xvap10(i);

```

```

state10(9,i)=e10(i);

% cooling tower
q_cooling_tow(i)=h9(i)-h10(i); %[kJ/kg]
Qcool_tow(i)=q_cooling_tow(i)*m10(i)/1000; %[MWth]
W_tower_fan(i)=Qcool_tow(i)*1000/kWt_to_kWe; %[kW]

% circulation pump
W_circumpump(i)=m10(i)/rho10(i)*3600*(p8(i)-p10(i))*10^5/3.6E6/eta_circumpump/1000;    %[MW]
end

%% result section
for i=1:lifetime
    W_net(i)=W_gen(i)-W_circumpump(i)-W_bdpump(i)-W_condpump(i)-W_tower_fan(i)/1000; %[MW]
    SFC(i)=m5(i)*3600/(W_net(i)*1000); %[kg/kWh]
    eta_th(i)=W_net(i)*1000/Q_exch(i);
end

if any(W_net(:)<0)
    W_net=zeros(1,lifetime);
end

```

VI. *Dry_steam_pure_electricity*

```

year=(1:time_step_per_year:lifetime);

p_prod=p_time;
T_prod=T_time;
m_prod=m_time;

p_limit=35;

for i=1:length(year)
    h_prod(i)=XSteam('h_pT',p_prod(i),T_prod(i));
    if p_prod(i)>p_limit
        p_prod(i)=p_limit;
    end
    s_prod(i)=XSteam('s_ph',p_prod(i),h_prod(i));
    rho_prod(i)=XSteam('rho_ph',p_prod(i),h_prod(i));
    rhoL_prod(i)=XSteam('rhoL_p',p_prod(i));
    rhoV_prod(i)=XSteam('rhoV_p',p_prod(i));
    x_prod(i)=XSteam('x_ph',p_prod(i),h_prod(i));
end

% cyclone filter
A_filter=pi*D_filter^2/4;
deltaH=10;
g=9.81;

vol_flow_rate=m_prod./rho_prod;
velocity=vol_flow_rate/A_filter;
deltaP_cs=0.5*rhoV_prod./rhoL_prod.*velocity.^2*(deltaH/g)*98.0665/1e5; % [bar]

m1=m_prod; %[kg/s] production well flow rate
T1=T_prod; %[°C] production well temperature
p1=p_prod-deltaP_cs; %[bar] production well pressure

%% vector containing all the state parameters (1=pressure,2=Temperature,3=mass flow rate,4=specific heat at
constant pressure,5=density,6=enthalpy,7=entropy,8=vapour mass fraction,9=specific exergy)
state1=zeros(9,lifetime); % turbine inlet
state2=zeros(9,lifetime); % turbine outlet and inlet condenser
state3=zeros(9,lifetime); % outlet condenser and inlet condensation pump
state4=zeros(9,lifetime); % outlet condensation pump and inlet cooling tower
state5=zeros(9,lifetime); % outlet cooling tower
state6=zeros(9,lifetime); % injection state after blowdown pump

%% enthalpy and entropy at ambient condition
h0=XSteam('h_pT',p0,T0);    %[kJ/kg]
s0=XSteam('s_ph',p0,h0);    %[kJ/kg/°C]
cp0=XSteam('Cp_ph',p0,h0);  %[kJ/kg/°C]

%% state 1: Production state and turbine inlet
% evaluation of unknown parameter
for i=1:length(year)

```

```

h1(i)=XSteam('h_pT',p1(i),T1(i));           %[kJ/kg]
cp1(i)=XSteam('Cp_ph',p1(i),h1(i));         %[kJ/kg/°C]
rho1(i)=XSteam('rho_ph',p1(i),h1(i));       %[kg/m^3]
s1(i)=XSteam('s_ph',p1(i),h1(i));           %[kJ/kg/°C]
xvap1(i)=XSteam('x_ph',p1(i),h1(i));        [%]
e1(i)=(h1(i)-h0)-(T0+273.15)*(s1(i)-s0);    %[kJ/kg]

state1(1,i)=p1(i);
state1(2,i)=T1(i);
state1(3,i)=m1(i);
state1(4,i)=cp1(i);
state1(5,i)=rho1(i);
state1(6,i)=h1(i);
state1(7,i)=s1(i);
state1(8,i)=xvap1(i);
state1(9,i)=e1(i);
end

if xvap1(:)<0.95
    W_net=zeros(1,lifetime);
    return
end

%% state 2: Turbine outlet and condenser inlet
m2=m1;
p2=ones(1,lifetime)*pressure_turb_outlet;

for i=1:length(year)

    state2(1,i)=p2(i);
    state2(3,i)=m2(i);
    p2(i)=state2(1,i);
    h2s(i)=XSteam('h_ps',p2(i),s1(i));       %[kJ/kg]
    h2(i)=h1(i)-eta_turb*(h1(i)-h2s(i));     %[kJ/kg]
    T2(i)=XSteam('T_ph',p2(i),h2(i));        %[°C]
    cp2(i)=XSteam('Cp_ph',p2(i),h2(i));     %[kJ/kg/°C]
    rho2(i)=XSteam('rho_ph',p2(i),h2(i));    %[kg/m^3]
    s2(i)=XSteam('s_ph',p2(i),h2(i));       %[kJ/kg/°C]
    e2(i)=(h2(i)-h0)-(T0+273.15)*(s2(i)-s0);

    xvap2(i)=XSteam('x_ph',p2(i),h2(i));
    if xvap2(i)<xvap_lim
        W_net=zeros(1,lifetime);
        return;
    end

    if isnan(cp2(i))
        cp2l(i)=XSteam('CpL_T',T2(i));
        cp2v(i)=XSteam('CpV_T',T2(i));
        cp2(i)=cp2l(i)*(1-xvap2(i))+cp2v(i)*xvap2(i);
    end

    state2(2,i)=T2(i);
    state2(4,i)=cp2(i);
    state2(5,i)=rho2(i);
    state2(6,i)=h2(i);
    state2(7,i)=s2(i);
    state2(8,i)=xvap2(i);
    state2(9,i)=e2(i);

    % evaluation of the turbine's specific work
    w_turb(i)=h1(i)-h2(i);
    W_turb(i)=w_turb(i)*m1(i)/1000; %[MWe]
    W_gen(i)=eta_gen*W_turb(i); %[MWe]
end

%% state 3: Outlet condenser and inlet condensation pump
% evaluation of condensation heat. The steam is totally condensate
p3=p2;
for i=1:length(year)
    h3(i)=XSteam('hL_T',T2(i));
    qcond(i)=h2(i)-h3(i)*(1-x_ncg);          %[kJ/kg]
    Qcond(i)=m2(i)*qcond(i)/1000;           %[MWth]
    % evaluation of sprayed flow rate
    T5(i)=Tinj;
end

```

```

if Tinj>T2(i)
    W_net=zeros(1,lifetime);
    return;
end
h5(i)=XSteam('hL_T',T5(i));
mspray(i)=m2(i)*qcond(i)/(h3(i)-h5(i));

% set state3 parameter
T3(i)=XSteam('Tsat_p',p3(i));
m3(i)=(1-x_ncg)*m2(i)+mspray(i);
cp3(i)=XSteam('Cp_ph',p3(i),h3(i));
rho3(i)=XSteam('rho_ph',p3(i),h3(i));
s3(i)=XSteam('s_ph',p3(i),h3(i));
xvap3(i)=XSteam('x_ph',p3(i),h3(i));
e3(i)=(h3(i)-h0)-(T0+273.15)*(s3(i)-s0);

state3(1,i)=p3(i);
state3(2,i)=T3(i);
state3(3,i)=m3(i);
state3(4,i)=cp3(i);
state3(5,i)=rho3(i);
state3(6,i)=h3(i);
state3(7,i)=s3(i);
state3(8,i)=xvap3(i);
state3(9,i)=e3(i);

% Power of compressor to remove uncondensed gas (CO2+H2S+Hg)
m_ncg(i)=m2(i)*x_ncg;
Runiversal=8.314; % [J/molK]
MM_Hg=200.59; % [g/mol]
MM_CO2=44.01; % [g/mol]
MM_H2S=34.0818; % [g/mol]
R_Hg=Runiversal/xHg*1000; % [J/gK]=*1000[J/kgK]
R_CO2=Runiversal/MM_CO2*1000; % [J/gK]=*1000[J/kgK]
R_H2S=Runiversal/MM_H2S*1000; % [J/gK]=*1000[J/kgK]
R_ncg=R_Hg*xHg+R_H2S*xH2S+R_CO2*xCO2;
% Ideal gas law assumption
rho_Hg=p3(i)*1e5/R_Hg/(T3(i)+273.15); %[kg/m3]
rho_CO2=p3(i)*1e5/R_CO2/(T3(i)+273.15); %[kg/m3]
rho_H2S=p3(i)*1e5/R_H2S/(T3(i)+273.15); %[kg/m3]
cv_co2=0.658; %kJ/kgK
cv_h2s=0.76;
cv_hg=1.0145;
cv_ncg=cv_co2*xCO2+cv_h2s*xH2S+cv_hg*xHg;

cp_co2=0.849; %kJ/kgK
cp_h2s=1.01;
cp_hg=0.140;
cp_ncg=cp_co2*xCO2+cp_h2s*xH2S+cp_hg*xHg;
n_ncg=cp_ncg/cv_ncg;

p_compr=pressure_compr_outlet;
if isnan(n_ncg)
    W_compr(i)=0;
else
    W_compr(i)=m_ncg(i)*R_ncg*(T2(i)+273.15)*n_ncg/(n_ncg-1)/eta_compr*((p_compr/p2(i))^((n_ncg-1)/n_ncg)-1)/10^6; % [MW]
end

end

%% state 4: outlet condensation pump and inlet cooling tower
p4=p3+1.5;
m4=m3;
for i=1:lifetime
    h4s(i)=XSteam('h_ps',p4(i),s3(i));
    h4(i)=h3(i)-(h3(i)-h4s(i))/eta_condpump;
    T4(i)=XSteam('T_ph',p4(i),h4(i));
    cp4(i)=XSteam('cp_ph',p4(i),h4(i));
    rho4(i)=XSteam('rho_ph',p4(i),h4(i));
    s4(i)=XSteam('s_ph',p4(i),h4(i));
    xvap4(i)=XSteam('x_ph',p4(i),h4(i));
    e4(i)=(h4(i)-h0)-(T0+273.15)*(s4(i)-s0);

    state4(1,i)=p4(i);

```

```

state4(2,i)=T4(i);
state4(3,i)=m4(i);
state4(4,i)=cp4(i);
state4(5,i)=rho4(i);
state4(6,i)=h4(i);
state4(7,i)=s4(i);
state4(8,i)=xvap4(i);
state4(9,i)=e4(i);

% Condensation pump power consumption
W_condpump(i)=m4(i)/rho4(i)*3600*(p4(i)-p3(i))*10^5/eta_condpump/3.6E6/1000;    %[MW]
end

%% state 5: outlet cooling tower
% here the temperature is reduced up to the injection temperature
p5=p4;
m5=m4;
for i=1:lifetime
    T5(i)=Tinj;
    if T4(i)<T5(i) %The injection temperature is higher than the temperature entering in the cooling tower
        W_net=zeros(1,lifetime);
        return
    end

    h5(i)=XSteam('h_pT',p5(i),T5(i));
    cp5(i)=XSteam('Cp_ph',p5(i),h5(i));
    rho5(i)=XSteam('rho_ph',p5(i),h5(i));
    s5(i)=XSteam('s_ph',p5(i),h5(i));
    xvap5(i)=XSteam('x_ph',p5(i),h5(i));
    e5(i)=(h5(i)-h0)-(T0+273.15)*(s5(i)-s0);

    state5(1,i)=p5(i);
    state5(2,i)=T5(i);
    state5(3,i)=m5(i);
    state5(4,i)=cp5(i);
    state5(5,i)=rho5(i);
    state5(6,i)=h5(i);
    state5(7,i)=s5(i);
    state5(8,i)=xvap5(i);
    state5(9,i)=e5(i);

