EMB3Rs

Heat and Cold matching platform

D3.3 - Core Functionalities, Knowledge Base and Geographic Information System (GIS) Module

ANDRE LISBOA, JOSE MARIA CUNHA, MAFALDA SILVA – INEGI
CARLOS MARQUES, LUIS CAMPOS – PDM
ALI KOK – TUWIEN
MARCUS HUMMEL – E-THINK

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement

SPIRE PROJECT

Technical References

Project Acronym	EMB3Rs
Project Title	User-driven Energy-Matching & Business Prospection Tool for Industrial Excess Heat / Cold Reduction, Recovery and Redistribution
Project Coordinator	Mafalda Silva
Project Duration	September 2019 – May 2023 (45 months)

Deliverable No.	3.3. - Core Functionalities Module and Knowledge System
Dissemination level [1]	PU
Work Package	WP3 – Platform Creation - Modules Development, Integration & Lab-scale Validation
Task	T3.1 – Individual Module Development
Lead beneficiary	PDMFC
Contributing beneficiary(ies)	INEGI, e-think, TU Wien
Due date of deliverable	01/02/2023
Actual submission date	26/10/2022

[1] PU = Public
PP = Restricted to other programme participants (including the Commission Services)
RE = Restricted to a group specified by the consortium (including the Commission Services)
CO = Confidential, only for members of the consortium (including the Commission Services)

Document history

V	Date	Beneficiary	Author
1	20/04/2022	INEGI, TUWien, e-think	André Lisboa, José Maria Cunha, Mafalda Silva, Ali Kok, Marcus Hummel
2	25/10/2022	INEGI, TUWien, e-think	André Lisboa, José Maria Cunha, Mafalda Silva, Ali Kok, Marcus Hummel
3			

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

Summary

This deliverable presents the User and System Manuals for the Core Functionalities (CF) module, the Knowledge Base and the Geographic Information System (GIS) module.

The user manual for these modules will guide and help the user understand and navigate in these modules, as part of the integrated platform. The user manual is developed for an average user, and includes examples and infographics to promote the user-friendliness of the platform.

The system manual is designed for the advanced user. It is more comprehensive and includes the full variables description, along with the coding specifications and detailed presentation of all functionalities.

The EMB3RS project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 847121. This module is part of a larger assessment toolbox called the 'EMB3RS platform'.

Disclaimer

Table of Contents

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

1 Introduction

The main aim of this report is to describe the core functionalities (CF) module of the EMB3RS platform along with its user and system manuals. The report consists of 3 main sections. Firstly, an overview of the EMB3RS project is presented in section 1. Section 2 refers to the CF module. It provides the system description of the module in detail, including the structure and functioning of the characterization and simulation routines, which is explained in detail along with its inputs and outputs. It also includes the CF user manual, which provides the user the necessary information to run the routines and obtain the respective the reports. Along similar lines, Section 3 presents the system and user manual of the GIS module.

1.1EMB3RS project

EMB3Rs ("User-driven Energy-Matching & Business Prospection Tool for Industrial Excess Heat Reduction, Recovery and Redistribution") is a European project funded under the H2020 programme (Grant Agreement No.847121). It aims to develop an open-source tool to match potential sources of excess thermal energy with compatible users of heat and cold. For more information about the EMB3RS project, visit the EMB3RS website.

Users, like industries and other sources that produce excess heat, will provide the essential parameters, such as their location and the available excess thermal energy. The EMB3Rs platform will then autonomously and intuitively assess the feasibility of new business scenarios and identify the technical solutions to match these sources with compatible sinks. End-users such as building managers, energy communities or individual consumers will be able to determine the costs and benefits of industrial excess heat utilisation routes and define the requirements for implementing the most promising solutions. The EMB3Rs platform will integrate several analysis modules that will allow a full exploration of the feasible technical routes to the recovery and use of the available excess thermal energy.