    % cooling tower, fan power evaluation
    q_cooling_tow(i)=h4(i)-h5(i); %[kJ/kg]
    Qcool_tow(i)=q_cooling_tow(i)*m5(i)/1000; %[MWth]

    W_tower_fan(i)=Qcool_tow(i)*1000/kWt_to_kWe; %[kW]
end

%% state 6: Injection state, after blowdown pump
L_inj=L_prod/2;
pressure_injection=rho5(1)*9.81*L_inj/1e5;
p6=ones(1,lifetime)*pressure_injection;
T6=T5;
m6=m5-mspray;
for i=1:lifetime
    h6(i)=XSteam('h_pT',p6(i),T6(i));
    cp6(i)=XSteam('Cp_ph',p6(i),h6(i));
    rho6(i)=XSteam('rho_ph',p6(i),h6(i));
    s6(i)=XSteam('s_ph',p6(i),h6(i));
    xvap6(i)=XSteam('x_ph',p6(i),h6(i));
    e6(i)=(h6(i)-h0)-(T0+273.15)*(s6(i)-s0);

    state6(1,i)=p6(i);
    state6(2,i)=T6(i);
    state6(3,i)=m6(i);
    state6(4,i)=cp6(i);
    state6(5,i)=rho6(i);
    state6(6,i)=h6(i);
    state6(7,i)=s6(i);
    state6(8,i)=xvap6(i);
    state6(9,i)=e6(i);

    % Blow down pump power consumption
    W_bdpump(i)=m6(i)/rho6(i)*3600*(p6(i)-p5(i))*10^5/eta_bdpump/3.6E6/1000;    %[MW]
end

```

```

%% circulation pump power
for i=1:lifetime
    W_circumpump(i)=mspray(i)/rho5(i)*3600*(p5(i)-p3(i))*10^5/eta_circumpump/3.6E6/1000;    %[MW]
end

%% result section
W_net=zeros(1,lifetime);
for i=1:lifetime
    W_net(i)=W_gen(i)-W_circumpump(i)-W_bdpump(i)-W_condpump(i)-W_tower_fan(i)/1000-W_compr(i); %[MW]
    SSC(i)=m1(i)*3600/(W_net(i)*1000); %[kg/kWh]
    eta_util(i)=W_net(i)*1000/(m1(i)*e1(i));
end

```

VII. Dry_steam_top_spillation

```

year=(1:1/time_step_per_year:lifetime);

p_prod=p_time;
T_prod=T_time;
m_prod=m_time;

p_limit=35;

for i=1:length(year)
    h_prod(i)=XSteam('h_pT',p_prod(i),T_prod(i));
    if p_prod(i)>p_limit
        p_prod(i)=p_limit;
    end
    s_prod(i)=XSteam('s_ph',p_prod(i),h_prod(i));
    rho_prod(i)=XSteam('rho_ph',p_prod(i),h_prod(i));
    rhoL_prod(i)=XSteam('rhoL_p',p_prod(i));
    rhoV_prod(i)=XSteam('rhoV_p',p_prod(i));
end

% cyclone filter
A_filter=pi*D_filter^2/4;
deltaH=10;
g=9.81;

vol_flow_rate=m_prod./rho_prod;
velocity=vol_flow_rate/A_filter;
deltaP_cs=0.5*rhoV_prod./rhoL_prod.*velocity.^2*(deltaH/g)/1e5; % [bar]

m_enduse=x_enduse*m_prod;

m1=m_prod-m_enduse; %[kg/s] production well flow rate
T1=T_prod; %[°C] production well temperature
p1=p_prod-deltaP_cs; %[bar] production well pressure

%% vector containing all the state parameters (1=pressure,2=Temperature,3=mass flow rate,4=specific heat at
constant pressure,5=density,6=enthalpy,7=entropy,8=vapour mass fraction,9=specific exergy)
state1=zeros(9,lifetime); % turbine inlet
state2=zeros(9,lifetime); % turbine outlet and inlet condenser
state3=zeros(9,lifetime); % outlet condenser and inlet condensation pump
state4=zeros(9,lifetime); % outlet condensation pump and inlet cooling tower
state5=zeros(9,lifetime); % outlet cooling tower
state6=zeros(9,lifetime); % injection state after blowdown pump

%% enthalpy and entropy at ambient condition
h0=XSteam('h_pT',p0,T0);    %[kJ/kg]
s0=XSteam('s_pT',p0,T0);    %[kJ/kg/°C]
cp0=XSteam('Cp_pT',p0,T0);  %[kJ/kg/°C]

%% state 1: Production state and turbine inlet
% evaluation of unknown parameter
for i=1:length(year)
    h1(i)=XSteam('h_pT',p1(i),T1(i));    %[kJ/kg]
    cp1(i)=XSteam('Cp_ph',p1(i),h1(i));  %[kJ/kg/°C]
    rho1(i)=XSteam('rho_ph',p1(i),h1(i)); %[kg/m^3]
    s1(i)=XSteam('s_ph',p1(i),h1(i));    %[kJ/kg/°C]
    xvap1(i)=XSteam('x_ph',p1(i),h1(i));  [%]
    e1(i)=(h1(i)-h0)-(T0+273.15)*(s1(i)-s0); %[kJ/kg]
end

```

```

state1(1,i)=p1(i);
state1(2,i)=T1(i);
state1(3,i)=m1(i);
state1(4,i)=cp1(i);
state1(5,i)=rho1(i);
state1(6,i)=h1(i);
state1(7,i)=s1(i);
state1(8,i)=xvap1(i);
state1(9,i)=e1(i);
end

%% state 2: Turbine outlet and condenser inlet
m2=m1;
p2=ones(1,lifetime)*pressure_turb_outlet;

if xvap1(:)<0.95
    W_net=zeros(1,lifetime);
    return
end

for i=1:length(year)

    state2(1,i)=p2(i);
    state2(3,i)=m2(i);
    p2(i)=state2(1,i);
    h2s(i)=XSteam('h_ps',p2(i),s1(i));    %[kJ/kg]
    h2(i)=h1(i)-eta_turb*(h1(i)-h2s(i));  %[kJ/kg]
    T2(i)=XSteam('T_ph',p2(i),h2(i));    %[°C]
    cp2(i)=XSteam('Cp_ph',p2(i),h2(i));  %[kJ/kg/°C]
    rho2(i)=XSteam('rho_ph',p2(i),h2(i)); %[kg/m^3]
    s2(i)=XSteam('s_ph',p2(i),h2(i));    %[kJ/kg/°C]
    e2(i)=(h2(i)-h0)-(T0+273.15)*(s2(i)-s0);

    xvap2(i)=XSteam('x_ph',p2(i),h2(i));
    if xvap2(i)<xvap_lim
        W_net=zeros(1,lifetime);
        return;
    end

    if isnan(cp2(i))
        cp2l(i)=XSteam('CpL_T',T2(i));
        cp2v(i)=XSteam('CpV_T',T2(i));
        cp2(i)=cp2l(i)*(1-xvap2(i))+cp2v(i)*xvap2(i);
    end

    state2(2,i)=T2(i);
    state2(4,i)=cp2(i);
    state2(5,i)=rho2(i);
    state2(6,i)=h2(i);
    state2(7,i)=s2(i);
    state2(8,i)=xvap2(i);
    state2(9,i)=e2(i);

    % evaluation of the turbine's specific work
    w_turb(i)=h1(i)-h2(i);
    W_turb(i)=w_turb(i)*m1(i)/1000;    %[MWe]
    W_gen(i)=eta_gen*W_turb(i);    %[MWe]
end

%% state 3: Outlet condenser and inlet condensation pump
% evaluation of condensation heat. The steam is totally condensate
p3=p2;
for i=1:length(year)
    h3(i)=XSteam('hL_T',T2(i));
    qcond(i)=h2(i)-h3(i)*(1-x_ncg);    %[kJ/kg]
    Qcond(i)=m2(i)*qcond(i)/1000;    %[MWth]
    % evaluation of sprayed flow rate
    T5(i)=Tinj;
    if Tinj>T2(i)
        W_net=zeros(1,lifetime);
        return;
    end
    h5(i)=XSteam('hL_T',T5(i));
    mspray(i)=m2(i)*qcond(i)/(h3(i)-h5(i));
end

```

```

% set state3 parameter
T3(i)=XSteam('Tsat_p',p3(i));
m3(i)=(1-x_ncg)*m2(i)+mspray(i);
cp3(i)=XSteam('Cp_ph',p3(i),h3(i));
rho3(i)=XSteam('rho_ph',p3(i),h3(i));
s3(i)=XSteam('s_ph',p3(i),h3(i));
xvap3(i)=XSteam('x_ph',p3(i),h3(i));
e3(i)=(h3(i)-h0)-(T0+273.15)*(s3(i)-s0);

state3(1,i)=p3(i);
state3(2,i)=T3(i);
state3(3,i)=m3(i);
state3(4,i)=cp3(i);
state3(5,i)=rho3(i);
state3(6,i)=h3(i);
state3(7,i)=s3(i);
state3(8,i)=xvap3(i);
state3(9,i)=e3(i);

% Power of compressor to remove uncondensed gas (CO2+H2S+Hg)
m_ncg(i)=m2(i)*x_ncg;
Runiversal=8.314; % [J/molK]
MM_Hg=200.59; % [g/mol]
MM_CO2=44.01; % [g/mol]
MM_H2S=34.0818; % [g/mol]
R_Hg=Runiversal/xHg*1000; % [J/gK]=*1000[J/kgK]
R_CO2=Runiversal/MM_CO2*1000; % [J/gK]=*1000[J/kgK]
R_H2S=Runiversal/MM_H2S*1000; % [J/gK]=*1000[J/kgK]
R_ncg=R_Hg*xHg+R_H2S*xH2S+R_CO2*xCO2;
% Ideal gas law assumption
rho_Hg=p3(i)*1e5/R_Hg/(T3(i)+273.15); %[kg/m3]
rho_CO2=p3(i)*1e5/R_CO2/(T3(i)+273.15); %[kg/m3]
rho_H2S=p3(i)*1e5/R_H2S/(T3(i)+273.15); %[kg/m3]
cv_co2=0.658; %kJ/kgK
cv_h2s=0.76;
cv_hg=1.0145;
cv_ncg=cv_co2*xCO2+cv_h2s*xH2S+cv_hg*xHg;

cp_co2=0.849; %kJ/kgK
cp_h2s=1.01;
cp_hg=0.140;
cp_ncg=cp_co2*xCO2+cp_h2s*xH2S+cp_hg*xHg;
n_ncg=cp_ncg/cv_ncg;

p_compr=pressure_compr_outlet;
if isnan(n_ncg)
    w_compr(i)=0;
else
    w_compr(i)=m_ncg(i)*R_ncg*(T2(i)+273.15)*n_ncg/(n_ncg-1)/eta_compr*((p_compr/p2(i))^(n_ncg-1)/n_ncg-1)/10^6; % [MW]
end

end

%% state 4: outlet condensation pump and inlet cooling tower
% need to know the pressure output to evaluate the pump power
p4=p3+1.5;
m4=m3;
for i=1:lifetime
    h4s(i)=XSteam('h_ps',p4(i),s3(i));
    h4(i)=h3(i)-(h3(i)-h4s(i))/eta_condpump;
    T4(i)=XSteam('T_ph',p4(i),h4(i));
    cp4(i)=XSteam('cp_pT',p4(i),T4(i));
    rho4(i)=XSteam('rho_pT',p4(i),T4(i));
    s4(i)=XSteam('s_pT',p4(i),T4(i));
    xvap4(i)=XSteam('x_ph',p4(i),h4(i));
    e4(i)=(h4(i)-h0)-(T0+273.15)*(s4(i)-s0);

    state4(1,i)=p4(i);
    state4(2,i)=T4(i);
    state4(3,i)=m4(i);
    state4(4,i)=cp4(i);
    state4(5,i)=rho4(i);
    state4(6,i)=h4(i);
    state4(7,i)=s4(i);

```

```

state4(8,i)=xvap4(i);
state4(9,i)=e4(i);

% Condensation pump power consumption
W_condpump(i)=m4(i)/rho4(i)*3600*(p4(i)-p3(i))*10^5/eta_condpump/3.6E6/1000;    %[MW]
end

%% state 5: outlet cooling tower
% here the temperature is reduced up to the injection temperature
p5=p4;
m5=m4;
for i=1:lifetime
    T5(i)=Tinj;
    if T4(i)<T5(i)
        error('The injection temperature is higher than the temperature entering in the cooling tower')
    end

    cp5(i)=XSteam('Cp_pT',p5(i),T5(i));
    rho5(i)=XSteam('rho_pT',p5(i),T5(i));
    h5(i)=XSteam('h_pT',p5(i),T5(i));
    s5(i)=XSteam('s_pT',p5(i),T5(i));
    xvap5(i)=XSteam('x_ph',p5(i),h5(i));
    e5(i)=(h5(i)-h0)-(T0+273.15)*(s5(i)-s0);

    state5(1,i)=p5(i);
    state5(2,i)=T5(i);
    state5(3,i)=m5(i);
    state5(4,i)=cp5(i);
    state5(5,i)=rho5(i);
    state5(6,i)=h5(i);
    state5(7,i)=s5(i);
    state5(8,i)=xvap5(i);
    state5(9,i)=e5(i);

    % cooling tower, fan power evaluation
    q_cooling_tow(i)=h4(i)-h5(i); %[kJ/kg]
    Qcool_tow(i)=q_cooling_tow(i)*m5(i)/1000; %[MWth]

    W_tower_fan(i)=Qcool_tow(i)*1000/kwt_to_kWe; %[kW]
end

%% state 6: Injection state, after blowdown pump
pressure_injection=5;
p6=ones(1,lifetime)*pressure_injection;
T6=T5;
m6=m5-mspray;
for i=1:lifetime
    cp6(i)=XSteam('Cp_pT',p6(i),T6(i));
    rho6(i)=XSteam('rho_pT',p6(i),T6(i));
    h6(i)=XSteam('h_pT',p6(i),T6(i));
    s6(i)=XSteam('s_pT',p6(i),T6(i));
    xvap6(i)=XSteam('x_ph',p6(i),h6(i));
    e6(i)=(h6(i)-h0)-(T0+273.15)*(s6(i)-s0);

    state6(1,i)=p6(i);
    state6(2,i)=T6(i);
    state6(3,i)=m6(i);
    state6(4,i)=cp6(i);
    state6(5,i)=rho6(i);
    state6(6,i)=h6(i);
    state6(7,i)=s6(i);
    state6(8,i)=xvap6(i);
    state6(9,i)=e6(i);

    % Blow down pump power consumption
    W_bdpump(i)=m6(i)/rho6(i)*3600*(p6(i)-p5(i))*10^5/eta_bdpump/3.6E6/1000;    %[MW]
end

%% circulation pump power
for i=1:lifetime
    W_circpump(i)=mspray(i)/rho5(i)*3600*(p5(i)-p3(i))*10^5/eta_circpump/3.6E6/1000;    %[MW]
end

%% result section
W_net=zeros(1,lifetime);

```

```

for i=1:lifetime
    W_net(i)=W_gen(i)-W_circpump(i)-W_bdpump(i)-W_condpump(i)-W_tower_fan(i)/1000-W_compr(i); %[MW]
    SSC(i)=m1(i)*3600/(W_net(i)*1000); %[kg/kWh]
    eta_util(i)=W_net(i)*1000/(m1(i)*e1(i));
end

```

VIII. Single flash

```

year=(1:1/time_step_per_year:lifetime);

p_prod=p_time;
T_prod=T_time;
m_prod=m_time;

p_limit=35;

for i=1:length(year)
    h_prod(i)=XSteam('h_pT',p_prod(i),T_prod(i));
    if p_prod(i)>p_limit
        p_prod(i)=p_limit;
    end
    s_prod(i)=XSteam('s_ph',p_prod(i),h_prod(i));
    rho_prod(i)=XSteam('rho_ph',p_prod(i),h_prod(i));
    x_prod(i)=XSteam('x_ph',p_prod(i),h_prod(i));
    rhoL_prod(i)=XSteam('rhoL_p',p_prod(i));
    rhoV_prod(i)=XSteam('rhoV_p',p_prod(i));
    if isnan(rho_prod(i))
        rho_prod(i)=x_prod(i)*rhoV_prod(i)+(1-x_prod(i))*rhoL_prod(i);
    end
end

% cyclone filter
A_filter=pi*D_filter^2/4;
deltaH=10;
g=9.81;

vol_flow_rate=m_prod./rho_prod;
velocity=vol_flow_rate/A_filter;
deltaP_cs=0.5*rhoV_prod./rhoL_prod.*velocity.^2*(deltaH/g)/1e5; % [bar]

% state after cyclone separator, the one which will enter the separator
m1=m_prod; %[kg/s] production well flow rate
T1=T_prod; %[°C] production well temperature
p1=p_prod-deltaP_cs; %[bar] production well pressure

%% vector containing all the state parameters(1=pressure,2=Temperature,3=mass flow rate,4=specific heat at
constant pressure,5=density,6=enthalpy,7=entropy,8=vapour mass fraction,9=specific exergy)
state1=zeros(9,lifetime); % production well/separator inlet
state2=zeros(9,lifetime); % expansion at constant enthalpy
state3=zeros(9,lifetime); % saturated vapour from flash process/inlet turbine
state4=zeros(9,lifetime); % saturated liquid from flash process/first injection state
state5=zeros(9,lifetime); % outlet turbine/inlet condenser
state6=zeros(9,lifetime); % outlet condenser/inlet condensation pump
state7=zeros(9,lifetime); % outlet condensation pump/inlet cooling tower
state8=zeros(9,lifetime); % outlet cooling tower/inlet blowdown pump and inlet circulation pump (split of
mass flow)
state9=zeros(9,lifetime); % outlet blowdown pump and second injection
state10=zeros(9,lifetime); % outlet circulation pump

%% enthalpy and entropy at ambient condition
h0=XSteam('h_pT',p0,T0); %[kJ/kg]
s0=XSteam('s_pT',p0,T0); %[kJ/kg/°C]
cp0=XSteam('Cp_pT',p0,T0); %[kJ/kg/°C]

%% state 1: Production state and inlet separator
% evaluation of unknown parameter
for i=1:length(year)
    h1(i)=XSteam('h_px',p1(i),x_prod(i)); %[kJ/kg]
    cp1(i)=XSteam('Cp_ph',p1(i),h1(i)); %[kJ/kg/°C]
    rho1(i)=XSteam('rho_ph',p1(i),h1(i)); %[kg/m^3]
    s1(i)=XSteam('s_ph',p1(i),h1(i)); %[kJ/kg/°C]
    xvap1(i)=XSteam('x_ph',p1(i),h1(i)); [%]
    e1(i)=(h1(i)-h0)-(T0+273.15)*(s1(i)-s0);
end

```

```

state1(1,i)=p1(i);
state1(2,i)=T1(i);
state1(3,i)=m1(i);
state1(4,i)=cp1(i);
state1(5,i)=rho1(i);
state1(6,i)=h1(i);
state1(7,i)=s1(i);
state1(8,i)=xvap1(i);
state1(9,i)=e1(i);
end

if xvap1(:)>0.5
    W_net=zeros(1,lifetime);
    return;
end

%% state 2: separation at constant enthalpy
p2=ones(1,lifetime)*pressure_separator; % separator pressure
m2=m1;
h2=h1;

for i=1:length(year)
    T2(i)=XSteam('T_ph',p2(i),h2(i));
    xvap2(i)=XSteam('x_ph',p2(i),h2(i));
    cp2l(i)=XSteam('cpL_p',p2(i));
    cp2v(i)=XSteam('cpV_p',p2(i));
    cp2(i)=xvap2(i)*cp2v(i)+(1-xvap2(i))*cp2l(i);
    rho2l(i)=XSteam('rhoL_p',p2(i));
    rho2v(i)=XSteam('rhoV_p',p2(i));
    rho2(i)=xvap2(i)*rho2v(i)+(1-xvap2(i))*rho2l(i);
    s2(i)=XSteam('s_ph',p2(i),h2(i));
    e2(i)=(h2(i)-h0)-(T0+273.15)*(s2(i)-s0);

    state2(1,i)=p2(i);
    state2(2,i)=T2(i);
    state2(3,i)=m2(i);
    state2(4,i)=cp2(i);
    state2(5,i)=rho2(i);
    state2(6,i)=h2(i);
    state2(7,i)=s2(i);
    state2(8,i)=xvap2(i);
    state2(9,i)=e2(i);
end

%% evaluation of the mass flow rates after the flash process
for i=1:length(year)
    h3(i)=XSteam('hV_p',p2(i));
    h4(i)=XSteam('hL_p',p2(i));
end
m4=m2.*(h3-h2)./(h3-h4);
m3=m2.*(h2-h4)./(h3-h4);

%% state 3: saturated vapour from flash process/inlet turbine
p3=p2;
T3=T2;

for i=1:length(year)
    cp3(i)=XSteam('cpV_p',p3(i));
    rho3(i)=XSteam('rhoV_p',p3(i));
    s3(i)=XSteam('s_ph',p3(i),h3(i));
    xvap3(i)=XSteam('x_ph',p3(i),h3(i));
    e3(i)=(h3(i)-h0)-(T0+273.15)*(s3(i)-s0);

    state3(1,i)=p3(i);
    state3(2,i)=T3(i);
    state3(3,i)=m3(i);
    state3(4,i)=cp3(i);
    state3(5,i)=rho3(i);
    state3(6,i)=h3(i);
    state3(7,i)=s3(i);
    state3(8,i)=xvap3(i);
    state3(9,i)=e3(i);
end

%% state 4: saturated liquid from flash process/first injection state

```

```

p4=p2;
T4=T2;

for i=1:length(year)
    cp4(i)=XSteam('cpL_p',p4(i));
    rho4(i)=XSteam('rhoL_p',p4(i));
    s4(i)=XSteam('s_ph',p4(i),h4(i));
    xvap4(i)=XSteam('x_ph',p4(i),h4(i));
    e4(i)=(h4(i)-h0)-(T0+273.15)*(s4(i)-s0);

    state4(1,i)=p4(i);
    state4(2,i)=T4(i);
    state4(3,i)=m4(i);
    state4(4,i)=cp4(i);
    state4(5,i)=rho4(i);
    state4(6,i)=h4(i);
    state4(7,i)=s4(i);
    state4(8,i)=xvap4(i);
    state4(9,i)=e4(i);
end
%% state 5: outlet turbine/inlet condenser
m5=m3;
p5=ones(1,lifetime)*pressure_turb_outlet;

for i=1:length(year)

    state5(1,i)=p5(i); % pressure outlet at turbine
    state5(3,i)=m5(i);

    h5s(i)=XSteam('h_ps',p5(i),s3(i)); % [kJ/kg]
    h5(i)=h3(i)-eta_turb*(h3(i)-h5s(i)); % [kJ/kg]
    T5(i)=XSteam('T_ph',p5(i),h5(i)); % [°C]
    cp5(i)=XSteam('Cp_pT',p5(i),T5(i)); % [kJ/kg/°C]
    rho5(i)=XSteam('rho_ph',p5(i),h5(i)); % [kg/m^3]
    s5(i)=XSteam('s_ph',p5(i),h5(i)); % [kJ/kg/°C]
    e5(i)=(h5(i)-h0)-(T0+273.15)*(s5(i)-s0);

    % check if fraction of vapour is high enough to not damage the turbine
    xvap5(i)=XSteam('x_ph',p5(i),h5(i));
    if xvap5(i)<xvap_lim
        W_net=zeros(1,lifetime);
        return;
    end

    if isnan(cp5(i))
        cp5l(i)=XSteam('CpL_T',T5(i));
        cp5v(i)=XSteam('CpV_T',T5(i));
        cp5(i)=cp5l(i)*(1-xvap5(i))+cp5v(i)*xvap5(i);
    end

    state5(2,i)=T5(i);
    state5(4,i)=cp5(i);
    state5(5,i)=rho5(i);
    state5(6,i)=h5(i);
    state5(7,i)=s5(i);
    state5(8,i)=xvap5(i);
    state5(9,i)=e5(i);