Several modules are part of the EMB3RS platform. Each module performs a specific task or analysis of excess heat and cold recovery. The models and their primary functionalities are listed below.

1.1.1 Core functionalities module

The purpose of the CF module is to provide a comprehensive quantification of the energy flows of the EMB3RS platform objects and costs associated with different options for excess heat recovery and use. The other analysis modules (GIS, TEO, MM and BM) to perform simulations according to user specifications use this information. The CF module main functionalities are:

1. Full characterization of objects – e.g., in terms of processes, equipment, building characteristics

2. To carry out a preliminary analysis of available supply and demand heat on sources and sinks, respectively.

3. To carry out an internal heat recovery analysis within the source’s streams.

1.1.2 GIS module

The purpose of the GIS model within EMB3Rs is to analyse possible network solutions for a given set of sources and sinks as well as an assumption of related network heat/cold losses and costs. The GIS thereby finds such a network solution along with the existing Open Street Map (OSM) Road Network connecting all sources and sinks. It currently outputs a graph/map that lets the user check the specifications of every single pipe element from the network found and a table that illustrates all source/sink specific losses, costs, network length and installed pipe capacity.

1.1.3 TEO Module

The TEO module identifies the least-cost combinations of technologies for using and conveying excess heating from defined sources to defined sinks. The user (representing the excess heat producer - i.e., source – or a demand point – i.e., sink) wants to evaluate the least-cost options of utilising excess heat generated to meet the heating/cooling demand for one or more known/assumed sinks. The objective of the optimisation is to find the least-cost mix of technologies (in terms of installed capacities – typically, in power units) and match between sources and sinks (in terms of energy flows) that satisfy the demands under constraints dictated by regulation, availability of heat, load profiles, techno-economic characteristics of technologies, investment plans.

1.1.4 Market Module

The Market Module (MM) will provide the user with economic and fairness indicators like energy transaction, market price, social welfare, and fairness among prices. This will be done by short-term and long-term market analyses that simulate various market structures and incorporate business conditions and network models. The MM will consider the existing Pool market as well as new forms of a decentralized market based on peer-to-peer and community systems. The modelling of heat/cold sources and sinks will include flexibility, offering price and business preferences.

1.1.5 Business Module

Business Model Module evaluates various business models for DHC which incorporate excess heat. This is done by calculating matrices like Net Present Value (NPV), Levelized Cost of Heat (LCOH) and Internal Rate of Return (IRR) under different ownership structures and market frameworks.

2 CORE FUNCTIONALITIES MODULE

2.1 System Manual

2.1.1 Purpose and scope

The purpose of the Core Functionalities (CF) module is to allow for the full characterization of the EMB3Rs platform objects (sinks and sources) and to provide information to all the analysis modules, namely the geographical information systems (GIS) module, the techno-economic (TEO) module, the market module (MM), and the business module (BM); to run their simulations.

The CF module, for both sinks and sources objects, is divided into two main types of routines: characterization and simulation. The characterization routines focus on receiving the user inputs and performing the needed computations to characterize the created objects, e.g., when the user creates a sink object, namely a greenhouse, the CF will compute its hourly heating needs according to its location, dimensions, and other input parameters. The simulation routines focus on performing analysis based on the characterization information, e.g. for a source's excess heat streams (which were computed in the characterization), the conversion simulation will evaluate the available amount of energy that can be provided to a district heating network (DHN).

2.1.2 Main Features of the CF Module

The main features of the CF module according to the object type are:

- **Source:**
    - **Simple characterization: excess heat streams characterization (characterization)**
    - **Detailed characterization: Industry's equipment, processes, and streams characterization (characterization)** – this has been implemented in previous iterations of the platform and kept here for reference, although not used anymore.
    - **Internal heat recovery analysis (simulation)**
    - **Conversion of the source's excess heat streams to the DHN and evaluation of the technologies to be implemented (simulation)**
- **Sink:**
    - **Simple characterization: main circuits (hot water, steam and chilled water) characterization**
    - **Detailed characterization: Industry and buildings – greenhouse, hotel, residential, office - heating/cooling demand and streams characterization (characterization)**
    - **Conversion of the DHN heat to the sink needs and evaluation of the technologies to be implemented (simulation)**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