    % evaluation of the turbine's specific work
    w_turb(i)=h3(i)-h5(i);
    W_turb(i)=w_turb(i)*m3(i)/1000; % [MWe]
    W_gen(i)=eta_gen*W_turb(i); % [MWe]
end

%% state 6: outlet condenser/inlet condensation pump
% evaluation of condensation heat. The steam is totally condensate
p6=p5;
for i=1:length(year)
    h6(i)=XSteam('hL_T',T5(i));
    qcond(i)=h5(i)-h6(i)*(1-x_ncg); % [kJ/kg]
    Qcond(i)=m5(i)*qcond(i)/1000; % [MWth]

    % evaluation of sprayed flow rate
    T10(i)=Tinj;
    if Tinj>T5(i)

```

```

        W_net=zeros(1,length(year));
        return;
end
h10(i)=XSteam('hL_T',T10(i));
mspray(i)=m5(i)*qcond(i)/(h6(i)-h10(i));

% set state6 parameter
T6(i)=XSteam('Tsat_p',p5(i));
m6(i)=(1-x_ncg)*m5(i)+mspray(i);
cp6(i)=XSteam('Cp_ph',p6(i),h6(i));
rho6(i)=XSteam('rho_ph',p6(i),h6(i));
s6(i)=XSteam('s_ph',p6(i),h6(i));
xvap6(i)=XSteam('x_ph',p6(i),h6(i));
e6(i)=(h6(i)-h0)-(T0+273.15)*(s6(i)-s0);

state6(1,i)=p6(i);
state6(2,i)=T6(i);
state6(3,i)=m6(i);
state6(4,i)=cp6(i);
state6(5,i)=rho6(i);
state6(6,i)=h6(i);
state6(7,i)=s6(i);
state6(8,i)=xvap6(i);
state6(9,i)=e6(i);

% Power of ejectors to remove uncondensed gas (CO2+H2S+Hg)
m_ncg(i)=m5(i)*x_ncg; %[kg/s]
Runiversal=8.314; %[J/molK]
MM_Hg=200.59; %[g/mol]
MM_CO2=44.01; %[g/mol]
MM_H2S=34.0818; %[g/mol]
R_Hg=Runiversal/MM_Hg*1000; %[J/gK]=*1000[J/kgK]
R_CO2=Runiversal/MM_CO2*1000; %[J/gK]=*1000[J/kgK]
R_H2S=Runiversal/MM_H2S*1000; %[J/gK]=*1000[J/kgK]
% Ideal gas law assumption
rho_Hg=p6(i)*1e5/R_Hg/(T6(i)+273.15); %[kg/m3]
rho_CO2=p6(i)*1e5/R_CO2/(T6(i)+273.15); %[kg/m3]
rho_H2S=p6(i)*1e5/R_H2S/(T6(i)+273.15); %[kg/m3]
cv_co2=0.658; %[kJ/kgK]
cv_h2s=0.76; %[kJ/kgK]
cv_hg=1.0145; %[kJ/kgK]
cv_ncg=cv_co2*xCO2+cv_h2s*xH2S+cv_hg*xHg;
cp_co2=0.849; %[kJ/kgK]
cp_h2s=1.01; %[kJ/kgK]
cp_hg=0.140; %[kJ/kgK]
cp_ncg=cp_co2*xCO2+cp_h2s*xH2S+cp_hg*xHg; %[kJ/kgK]
n_ncg=cp_ncg/cv_ncg; % [-]
R_ncg=R_Hg*xHg+R_CO2*xCO2+R_H2S*xH2S; %[J/kgK]
p_compr=pressure_compr_outlet; %[bar]
if isnan(n_ncg)
    W_compr(i)=0;
else
    W_compr(i)=m_ncg(i)*R_ncg*(T5(i)+273.15)*n_ncg/(n_ncg-1)/eta_compr*((p_compr/p5(i))^(n_ncg-1)/n_ncg-1)/10^6; %[MW]
end
end

%% state 7: outlet condensation pump and inlet cooling tower
% need to know the pressure output to evaluate the pump power
p7=p6+1.5;
m7=m6;
for i=1:lifetime
    h7s(i)=XSteam('h_ps',p7(i),s6(i));
    h7(i)=h6(i)-(h6(i)-h7s(i))/eta_condpump;
    T7(i)=XSteam('T_ph',p7(i),h7(i));
    cp7(i)=XSteam('cp_ph',p7(i),h7(i));
    rho7(i)=XSteam('rho_ph',p7(i),h7(i));
    s7(i)=XSteam('s_ph',p7(i),h7(i));
    xvap7(i)=XSteam('x_ph',p7(i),h7(i));
    e7(i)=(h7(i)-h0)-(T0+273.15)*(s7(i)-s0);

    state7(1,i)=p7(i);
    state7(2,i)=T7(i);
    state7(3,i)=m7(i);
    state7(4,i)=cp7(i);

```

```

state7(5,i)=rho7(i);
state7(6,i)=h7(i);
state7(7,i)=s7(i);
state7(8,i)=xvap7(i);
state7(9,i)=e7(i);

% Condensation pump power consumption
W_condpump(i)=m7(i)/rho7(i)*3600*abs(p7(i)-p6(i))*10^5/eta_condpump/3.6E6/1000;    %[MW]
end

%% state 8: outlet cooling tower
% here the temperature is reduced up to the injection temperature
p8=p7;
m8=m7;
for i=1:lifetime
    T8(i)=Tinj;
    if T7(i)<T8(i)
        W_net=zeros(1,lifetime); % The injection temperature is higher than the temperature entering in the
cooling tower
        return;
    end

    h8(i)=XSteam('h_pT',p8(i),T8(i));
    cp8(i)=XSteam('Cp_ph',p8(i),h8(i));
    rho8(i)=XSteam('rho_ph',p8(i),h8(i));
    s8(i)=XSteam('s_ph',p8(i),h8(i));
    xvap8(i)=XSteam('x_ph',p8(i),h8(i));
    e8(i)=(h8(i)-h0)-(T0+273.15)*(s8(i)-s0);

    state8(1,i)=p8(i);
    state8(2,i)=T8(i);
    state8(3,i)=m8(i);
    state8(4,i)=cp8(i);
    state8(5,i)=rho8(i);
    state8(6,i)=h8(i);
    state8(7,i)=s8(i);
    state8(8,i)=xvap8(i);
    state8(9,i)=e8(i);

    % cooling tower, fan power evaluation
    q_cooling_tow(i)=h7(i)-h8(i); %[kJ/kg]
    Qcool_tow(i)=q_cooling_tow(i)*m8(i)/1000; %[MWth]

    W_tower_fan(i)=Qcool_tow(i)*1000/kWt_to_kWe; %[kW]
end

%% state 9: Injection state, after blowdown pump
pressure_injection=rho8(1)*9.81*L_inj/1e5;
p9=ones(1,lifetime)*pressure_injection;
T9=T8;
m9=m8-mspray;
for i=1:lifetime
    h9(i)=XSteam('h_pT',p9(i),T9(i));
    cp9(i)=XSteam('Cp_ph',p9(i),h9(i));
    rho9(i)=XSteam('rho_ph',p9(i),h9(i));
    s9(i)=XSteam('s_ph',p9(i),h9(i));
    xvap9(i)=XSteam('x_ph',p9(i),h9(i));
    e9(i)=(h9(i)-h0)-(T0+273.15)*(s9(i)-s0);

    state9(1,i)=p9(i);
    state9(2,i)=T9(i);
    state9(3,i)=m9(i);
    state9(4,i)=cp9(i);
    state9(5,i)=rho9(i);
    state9(6,i)=h9(i);
    state9(7,i)=s9(i);
    state9(8,i)=xvap9(i);
    state9(9,i)=e9(i);

    % blow down pump power consumption
    W_bdpump(i)=m9(i)/rho9(i)*3600*(p9(i)-p8(i))*10^5/eta_bdpump/3.6E6/1000;    %[MW]
end

%% circulation pump power

```

```

for i=1:lifetime
    W_circumpump(i)=mspray(i)/rho8(i)*3600*(p8(i)-p5(i))*10^5/eta_circumpump/3.6E6/1000;    %[MW]
end

%% result section
W_net=zeros(1,lifetime);
for i=1:lifetime
    W_net(i)=W_gen(i)-W_circumpump(i)-W_bdpump(i)-W_condpump(i)-W_tower_fan(i)/1000-W_compr(i); %[MW]
    SSC(i)=m3(i)*3600/(W_net(i)*1000); %[kg/kWh]
    eta_util(i)=W_net(i)*1000/(m1(i)*e1(i));
end

```

IX. Economic_model_binary

```

%% economic model for binary power plant
% producer price index at year 2023
PPIpipe=2.971;
PPItg=1.612;
PPIhx=2.416;
PPIpump=2.064;
PPIpe=2.560;
PPIoeg=2.624;
PPIs=2.612;
PPIpermit=2.227;
PPIoegs=1.564;
CEPCI_index=2.017;

%% Surface plant cost (Cplant)
% indirect and project contingency for plant
XICp=1.12;
XPCp=1.15;
% equipment and construction cost fractions
Xse=1.15; % secondary equipment cost
Xcl=0.58; % construction labour cost
Xcm=0.11; % construction material cost
Xst=0.00; % sales taxes
Xf=0.4; % equipment freight

% Steam turbine cost
Stg=1.2; % accounts for type of fluid, different from steam =1.2
Ctg=0.67*PPItg*(Stg*2830*(max(W_turb)*1e3)^0.745+3680*(max(W_turb)*1e3)^0.617); % [$] power in kWe
(turbine+generator)

% Circulation and injection pump cost
Spump=1; % it is the material multiplayer factor, in our case for water and non-corrosive liquid, the base
case is the iron pump
Ccirc_pump=PPIpump*Spump*1185*(1.34*max(W_circumpump*1e3)^0.767); %[$] power in kWe
Cinj_pump=PPIpump*Spump*1750*(1.34*max(W_bdpump*1e3)^0.7);
Ccond_pump=PPIpump*Spump*1185*(1.34*max(W_condpump*1e3)^0.767);

% Cooling tower and condenser
Qref_BAC=1; %[MWth]
TDC=0.252; % tower design coefficient, in this case open water
deltaT_appr=T10(1)-T0; % difference between the fluid temperature leaving the tower and the ambient
temperature
deltaT_range=T9(1)-T10(1); % difference between inlet and exit fluid temperature
c_coolingcost=7.31*1e3*(1/deltaT_appr)+1.23*1e3*(1/(deltaT_appr+deltaT_range)); %[$/kWth]

Cref_BAC=PPIpe*TDC*Qref_BAC*1e3*c_coolingcost; %[$]
Ccooling_tower=Cref_BAC*(Qcool_tow(1)/Qref_BAC)^0.8; %[$]

% Condenser cost
if typeHX==1
    Chx=(239*max(A_hx)+13400)*PPIhx;
else if typeHX==2
    Chx=(69*max(A_hx)+4670)*PPIhx;
else
    Chx=(239*max(A_hx)+13400)*PPIhx;
    % fprintf('The evaporator is shell and tube type');
end
end

% Condenser cost

```

```

if typeCOND==1
    Ccondenser=(239*max(A_cond)+13400)*PPIhx;
else if typeCOND==2
    Ccondenser=(69*max(A_cond)+4670)*PPIhx;
else
    Ccondenser=(239*max(A_cond)+13400)*PPIhx;
end
end

% total plant cost
Cplant_PEC=Ctg+Ccond_pump+Ccooling_tower+Ccirc_pump+Cinj_pump+Ccondenser+Chx;
Cplant_TEC=Cplant_PEC*Xse;
Cplant_BEC=Cplant_TEC*(1+Xcl+Xcm+Xst+Xf);
Cplant=Cplant_BEC*XPCp*XICp;

%% Well cost (Cwell)
% indirect and project contingency for well
XICwell=1.05;
XPCwell=1.15;

Lwell_prod=3000;
Lwell_inj=Lwell_prod/2;
Cwell_inj=XICwell*XPCwell*PPIoeg*n_inj*(0.105*Lwell_inj^2+1776*Lwell_inj*d_inj+275300); %[$]
Cwell_prod=XICwell*XPCwell*PPIoeg*n_prod*(0.105*Lwell_prod^2+1776*Lwell_prod*d_prod+275300); %[$]
Cwell=(Cwell_inj+Cwell_prod)/Succrate; %[$]

% fprintf('The well cost is %.2f M$\n', Cwell/1e6);

%% Surface Piping system (Csurfacepiping)
% indirect and project contingency for piping
XICpipe=1.12;
XPCpipe=1.15;

dpipe=d_prod;
Lpipe=5000;
c_surfacepiping=2205*dpipe^2+134; %[$/m]
Csurfacepiping=XPCpipe*XICpipe*PPIpipe*(c_surfacepiping*Lpipe);

% fprintf('The surface piping cost is %.2f M$\n', Csurfacepiping/1e6);

%% Well field cost (Cwellfield)
% indirect and project contingency for permitting
XICwf=1.05;
XPCwf=1.15;

Cpermitting=XICwf*XPCwf*PPIpermit*665700; %[$/site]
Cwellfield=Cpermitting;

%% Exploration cost (Cexploration)
% indirect and project contingency for exploration
XICexpl=1.05;
XPCexpl=1.15;

Cmodeling=XICexpl*XPCexpl*PPIoegs*508000; %[$/site]
Nslim_hole=2;
Lslim_hole=3000;
dslim_hole=0.05;
Cslim_hole=XICexpl*XPCexpl*PPIoegs*Nslim_hole*(0.105*Lslim_hole^2+1776*Lslim_hole*dslim_hole+275300); %[$];
Cexploration=Cmodeling+Cslim_hole;

%% Stimulation cost (Cstimulation)
% EGSvalue=input('Is it a EGS system? 1=yes, 0=no: ');
% if EGSvalue==1
%     XICstim=1.05;
%     XPCstim=1.15;
%     Gwells=500; % mass flow rate for stimulation
%     Cstimulation=XICstim*XPCstim*Gwells*715000;
% else
%     Cstimulation=0;
% end
Cstimulation=0;

%% Grid cost (Cgrid)
conversion_euro_dollar=1.12; %[$/€]
Cgrid_investment=80/conversion_euro_dollar*W_gen(1)*1000; % [$]

```

```

Lgrid_connection=1000; % [m]
Cgrid_connection=Lgrid_connection*100/conversion_euro_dollar; % [$]
Cgrid=Cgrid_connection+Cgrid_investment;

%% Operation and maintenance cost (Coem)
% it can be between 0.045 and 0.065, use 0.065 for the worst scenario
Foem=0.05;

%% Greenfield cost (Cgreen) and Brownfield cost (Cbrown)
if fieldtype==1
    Cbrown=Cplant+Csurfacepiping+Cwell_prod+Cgrid;
    Coem_brown=Cbrown*Foem;
else
    Cgreen=Cplant+Csurfacepiping+Cwell+Cwellfield+Cexploration+Cstimulation+Cgrid;
    Coem_green=Cgreen*Foem;
end

%% LCOE risk-adjusted (LCOErad)
time_in_year=8766;

% WACC parameter
% kd=1.85569/100;
% Rf=1.13363/100;
% EMRP=5.7675/100;
% beta_e=1;
% ke=Rf+EMRP*beta_e;
% This equity and debt percentage are related to high risk for Independent
% Power Producer (IPP). For IPP and low risk set equity=0.3 and debt=0.7.
% In case of Investor-Owned Utility (IOU) and low risk set equity 0.5 and
% debt=0.55, for high risk set equity=0.5 and debt)0.45
% equity=0.4;
% debt=0.6;
% taxrate=0.1;
% Wd=debt/(equity+debt);
% We=equity/(equity+debt);
% WACC=Wd*kd*(1-taxrate)+ke*We;
WACC=0.05;

W_net=[0,W_net];

% LCOE greenfield, brownfield and general
if fieldtype==1
    capex_brown=zeros(1,lifetime+1);
    capex_brown(1)=Cbrown;
    opex_brown=ones(1,lifetime+1)*Coem_brown;
    opex_brown(1)=0;
    varcost_brown=ones(1,lifetime+1)*Coem_brown*0; %check value of variable cost
    varcost_brown(1)=0;
    for i=1:lifetime+1
        netenergy(i)=W_net(i)*CF*time_in_year; %[MWh]
        LCOEnum(i)=(capex_brown(i)+opex_brown(i)+varcost_brown(i))/(1+WACC)^i;
        LCOEden(i)=netenergy(i)/(1+WACC)^i;
    end
    LCOErad=sum(LCOEnum)/sum(LCOEden);
else
    capex_green=zeros(1,lifetime+1);
    capex_green(1)=Cgreen;
    opex_green=ones(1,lifetime+1)*Coem_green;
    opex_green(1)=0;
    varcost_green=ones(1,lifetime+1)*Coem_green*0; %check value of variable cost
    varcost_green(1)=0;
    for i=1:lifetime+1
        netenergy(i)=W_net(i)*CF*time_in_year; %[MWh]
        LCOEnum(i)=(capex_green(i)+opex_green(i)+varcost_green(i))/(1+WACC)^i;
        LCOEden(i)=netenergy(i)/(1+WACC)^i;
    end
    LCOErad=sum(LCOEnum)/sum(LCOEden);
end

%% NPV
% general
revenue_ele=netenergy*1000*eleprice;
IRR=WACC;
year=(0:1:lifetime);
if fieldtype==1

```

```

NPV(1)=-capex_brown(1)*(1-CAPEXincentives);
for i=2:lifetime+1
    netrevenue(i)=((revenue_ele(i)*(1+REVENUEincentives))-(opex_brown(i)*(1-OPEXincentives))-
varcost_brown(i))/(1+IRR)^year(i);
    NPV(i)=NPV(i-1)+netrevenue(i);
    if sign(NPV(i-1))+sign(NPV(i))==0
        PBT=i-1;
        % fprintf('The pay back time is in the year number %d\n', (i-1)-1);
    end
end
else
NPV(1)=-capex_green(1)*(1-CAPEXincentives);
for i=2:lifetime+1
    netrevenue(i)=((revenue_ele(i)*(1+REVENUEincentives))-(opex_green(i)*(1-OPEXincentives))-
varcost_green(i))/(1+IRR)^year(i);
    NPV(i)=NPV(i-1)+netrevenue(i);
    if sign(NPV(i-1))+sign(NPV(i))==0
        PBT=i-1;
        % fprintf('The pay back time is in the year number %d\n', (i-1)-1);
    end
end
end
end

```

X. *Economic_model_dry_steam*

```

%% economic model for dry steam power plant
% producer price index at year 2023
PPIpipe=2.971;
PPItg=1.612;
PPIhx=2.416;
PPIpump=2.064;
PPIpe=2.560;
PPIoeg=2.624;
PPIids=2.612;
PPIpermit=2.227;
PPIoegs=1.564;
CEPCI_index=2.017;

%% Surface plant cost (Cplant)
% indirect and project contingency for plant
XICp=1.12;
XPCp=1.15;
% equipment and construction cost fractions
Xse=1.15; % secondary equipment cost
Xcl=0.58; % construction labour cost
Xcm=0.11; % construction material cost
Xst=0.00; % sales taxes
Xf=0.4; % equipment freight

% Steam turbine cost
Stg=1; %accounts for the fluid, in this case steam
Ctg=0.67*PPItg*(Stg*2830*(max(W_turb)*1e3)^0.745+3680*(max(W_turb)*1e3)^0.617); % [$] power in kWe

% Circulation and injection pump cost
Spump=1; % it is the material multiplayer factor, in our case for water and non-corrosive liquid, the base
case is the iron pump
Ccirc_pump=PPIpump*Spump*1185*(1.34*(max(W_circump)*1e3)^0.767); %[$] power in kWe
Cinj_pump=PPIpump*Spump*1750*(1.34*(max(W_bdmpump)*1e3)^0.7);
Ccond_pump=PPIpump*Spump*1185*(1.34*(max(W_condump)*1e3)^0.767);

% Cooling tower and condenser
Qref_BAC=1; %[MWth]
TDC=0.252; % tower design coefficient, in this case open water
deltaT_appr=T5(1)-T0; % difference between the fluid temperature leaving the tower and the ambient
temperature
deltaT_range=T4(1)-T5(1); % difference between inlet and exit fluid temperature
c_coolingcost=7.31*1e3*(1/deltaT_appr)+1.23*1e3*(1/(deltaT_appr+deltaT_range)); %[$/kWth]

Cref_BAC=PPIpe*TDC*Qref_BAC*1e3*c_coolingcost; %[$]
Ccooling_tower=Cref_BAC*(max(Qcool_tow)/Qref_BAC)^0.8; %[$]

% Direct contact Condenser cost
FOB_condenser=22000; %[$]

```

```

n_condenser=0.6;
Ccondenser=FOB_condenser*((max(mspray)/rho5(1)*1000)/33)^n_condenser; % 33 L/s is the reference

% Cost of compressor (centrifugal=7MPa)
FOB_compr=875000; %[$]
compressor_rated_power=1000; % kW
n_compressor=0.53; % range 500-4000 kW
Fpressure=0.8; % pressure exiting max 1.7MPa
Ccompr=CEPCI_index*FOB_compr*Fpressure*(max(W_compr)*1000/compressor_rated_power)^n_compressor;

% Cost of AMIS unit
% activated carbon filter for Hg removal
FOB_removal=23500; %[$]
V_Hg=max(m_ncg)*xHg/rho_Hg(1)*1000; %[dm^3/s]
V_gas_ref=70; %[dm3/s]
n_ac_removal=0.32; %range 7-70 dm3/s
C_Hg_removal=FOB_removal*(V_Hg/V_gas_ref)^n_ac_removal; %[$]

% Titanium dioxide catalyst removal
V_H2S=max(m_ncg)*xH2S/rho_H2S(1)*1000; %[dm^3/s]
n_tio2_removal=0.67; %range 7-700 dm3/s
F_tio2=8;
C_H2S_removal=FOB_removal*F_tio2*(V_H2S/V_gas_ref)^n_tio2_removal; %[$]

% wet scrubber packed column
FOB_wet_scrubber=35000; % [$]
V_SO2=V_H2S/1000; %[m3/s]
n_wet_scrubber=0.39; % range 0.5-1.65 m3/s
V_scrubber_ref=1.65; % [m3/s]
C_wet_scrubber=FOB_wet_scrubber*(V_SO2/V_scrubber_ref)^n_wet_scrubber;

Camis=CEPCI_index*(C_Hg_removal+C_H2S_removal+C_wet_scrubber); %[$]

% total plant cost
Cplant_PEC=Ctg+Ccondenser+Ccond_pump+Ccooling_tower+Ccirc_pump+Cinj_pump+Ccompr+Camis;
Cplant_TEC=Cplant_PEC*Xse;
Cplant_BEC=Cplant_TEC*(1+Xcl+Xcm+Xst+Xf);
Cplant=Cplant_BEC*XPCp*XICp;

%% Well cost (Cwell)
% success rate in drilling
% indirect and project contingency for well
XICwell=1.05;
XPCwell=1.15;

Lwell_prod=2000;
Lwell_inj=Lwell_prod/2;
Cwell_inj=XICwell*XPCwell*PPIoeg*n_inj*(0.105*Lwell_inj^2+1776*Lwell_inj*d_inj+275300); %[$]
Cwell_prod=XICwell*XPCwell*PPIoeg*n_prod*(0.105*Lwell_prod^2+1776*Lwell_prod*d_prod+275300); %[$]
Cwell=(Cwell_inj+Cwell_prod)/Succrate; %[$]