2.1.3 General module architecture

The general module architecture can be found in Figure 1 and Figure 2. The CF module is developed in Python and organised in several functions to perform the characterization and simulation routines. The characterization routines need inputs from the user and to assess data from the Knowledge Base - KB. The simulation routines, of both sink and sources, utilize the data from the CF characterization and access data from the KB. Only the source's simulation of converting the excess heat streams to the DHN requires additional data from the GIS. A more detailed description of the module architecture, characterization and simulations routines can be found in the following sections.

Looking into more detail at the main platform objects. When a user creates a **source** (Figure 1), there are two methods to perform its characterization. A **simple** method if the user desires to characterize directly specific excess heat streams and a more **detailed** method for users who intend an industry complete characterization. The latter requires the users to introduce in detail their equipment and processes data. In terms of simulation, whether simplified or detailed characterization, the CF module will **convert** the source´s excess heat to the DHN, estimating the available conversion heat and the technologies that could be implemented. Moreover, it is also evaluated the conversion of heat to electricity, by performing the Organic Rankine Cycle (ORC) conversion - **internal heat recovery**. The pinch analysis is performed for users who have done the detailed characterization and users which have all the streams (streams that require heating and cooling) data - **internal heat recovery** - in which the CF suggests possible heat exchanger networks, so that it can be recovered heat within processes.

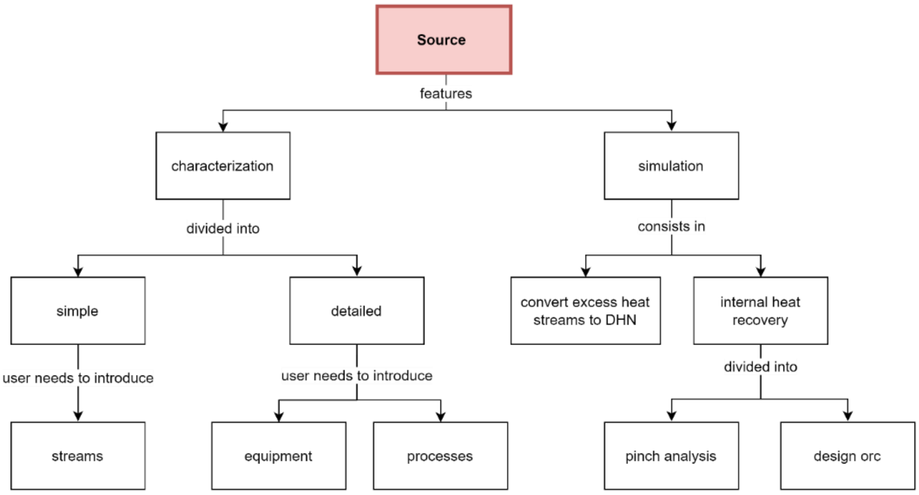


Figure 1 - Overview of the source architecture

When a user creates a **sink** (Figure 2), the user is prompted to characterize its heating/cooling demand. Similar to the source, there is a **simplified** form for the user to input directly a specific heat/cold stream demand, and a more **detailed** form for the users who which to characterize buildings – residential, offices, hotels, and greenhouses. According to the user's buildings specification, the CF will characterize the building by generating the heating/cooling demand. Simulation-wise, the CF will

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

evaluate the technologies that could be implemented on the DHN to meet the heat/cold sink´s needs.