%% Surface Piping system (Csurfacepiping)
% indirect and project contingency for piping
XICpipe=1.12;
XPCpipe=1.15;

dpipe=d_prod;
Lpipe=5000;
c_surfacepiping=2205*dpipe^2+134; %[$/m]
Csurfacepiping=XPCpipe*XICpipe*PPIpipe*(c_surfacepiping*Lpipe);

%% Well field cost (Cwellfield)
% indirect and project contingency for permitting
XICwf=1.05;
XPCwf=1.15;

Cpermitting=XICwf*XPCwf*PPIpermit*665700; %[$/site]
Cwellfield=Cpermitting;

%% Exploration cost (Cexploration)
% indirect and project contingency for exploration
XICexpl=1.05;
XPCexpl=1.15;

```

```

Cmodeling=XICexpl*XPCexpl*PPIoegs*508000; %[$/site]
Nslim_hole=2;
Lslim_hole=Lwell_prod;
dslim_hole=0.05;
Cslim_hole=XICexpl*XPCexpl*PPIoegs*Nslim_hole*(0.105*Lslim_hole^2+1776*Lslim_hole*dslim_hole+275300); %[$];
Cexploration=Cmodeling+Cslim_hole;

%% Stimulation cost (Cstimulation)
% EGSvalue=input('Is it a EGS system? 1=yes, 0=no: ');
% if EGSvalue==1
%   XICstim=1.05;
%   XPCstim=1.15;
%   Gwells=500; % mass flow rate for stimulation
%   Cstimulation=XICstim*XPCstim*Gwells*715000;
% else
%   Cstimulation=0;
% end
Cstimulation=0;

%% Grid cost (Cgrid)
conversion_euro_dollar=1.12; %[$/$]
Cgrid_investment=80/conversion_euro_dollar*W_gen(1)*1000; % [$]
Lgrid_connection=1000; % [m]
Cgrid_connection=Lgrid_connection*100/conversion_euro_dollar; % [$]
Cgrid=Cgrid_connection+Cgrid_investment;

%% Operation and maintenance cost (Coem)
% it can be between 0.045 and 0.065, use 0.065 for the worst scenario
Foem=0.05;

%% Greenfield cost (Cgreen) and Brownfield cost (Cbrown)
% fieldtype=input('Type of field. 1=brownfield, 2=greenfield: ');
if fieldtype==1
    Cbrown=Cplant+Csurfacepiping+Cwell_prod+Cgrid;
    Coem_brown=Cbrown*Foem;
else
    Cgreen=Cplant+Csurfacepiping+Cwell+Cwellfield+Cexploration+Cstimulation+Cgrid;
    Coem_green=Cgreen*Foem;
end

%% LCOE risk-adjusted (LOCERad)
time_in_year=8766;

% WACC parameter
% kd=1.85569/100;
% Rf=1.13363/100;
% EMRP=5.7675/100;
% beta_e=1;
% ke=Rf+EMRP*beta_e;
% This equity and debt percentage are related to high risk for Independent
% Power Producer (IPP). For IPP and low risk set equity=0.3 and debt=0.7.
% In case of Investor-Owned Utility (IOU) and low risk set equity 0.5 and
% debt=0.55, for high risk set equity=0.5 and debt)0.45
% equity=0.4;
% debt=0.6;
% taxrate=0.1;
% Wd=debt/(equity+debt);
% We=equity/(equity+debt);
% WACC=Wd*kd*(1-taxrate)+ke*We;
WACC=0.05; %[%]

W_net=[0,W_net];

% LCOE greenfield, brownfield and general
if fieldtype==1
    capex_brown=zeros(1,lifetime+1);
    capex_brown(1)=Cbrown;
    opex_brown=ones(1,lifetime+1)*Coem_brown;
    opex_brown(1)=0;
    varcost_brown=ones(1,lifetime+1)*Coem_brown*0; %check value of variable cost
    varcost_brown(1)=0;
    for i=1:lifetime+1
        netenergy(i)=W_net(i)*CF*time_in_year; %[MWh]
    end
end

```

```

        LCOEnum(i)=(capex_brown(i)+opex_brown(i)+varcost_brown(i))/(1+WACC)^i;
        LCOEden(i)=netenergy(i)/(1+WACC)^i;
    end
    LCOErad=sum(LCOEnum)/sum(LCOEden);
else
    capex_green=zeros(1,lifetime+1);
    capex_green(1)=Cgreen;
    opex_green=ones(1,lifetime+1)*Coem_green;
    opex_green(1)=0;
    varcost_green=ones(1,lifetime+1)*Coem_green*0; %check value of variable cost
    varcost_green(1)=0;
    for i=1:lifetime+1
        netenergy(i)=W_net(i)*CF*time_in_year; %[MWh]
        LCOEnum(i)=(capex_green(i)+opex_green(i)+varcost_green(i))/(1+WACC)^i;
        LCOEden(i)=netenergy(i)/(1+WACC)^i;
    end
    LCOErad=sum(LCOEnum)/sum(LCOEden);
end

%% NPV
% general
revenue_ele=netenergy*1000*eleprice;
IRR=WACC;
year=(0:1:lifetime);
if fieldtype==1
    NPV(1)=-capex_brown(1)*(1-CAPEXincentives);
    for i=2:lifetime+1
        netrevenue(i)=((revenue_ele(i)*(1+REVENUEincentives))-(opex_brown(i)*(1-OPEXincentives))-
varcost_brown(i))/(1+IRR)^year(i);
        NPV(i)=NPV(i-1)+netrevenue(i);
        if sign(NPV(i-1))+sign(NPV(i))==0
            PBT=i-1;
            % fprintf('The pay back time is in the year number %d\n', (i-1)-1);
        end
    end
else
    NPV(1)=-capex_green(1)*(1-CAPEXincentives);
    for i=2:lifetime+1
        netrevenue(i)=((revenue_ele(i)*(1+REVENUEincentives))-(opex_green(i)*(1-OPEXincentives))-
varcost_green(i))/(1+IRR)^year(i);
        NPV(i)=NPV(i-1)+netrevenue(i);
        if sign(NPV(i-1))+sign(NPV(i))==0
            PBT=i-1;
            % fprintf('The pay back time is in the year number %d\n', (i-1)-1);
        end
    end
end
end
end

```

XI. *Economic model_flash*

```

%% Economic model flash
% producer price index at year 2023
PPIpipe=2.971;
PPItg=1.612;
PPIhx=2.416;
PPIpump=2.064;
PPIpe=2.560;
PPIoeg=2.624;
PPI ds=2.612;
PPIpermit=2.227;
PPIoegs=1.564;
CEPCI_index=2.017;

% Surface plant cost (Cplant)
% indirect and project contingency for plant
XICp=1.12;
XPCp=1.15;
% equipment and construction cost fractions
Xse=1.15; % secondary equipment cost
Xcl=0.58; % construction labour cost
Xcm=0.11; % construction material cost
Xst=0.00; % sales taxes
Xf=0.4; % equipment freight

```

```

% Steam turbine cost
Stg=1; %accounts for the fluid, in this case steam
Ctg=0.67*PPItg*(Stg*2830*(max(W_turb)*1e3)^0.745+3680*(max(W_turb)*1e3)^0.617); % [$] power in kWe

% Circulation and injection pump cost
Spump=1; % it is the material multiplayer factor, in our case for water and non-corrosive liquid, the base
case is the iron pump
Ccirc_pump=PPIpump*Spump*1185*(1.34*(max(W_circump)*1e3)^0.767); %[$] power in kWe
Cinj_pump=PPIpump*Spump*1750*(1.34*(max(W_bdppump)*1e3)^0.7);
Ccond_pump=PPIpump*Spump*1185*(1.34*(max(W_condpump)*1e3)^0.767);

% Cooling tower and condenser
Qref_BAC=1; %[MWth]
TDC=0.252; % tower design coefficient, in this case open water
deltaT_appr=T8(1)-T0; % difference between the fluid temperature leaving the tower and the ambient
temperature
deltaT_range=T7(1)-T8(1); % difference between inlet and exit fluid temperature
c_coolingcost=7.31*1e3*(1/deltaT_appr)+1.23*1e3*(1/(deltaT_appr+deltaT_range)); %[$/kWth]

Cref_BAC=PPIpe*TDC*Qref_BAC*1e3*c_coolingcost; %[$]
Ccooling_tower=Cref_BAC*(Qcool_tow(1)/Qref_BAC)^0.8; %[$]

% Direct contact Condenser cost
FOB_condenser=22000; %[$]
n_condenser=0.6; % in theory it is 0.6 between 3.3 and 600 L/s
Ccondenser=FOB_condenser*((mspray(1)/rho8(1)*1000)/33)^n_condenser; % 33 L/s is the reference

% Cost of compressor (centrifugal=7MPa)
FOB_compr=875000; %[$]
compressor_rated_power=1000; % kW
n_compressor=0.53; % range 500-4000 kW
Fpressure=0.8; % pressure exiting max 1.7MPa
Ccompr=CEPCI_index*FOB_compr*Fpressure*(max(W_compr)*1000/compressor_rated_power)^n_compressor;

% Cost of AMIS unit
% activated carbon filter for Hg removal
FOB_removal=23500; %[$]
V_Hg=max(m_ncg)*xHg/rho_Hg(1)*1000; %[dm^3/s]
V_gas_ref=70; %[dm3/s]
n_ac_removal=0.32; %range 7-70 dm3/s
C_HG_removal=FOB_removal*(V_Hg/V_gas_ref)^n_ac_removal; %[$]

% Titanium dioxide catalyst removal
V_H2S=max(m_ncg)*xH2S/rho_H2S(1)*1000; %[dm^3/s]
n_tio2_removal=0.67; %range 7-700 dm3/s
F_tio2=8;
C_H2S_removal=FOB_removal*F_tio2*(V_H2S/V_gas_ref)^n_tio2_removal; %[$]

% wet scrubber packed column
FOB_wet_scrubber=35000; % [$]
V_SO2=V_H2S/1000; %[m3/s]
n_wet_scrubber=0.39; % range 0.5-1.65 m3/s
V_scrubber_ref=1.65; % [m3/s]
C_wet_scrubber=FOB_wet_scrubber*(V_SO2/V_scrubber_ref)^n_wet_scrubber;

Camis=CEPCI_index*(C_HG_removal+C_H2S_removal+C_wet_scrubber); %[$]

% total plant cost
Cplant_PEC=Ctg+Ccondenser+Ccond_pump+Ccooling_tower+Ccirc_pump+Cinj_pump+Ccompr+Camis;
Cplant_TEC=Cplant_PEC*Xse;
Cplant_BEC=Cplant_TEC*(1+Xcl+Xcm+Xst+Xf);
Cplant=Cplant_BEC*XPCp*XICp;

%% Well cost (Cwell)
% indirect and project contingency for well
XICwell=1.05;
XPCwell=1.15;

Lwell_prod=2000;
Lwell_inj=Lwell_prod/2;
Cwell_inj=XICwell*XPCwell*PPIoeg*n_inj*(0.105*Lwell_inj^2+1776*Lwell_inj*d_inj+275300); %[$]
Cwell_prod=XICwell*XPCwell*PPIoeg*n_prod*(0.105*Lwell_prod^2+1776*Lwell_prod*d_prod+275300); %[$]
Cwell=(Cwell_inj+Cwell_prod)/Succrate; %[$]

%% Surface Piping system (Csurfacepiping)

```

```

% indirect and project contingency for piping
XICpipe=1.12;
XPCpipe=1.15;

dpipe=d_prod;
Lpipe=5000;
c_surfacepiping=2205*dpipe^2+134; %[$/m]
Csurfacepiping=XPCpipe*XICpipe*PIpipe*(c_surfacepiping*Lpipe);

%% Well field cost (Cwellfield)
% indirect and project contingency for permitting
XICwf=1.05;
XPCwf=1.15;

Cpermitting=XICwf*XPCwf*PPipermit*665700; %[$/site]
Cwellfield=Cpermitting;

%% Exploration cost (Cexploration)
% indirect and project contingency for exploration
XICexpl=1.05;
XPCexpl=1.15;