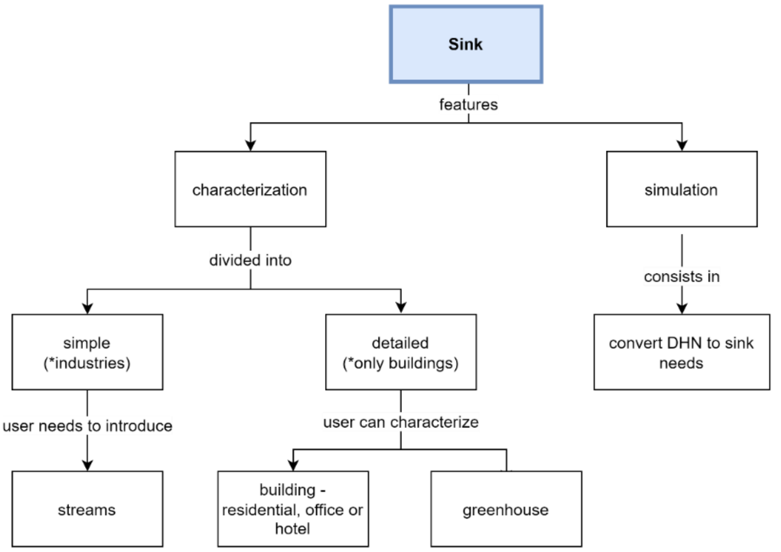
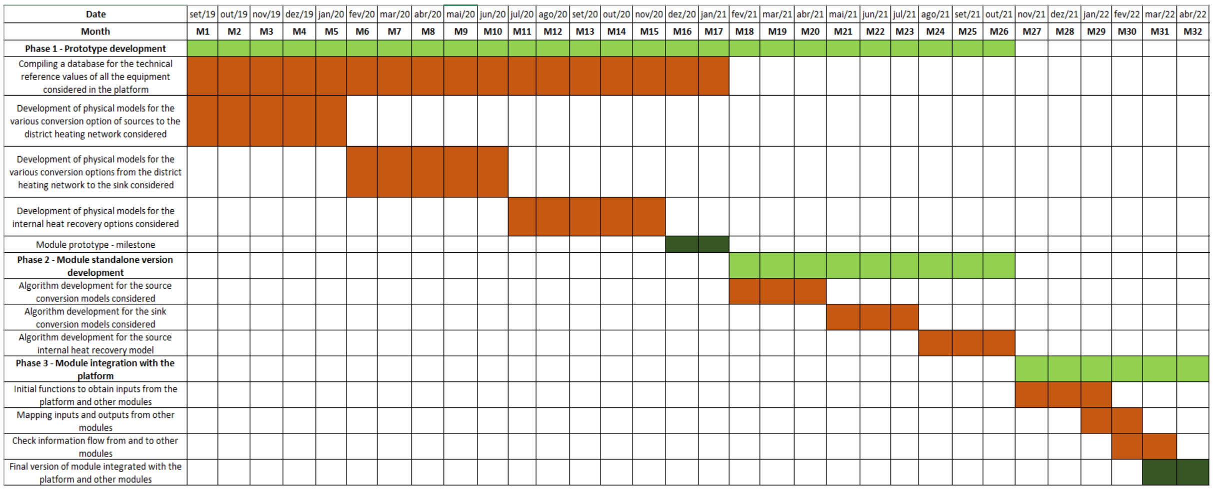


Figure 2 -Overview of the sink architecture

2.1.4 Module development timeline

The CF module has been evolving through different stages over the course of the EMB3RS project. The standalone version was continuously tested on the course of its development, and the integrated version with all the modules was tested on a case study involving an industrial park. The main activities (until M32) are detailed in the figure presented below. After this, the bulk of the work developed focused on supporting the integration on the platform.



Date	set/19	out/19	nov/19	dez/19	jan/20	fev/20	mar/20	abr/20	mai/20	jun/20	jul/20	ago/20	set/20	out/20	nov/20	dez/20	jan/21	fev/21	mar/21	abr/21	mai/21	jun/21	jul/21	ago/21	set/21	out/21	nov/21	dez/21	jan/22	fev/22	mar/22	abr/22
Month	**M1**	**M2**	**M3**	**M4**	**M5**	**M6**	**M7**	**M8**	**M9**	**M10**	**M11**	**M12**	**M13**	**M14**	**M15**	**M16**	**M17**	**M18**	**M19**	**M20**	**M21**	**M22**	**M23**	**M24**	**M25**	**M26**	**M27**	**M28**	**M29**	**M30**	**M31**	**M32**
Phase 1 - Prototype development	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩						
Compiling a database for the technical reference values of all the equipment considered in the platform	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧															
Development of physical models for the various conversion option of sources to the district heating network considered	🟧	🟧	🟧	🟧	🟧																											
Development of physical models for the various conversion options from the district heating network to the sink considered						🟧	🟧	🟧	🟧	🟧																						
Development of physical models for the internal heat recovery options considered											🟧	🟧	🟧	🟧	🟧																	
Module prototype - milestone																■	■															
Phase 2 - Module standalone version development																		🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩						
Algorithm development for the source conversion models considered																		🟧	🟧	🟧												
Algorithm development for the sink conversion models considered																					🟧	🟧	🟧									
Algorithm development for the source internal heat recovery model																								🟧	🟧	🟧						
Phase 3 - Module integration with the platform																											🟩	🟩	🟩	🟩	🟩	🟩
Initial functions to obtain inputs from the platform and other modules																											🟧	🟧	🟧			
Mapping inputs and outputs from other modules																													🟧	🟧		
Check information flow from and to other modules																														🟧	🟧	
Final version of module integrated with the platform and other modules																															■	■

Figure 3 -Module development timeline

All the functionalities developed in the core functionalities module are detailed in the following sections of this report.

2.1.5 Module requirements

The CF module is developed using Python 3.8.0. The packages needed to run the CF module are listed below:

-numpy=1.20.3
-pandas=1.3.4
-geopy=2.20
-pvlib-python=0.9.0
-urllib3=1.26.7
-beautifulsoup4=4.10.0
-matplotlib=3.4.3
-requests=2.26
-mpld3
-pydantic

2.1.6 Module Features

2.1.6.1 Structure of the CF module

The Core Functionalities (CF) module allows for the full characterization of the main objects, e.g., source and sinks, of the EMB3Rs platform which will be the basis for all the calculations and simulations performed in the remaining modules (GIS, TEO, MM, BM). A detailed characterization of the excess heat flows from sources is key for a realistic technical assessment of heat recovery. Moreover, the characterization of the source together with the required heat/cooling flows for the sinks enables the assessment of potential District Heating Networks (DHN) links. The CF module is therefore divided into two main submodules: Sources and Sinks.

2.1.6.2 Pre-Conditions

For the characterization:

- The user must be logged into the platform

For the simulations, in addition to the previous condition:

- The sources/sinks characterization must have been performed

2.1.6.3 KNOWLEDGE BASE: Contributions and requirements

The core functionalities (CF), is the backbone of the EMB3Rs platform, and retrieves from the KB all the default values to present to the user (when these are available) via the user interface during the data input phase. Additionally, all the technical details of the equipment, the building characteristics, fuels characteristics and fluids properties are retrieved from the KB.

a. Equipment

The equipment group comprises all the relevant technical parameters and costing for each equipment family pertaining to the recovery and use of excess thermal energy

and local generation of renewable heat. The data that constitutes this database is being collected from catalogues supplied by various manufacturers and the literature and is meant to be updated yearly. This information will be collected for a reference year (current) as well as how costs and performance are expected to evolve in the future. The current equipment selection was based on the main equipment used by different industries to generate process heating or cooling (boilers, heat pumps, cooling equipment, combined heat and power, solar thermal systems) and the equipment required to convert excess heat streams into useful electricity/thermal energy (heat exchangers, circulation pumps, organic Rankine cycle, and thermal storage).
The correlations implemented are based on the following equation,

$$Value = S + cx^n$$

where, the coefficients S, c, and n, are obtained from Table 1. An example of stored data can be found in Figure 4.