Cmodeling=XICexpl*XPCexpl*PIIoegs*508000; %[$/site]
Nslim_hole=2;
Lslim_hole=3000;
dslim_hole=0.05;
Cslim_hole=XICexpl*XPCexpl*PIIoegs*Nslim_hole*(0.105*Lslim_hole^2+1776*Lslim_hole*dslim_hole+275300); %[$];
Cexploration=Cmodeling+Cslim_hole;

%% Stimulation cost (Cstimulation)
% EGSvalue=input('Is it a EGS system? 1=yes, 0=no: ');
% if EGSvalue==1
%   XICstim=1.05;
%   XPCstim=1.15;
%   Gwells=500; % mass flow rate for stimulation
%   Cstimulation=XICstim*XPCstim*Gwells*715000;
% else
%   Cstimulation=0;
% end
Cstimulation=0;

%% Grid cost (Cgrid)
conversion_euro_dollar=1.12; %[$/€]
Cgrid_investment=80/conversion_euro_dollar*W_gen(1)*1000; % [$]
Lgrid_connection=1000; % [m]
Cgrid_connection=Lgrid_connection*100/conversion_euro_dollar; % [$]
Cgrid=Cgrid_connection+Cgrid_investment;

%% Operation and maintenance cost (Coem)
% it can be between 0.045 and 0.065, use 0.065 for the worst scenario
Foem=0.05;

%% Greenfield cost (Cgreen) and Brownfield cost (Cbrown)
if fieldtype==1
    Cbrown=Cplant+Csurfacepiping+Cwell_prod+Cgrid;
    Coem_brown=Cbrown*Foem;
else
    Cgreen=Cplant+Csurfacepiping+Cwell+Cwellfield+Cexploration+Cstimulation+Cgrid;
    Coem_green=Cgreen*Foem;
end

%% LCOE risk-adjusted (LOCerad)
time_in_year=8766;
% WACC parameter
% kd=1.85569/100;
% Rf=1.13363/100;
% EMRP=5.7675/100;
% beta_e=1;
% ke=Rf+EMRP*beta_e;
% This equity and debt percentage are related to high risk for Independent
% Power Producer (IPP). For IPP and low risk set equity=0.3 and debt=0.7.
% In case of Investor-Owned Utility (IOU) and low risk set equity 0.5 and
% debt=0.55, for high risk set equity=0.5 and debt)0.45
% equity=0.4;

```

```

% debt=0.6;
% taxrate=0.1;
% Wd=debt/(equity+debt);
% We=equity/(equity+debt);
% WACC=Wd*kd*(1-taxrate)+ke*We;
WACC=0.05;

W_net=[0,W_net];

% LCOE greenfield, brownfield and general
if fieldtype==1
    capex_brown=zeros(1,lifetime+1);
    capex_brown(1)=Cbrown;
    opex_brown=ones(1,lifetime+1)*Coem_brown;
    opex_brown(1)=0;
    varcost_brown=ones(1,lifetime+1)*Coem_brown*0; %check value of variable cost
    varcost_brown(1)=0;
    for i=1:lifetime+1
        netenergy(i)=W_net(i)*CF*time_in_year; %[MWh]
        LCOEnum(i)=(capex_brown(i)+opex_brown(i)+varcost_brown(i))/(1+WACC)^i;
        LCOEden(i)=netenergy(i)/(1+WACC)^i;
    end
    LCOErad=sum(LCOEnum)/sum(LCOEden);
else
    capex_green=zeros(1,lifetime+1);
    capex_green(1)=Cgreen;
    opex_green=ones(1,lifetime+1)*Coem_green;
    opex_green(1)=0;
    varcost_green=ones(1,lifetime+1)*Coem_green*0; %check value of variable cost
    varcost_green(1)=0;
    for i=1:lifetime+1
        netenergy(i)=W_net(i)*CF*time_in_year; %[MWh]
        LCOEnum(i)=(capex_green(i)+opex_green(i)+varcost_green(i))/(1+WACC)^i;
        LCOEden(i)=netenergy(i)/(1+WACC)^i;
    end
    LCOErad=sum(LCOEnum)/sum(LCOEden);
end

%% NPV
% general
revenue_ele=netenergy*1000*eleprice;
IRR=WACC;
year=(0:1:lifetime);
if fieldtype==1
    NPV(1)=-capex_brown(1)*(1-CAPEXincentives);
    for i=2:lifetime+1
        netrevenue(i)=((revenue_ele(i)*(1+REVENUEincentives))-(opex_brown(i)*(1-OPEXincentives)))-
        varcost_brown(i))/(1+IRR)^year(i);
        NPV(i)=NPV(i-1)+netrevenue(i);
        if sign(NPV(i-1))+sign(NPV(i))==0
            PBT=i-1;
            % fprintf('The payback time is in the year number %d\n', (i-1)-1);
        end
    end
else
    NPV(1)=-capex_green(1)*(1-CAPEXincentives);
    for i=2:lifetime+1
        netrevenue(i)=((revenue_ele(i)*(1+REVENUEincentives))-(opex_green(i)*(1-OPEXincentives)))-
        varcost_green(i))/(1+IRR)^year(i);
        NPV(i)=NPV(i-1)+netrevenue(i);
        if sign(NPV(i-1))+sign(NPV(i))==0
            PBT=i-1;
            % fprintf('The payback time is in the year number %d\n', (i-1)-1);
        end
    end
end
end

```

XII. Economic_model_dry_steam_top_spillation

```

%% economic model for dry steam power plant
% producer price index at year 2023
PPIpipe=2.971;
PPItg=1.612;
PPIhx=2.416;

```

```

PPIpump=2.064;
PPIpe=2.560;
PPIoeg=2.624;
PPIids=2.612;
PPIpermit=2.227;
PPIoegs=1.564;
CEPCI_index=2.017;

%% Surface plant cost (Cplant)
% indirect and project contingency for plant
XICp=1.12;
XPCp=1.15;
% equipment and construction cost fractions
Xse=1.15; % secondary equipment cost
Xcl=0.58; % construction labour cost
Xcm=0.11; % construction material cost
Xst=0.00; % sales taxes
Xf=0.4; % equipment freight

% Steam turbine cost
Stg=1; %accounts for the fluid, in this case steam
Ctg=0.67*PPItg*(Stg*2830*(max(W_turb)*1e3)^0.745+3680*(max(W_turb)*1e3)^0.617); % [$] power in kWe

% Circulation and injection pump cost
Spump=1; % it is the material multiplier factor, in our case for water and non-corrosive liquid, the base
case is the iron pump
Ccirc_pump=PPIpump*Spump*1185*(1.34*(max(W_circpump)*1e3)^0.767); %[$] power in kWe
Cinj_pump=PPIpump*Spump*1750*(1.34*(max(W_bdmpump)*1e3)^0.7);
Ccond_pump=PPIpump*Spump*1185*(1.34*(max(W_condpump)*1e3)^0.767);

% Cooling tower and condenser
Qref_BAC=1; %[MWth]
TDC=0.252; % tower design coefficient, in this case open water
deltaT_appr=T5(1)-T0; % difference between the fluid temperature leaving the tower and the ambient
temperature
deltaT_range=T4(1)-T5(1); % difference between inlet and exit fluid temperature
c_coolingcost=7.31*1e3*(1/deltaT_appr)+1.23*1e3*(1/(deltaT_appr+deltaT_range)); %[$/kWth]

Cref_BAC=PPIpe*TDC*Qref_BAC*1e3*c_coolingcost; %[$]
Ccooling_tower=Cref_BAC*(max(Qcool_tow)/Qref_BAC)^0.8; %[$]

% Direct contact Condenser cost
FOB_condenser=22000; %[$]
n_condenser=0.6; % in theory it is 0.6 between 3.3 and 600 L/s
Ccondenser=FOB_condenser*((max(mspray)/rho5(1)*1000)/33)^n_condenser; % 33 L/s is the reference

% Cost of compressor (centrifugal=7MPa)
FOB_compr=875000; %[$]
compressorRatedPower=1000; % kW
n_compressor=0.53; % range 500-4000 kW
Fpressure=0.8; % pressure exiting max 1.7MPa
Ccompr=CEPCI_index*FOB_compr*Fpressure*(max(W_compr)*1000/compressorRatedPower)^n_compressor;

% Cost of AMIS unit
% activated carbon filter for Hg removal
FOB_removal=23500; %[$]
V_Hg=max(m_ncg)*xHg/rho_Hg(1)*1000; %[dm^3/s]
V_gas_ref=70; %[dm3/s]
n_ac_removal=0.32; %range 7-70 dm3/s
C_Hg_removal=FOB_removal*(V_Hg/V_gas_ref)^n_ac_removal; %[$]

% Titanium dioxide catalyst removal
V_H2S=max(m_ncg)*xH2S/rho_H2S(1)*1000; %[dm^3/s]
n_tio2_removal=0.67; %range 7-700 dm3/s
F_tio2=8;
C_H2S_removal=FOB_removal*F_tio2*(V_H2S/V_gas_ref)^n_tio2_removal; %[$]

% wet scrubber packed column
FOB_wet_scrubber=35000; % [$]
V_SO2=V_H2S/1000; %[m3/s]
n_wet_scrubber=0.39; % range 0.5-1.65 m3/s
V_scrubber_ref=1.65; % [m3/s]
C_wet_scrubber=FOB_wet_scrubber*(V_SO2/V_scrubber_ref)^n_wet_scrubber;

```

```

Camis=CEPCI_index*(C_HG_removal+C_H2S_removal+C_wet_scrubber); %[$]

% total plant cost
Cplant_PEC=Ctg+Ccondenser+Ccond_pump+Ccooling_tower+Ccirc_pump+Cinj_pump+Ccompr+Camis;
Cplant_TEC=Cplant_PEC*Xse;
Cplant_BEC=Cplant_TEC*(1+Xc1+Xcm+Xst+Xf);
Cplant=Cplant_BEC*XPCp*XICp;

%% Well cost (Cwell)
% success rate in drilling
% indirect and project contingency for well
XICwell=1.05;
XPCwell=1.15;

Lwell_prod=2000;
Lwell_inj=Lwell_prod/2;
Cwell_inj=XICwell*XPCwell*PPIoeg*n_inj*(0.105*Lwell_inj^2+1776*Lwell_inj*d_inj+275300); %[$]
Cwell_prod=XICwell*XPCwell*PPIoeg*n_prod*(0.105*Lwell_prod^2+1776*Lwell_prod*d_prod+275300); %[$]
Cwell=(Cwell_inj+Cwell_prod)/Sucbrate; %[$]

%% Surface Piping system (Csurfacepiping)
% indirect and project contingency for piping
XICpipe=1.12;
XPCpipe=1.15;

dpipe=d_prod;
Lpipe=5000; %find a method to evaluate it approximatly
c_surfacepiping=2205*dpipe^2+134; %[$/m]
Csurfacepiping=XPCpipe*XICpipe*PPIpipe*(c_surfacepiping*Lpipe);

%% Well field cost (Cwellfield)
% indirect and project contingency for permitting
XICwf=1.05;
XPCwf=1.15;

Cpermitting=XICwf*XPCwf*PPIpermit*665700; %[$/site]
Cwellfield=Cpermitting;

%% Exploration cost (Cexploration)
% indirect and project contingency for exploration
XICexpl=1.05;
XPCexpl=1.15;

Cmodeling=XICexpl*XPCexpl*PPIoegs*508000; %[$/site]
Nslim_hole=2;
Lslim_hole=Lwell_prod;
dslim_hole=0.05;
Cslim_hole=XICexpl*XPCexpl*PPIoegs*Nslim_hole*(0.105*Lslim_hole^2+1776*Lslim_hole*dslim_hole+275300); %[$];
Cexploration=Cmodeling+Cslim_hole;

%% Stimulation cost (Cstimulation)
% EGsvalue=input('Is it a EGS system? 1=yes, 0=no: ');
% if EGsvalue==1
%   XICstim=1.05;
%   XPCstim=1.15;
%   Gwells=500; % mass flow rate for stimulation
%   Cstimulation=XICstim*XPCstim*Gwells*715000;
% else
%   Cstimulation=0;
% end
Cstimulation=0;

%% Grid cost (Cgrid)
conversion_euro_dollar=1.12; %[$/€]
Cgrid_investment=80/conversion_euro_dollar*W_gen(1)*1000; % [$]
Lgrid_connection=1000; % [m]
Cgrid_connection=Lgrid_connection*100/conversion_euro_dollar; % [$]
Cgrid=Cgrid_connection+Cgrid_investment;

```

```

%% Operation and maintenance cost (Coem)
% it can be between 0.045 and 0.065, use 0.065 for the worst scenario
Foem=0.05;

%% Greenfield cost (Cgreen) and Brownfield cost (Cbrown)
% fieldtype=input('Type of field. 1=brownfield, 2=greenfield: ');
if fieldtype==1
    Cbrown=Cplant+Csurfacepiping+Cwell_prod+Cgrid;
    Coem_brown=Cbrown*Foem;
else
    Cgreen=Cplant+Csurfacepiping+Cwell+Cwellfield+Cexploration+Cstimulation+Cgrid;
    Coem_green=Cgreen*Foem;
end

time_in_year=8766;

% WACC parameter
% kd=1.85569/100;
% Rf=1.13363/100;
% EMRP=5.7675/100;
% beta_e=1;
% ke=Rf+EMRP*beta_e;
% This equity and debt percentage are related to high risk for Independent
% Power Producer (IPP). For IPP and low risk set equity=0.3 and debt=0.7.
% In case of Investor-Owned Utility (IOU) and low risk set equity 0.5 and
% debt=0.55, for high risk set equity=0.5 and debt)0.45
% equity=0.4;
% debt=0.6;
% taxrate=0.1;
% Wd=debt/(equity+debt);
% We=equity/(equity+debt);
% WACC=Wd*kd*(1-taxrate)+ke*We;
WACC=0.05; %[%]

W_net=[0,W_net];

% LCOE greenfield, brownfield and general
if fieldtype==1
    capex_brown=zeros(1,lifetime+1);
    capex_brown(1)=Cbrown;
    opex_brown=ones(1,lifetime+1)*Coem_brown;
    opex_brown(1)=0;
    varcost_brown=ones(1,lifetime+1)*Coem_brown*0; %check value of variable cost
    varcost_brown(1)=0;
    for i=1:lifetime+1
        netenergy(i)=W_net(i)*CF*time_in_year; %[MWh]
        LCOEnum(i)=(capex_brown(i)+opex_brown(i)+varcost_brown(i))/(1+WACC)^i;
        LCOEden(i)=netenergy(i)/(1+WACC)^i;
    end
    LCOErad=sum(LCOEnum)/sum(LCOEden);
else
    capex_green=zeros(1,lifetime+1);
    capex_green(1)=Cgreen;
    opex_green=ones(1,lifetime+1)*Coem_green;
    opex_green(1)=0;
    varcost_green=ones(1,lifetime+1)*Coem_green*0; %check value of variable cost
    varcost_green(1)=0;
    for i=1:lifetime+1
        netenergy(i)=W_net(i)*CF*time_in_year; %[MWh]
        LCOEnum(i)=(capex_green(i)+opex_green(i)+varcost_green(i))/(1+WACC)^i;
        LCOEden(i)=netenergy(i)/(1+WACC)^i;
    end
    LCOErad=sum(LCOEnum)/sum(LCOEden);
end

%% NPV
% general
revenue_ele=netenergy*1000*eleprice;
revenue_heat=(m_enduse.*h_prod.*eta_enduse)*CF*time_in_year;
IRR=WACC;
year=(0:1:lifetime);
if fieldtype==1
    NPV(1)=-capex_brown(1)*(1-CAPEXincentives);
    for i=2:lifetime+1

```

```

        netrevenue(i)=(((revenue_ele(i)+revenue_heat(i))*(1+REVENUEincentives))-(opex_brown(i)*(1-
OPEXincentives))-varcost_brown(i))/(1+IRR)^year(i);
        NPV(i)=NPV(i-1)+netrevenue(i);
        if sign(NPV(i-1))+sign(NPV(i))==0
            PBT=i-1;
            % fprintf('The pay back time is in the year number %d\n', (i-1)-1);
        end
    end
else
    NPV(1)=-capex_green(1)*(1-CAPEXincentives);
    for i=2:lifetime+1
        netrevenue(i)=((revenue_ele(i)*(1+REVENUEincentives))-(opex_green(i)*(1-OPEXincentives))-
varcost_green(i))/(1+IRR)^year(i);
        NPV(i)=NPV(i-1)+netrevenue(i);
        if sign(NPV(i-1))+sign(NPV(i))==0
            PBT=i-1;
            % fprintf('The pay back time is in the year number %d\n', (i-1)-1);
        end
    end
end
end

```

XIII. Ramey model

```

clear all
close all
clc

load('p3km.mat','p3km');
load('T3km.mat','T3km');
load('rho3km.mat','rho3km');

index_x=size(T3km,1);
index_y=size(T3km,2);

yrs = 1:1:30;
t = 3600*8766*yrs;

index_t=length(yrs);

Beta=1;
K=1e-14;

delta_x=1000; % [m] x-dimension of a cell
delta_y=1000; % [m] y-dimension of a cell
delta_z=100; % [m] z-dimension of a cell
h_intake=100; % [m]
re=0.2*sqrt(delta_x*delta_y);
rw=0.1847/2; % [m]
p_dynamic=0.7*p3km;
S=0;
skin_factor=S-0.75;

for i=1:index_x
    for j=1:index_y
        if p3km(i,j)~=0
            mu_w(i,j)=XSteam('my_pT', p3km(i,j),T3km(i,j));
            cp_w(i,j)=XSteam('cp_pT', p3km(i,j),T3km(i,j))*1000; % [J/kgK]
            tc_w(i,j)=XSteam('tc_pT', p3km(i,j),T3km(i,j));
            q(i,j)=2*pi*K*h_intake/(mu_w(i,j)*Beta*(log(re/rw)+skin_factor))*(p3km(i,j)-p_dynamic(i,j))*1e5;
        end
    end
end

q=repmat(q,1,1,index_t);
mu_w=repmat(mu_w,1,1,index_t);
cp_w=repmat(cp_w,1,1,index_t);
tc_w=repmat(tc_w,1,1,index_t);
rho3km=repmat(rho3km,1,1,index_t);
T3km=repmat(T3km,1,1,index_t);
p3km=repmat(p3km,1,1,index_t);

% Borehole properties
Z = 3000; % depth [m]

```

```

e = 0.0000015; % 0.0000015
D1 = 0.1847;
D2 = 0.1937;
D3 = 0.2137;
R1 = D1/2;
R2 = D2/2;
R3 = D3/2;
delta_pipe = D2 - D1;
delta_grout = D3 - D2;
k_pipe = 45;
k_grout = 1.5;

Ts = 18; % surface temperature [°C]
G = -(T3km-Ts)./Z; % geothermal gradient [°C/m]
rho_r = 2700; % rock density [kg/mc]
cpr = 805; % rock specific heat [J/(kg °C)]
kr = 2.8; % rock thermal conductivity [W/(m °C)]
chir = kr/(rho_r*cpr); % rock thermal diffusivity [m^2/s]

A = pi.*R1.^2;
m_prod_time=q.*rho3km;

Re = (m_prod_time*D1)./(mu_w*A);
Pr = (cp_w.*mu_w)./ tc_w;
fD = (-1.8.*log10((6.9./Re)+(e./D1)./3.7).^1.11).^(-2);
Nu = ((fD./8).*(Re - 1000).*Pr)./(1 + 12.7 .* sqrt(fD./8) .* (Pr.^(2/3) - 1));
h = Nu.*(tc_w/D1);

U = 1 ./ ((R2)./(R1 .* h) + (R2) * (log((R2)/R1)/k_pipe + log(R3/(R2))/k_grout));

t_D = kr .* t ./ (rho_r * cpr * R3^2);

if t_D > 1.5
    ft = (0.4063 + 0.5 .* log(t_D)) .* (1 + 0.6 ./ t_D);
else
    ft = 1.1281 .* sqrt(t_D) .* (1 - 0.3 .* sqrt(t_D));
end

Tcrit=373.9;

for i=1:index_x
    for j=1:index_y
        for k=1:index_t
            if p3km(i,j,k)~=0
                B(i,j,k) = m_prod_time(i,j,k).*cp_w(i,j,k).*(kr + R1.*U(i,j,k).*ft(k)) ./
(2.*pi.*R1.*U(i,j,k).*kr);
                Tout(i,j,k) = G(i,j,k)*Z + Ts - G(i,j,k).*B(i,j,k) + (T3km(i,j,k) + G(i,j,k).*B(i,j,k) -
Ts).*exp(-Z./B(i,j,k)));
                if Tout(i,j,k)>Tcrit
                    rho_sup=XSteam('rhoL_T',Tcrit);
                else
                    rho_sup=XSteam('rhoL_T',Tout(i,j,k));
                end
                p_prod_time(i,j,k)=(p3km(i,j,k)*1e5-rho_sup*9.81*Z)/1e5;
            end
        end
    end
end
T_prod_time=Tout;

```

XIV. Main

```

clear all
close all
clc

%% load matrix from geological model
% the matlab file name must be: "name of the variable"_prod_"depth"km
% the "name of variable" can be: p,T,m
% the variables saved inside the matlab file must be called "p_prod_time,T_prod_time,m_prod_time")
% load('T_prod_2km.mat');
% load('m_prod_2km.mat');
% load('p_prod_2km.mat');

```

```

%% choose the type of plant for which make the analysis
type_of_plant=0;
while type_of_plant==0
    type_of_plant=input('Select the type of plant to simulate:\n1=Dry steam pure electricity\n2=Dry steam top
spillation\n3=Single flash\n4=Binary\nChoice: ');
    % Check if the input is valid (non-empty and a number)
    if isempty(type_of_plant) || ~isnumeric(type_of_plant) || type_of_plant>4
        disp('Invalid input. Please enter a number between 1 and 4.');
```

type_of_plant = 0; % Reset the value to stay in the loop

```

    end
end

%% potential evaluation
if type_of_plant==1
    askinput_dry_steam;
    potential_dry_steam;
else if type_of_plant==2
    askinput_dry_steam_top_spillation;
    potential_dry_steam_top_spillation;
else if type_of_plant==3
    askinput_single_flash;
    potential_single_flash;
else if type_of_plant==4
    askinput_binary;
    potential_binary;
end
end
end
end

%% Figure section
depth=3000/1000;
MW_pot_year30=10;
figure(1)
imagesc(rot90(MW_pot_year30)); % rot90 is here used to have the matrix with the correct orientation for the
case study
hold on
grid on
c = colorbar;
c.Label.String = 'Net Power [MW]';
xlabel('x [km]')
ylabel('y [km]')
title(sprintf('Geothermal Potential for dry steam power plant at %.0f km of depth',depth))

figure(2)
imagesc(rot90(LCOE)); % rot90 is here used to have the matrix with the correct orientation for the case study
hold on
grid on
c = colorbar;
c.Label.String = 'LCOE [$/MWh]';
xlabel('x [km]')
ylabel('y [km]')
title('Levelized Cost Of Energy for dry steam power plant')

figure(3)
imagesc(rot90(NPV_end/1e6)); % rot90 is here used to have the matrix with the correct orientation for the
case study
hold on
grid on
c = colorbar;
c.Label.String = 'NPV [M$]';
xlabel('x [km]')
ylabel('y [km]')
title('Net Present Value after 30 year for dry steam power plant')

%% Performance parameter
fprintf('LCOEmin=%f\n',min(min(LCOE(LCOE~=0))));
fprintf('LCOEmax=%f\n',max(max(LCOE)));
fprintf('SSCmin=%f\n',min(min(SSC_min(SSC_min~=0))));
fprintf('SSCmax=%f\n',max(max(SSC_max)));
fprintf('eta_min=%f\n',min(min(eta_util_min(eta_util_min~=0))));
fprintf('eta_max=%f\n',max(max(eta_util_max)));

```