Table 1 - Equipment KB; Parameters coefficients

Equipment KB			
equation	description	parameter	unit
$turnkey = S + cx^n$	equation that provide the equipment turnkey; 1) for heating/cooling technologies (boilers, CHP, and chillers) x is the supply thermal power; 2) for ORC, x is the supplied electrical power; 3) for thermal storage, x is the volume [m3]; 4) for solar thermal, x is the area [m2]; 5) for heat exchangers, x is the area [m2], except for economizer [kW]; 6) for circulation pumping, x is the area [m3/h];	turnkey_cost_S	€
		turnkey_cost_c	€/kW or €/m^2
		turnkey_cost_n	-
$\eta_{global} = S + cx^n$	equation that provide the equipment global efficiency, only for: 1) heating/cooling technologies (boilers, chp, and chillers) x is the supply thermal power; 2) heat exchangers have a fixed value for $S = 0.95$.	global_conversion_efficiency_S	-
		global_conversion_efficiency_c	1/kW
		global_conversion_efficiency_n	-
$\eta_{electrical} = S + cx^n$	equation that provide the equipment electrical efficiency, only for: 1) CHP, where x is the supply thermal power;	electrical_efficiency_c	1/kW
		electrical_efficiency_n	-
$O\&M = S + c * turnkey^n$	equation that provides the equipment O&M fix costs;	fixed_om_c	1/year
		fixed_om_n	-

```
"hot_water_boiler": {
  "equipment": "hot_water_boiler",
  "unit": "kW",
  "turnkey_cost_S": "0",
  "turnkey_cost_c": "1307.2",
  "turnkey_cost_n": "0.432",
  "global_conversion_efficiency_S": "0",
  "global_conversion_efficiency_c": "0.8054",
  "global_conversion_efficiency_n": "0.0215",
  "electrical_efficiency_c": "0",
  "electrical_efficiency_n": "0",
  "fixed_om_c": "1.95",
  "fixed_om_n": "1",
  "co2_emissions_c": "0",
  "co2_emissions_n": "0",
  "pumping_power_c": "0",
  "pumping_power_n": "0"
},
```

Figure 4- Equipment KB example: hot water boiler parameters

To design the heat exchangers according to the type of fluids, the U values which are stored in the KB and presented on Table 2, are used.

Table 2 - U values (W/m^2.K) database stored in the KB for the heat exchangers selected

Heat exchanger type	Fluid 2	Fluid 1		
		steam	liquid	gas
Shell and tubes heat exchanger	steam	0	800	0
	liquid	800	1000	70
	gas	0	70	35
Plate heat exchanger	steam	0	0	0
	liquid	0	2000	50
	gas	0	50	0
Economiser	steam	0	0	0
	liquid	0	0	50
	gas	0	50	0
Waste heat boiler	steam	0	0	50
	liquid	0	0	50
	gas	50	50	0

b. Building

The building properties provide the parameters necessary to run the lumped capacitance thermal simulations to determine the heating and cooling demand of a building according to the climate. These properties vary according to the location of the building, and are stored in the KB as presented in Table 3 and Figure 5.

Table 3 - Building KB; Parameters coefficients

Building KB		
parameter	description	unit
country	country name	-
residential_u_wall	residential buildings wall U value	W/m².K
residential_u_roof	residential buildings roof U value	W/m².K
residential_u_floor	residential buildings floor U value	W/m².K
residential_u_glass	residential buildings glass U value	W/m².K
non-residential_u_wall	Non-residential buildings wall U value	W/m².K
non-residential_u_roof	Non residential buildings roof U value	W/m².K
non-residential_u_floor	Non residential buildings floor U value	W/m².K
non-residential_u_glass	Non residential buildings glass U value	W/m².K
u_exterior	exterior heat transfer coefficient	W/m².K
u_interior	indoor heat transfer coefficient	W/m².K
air_change_hour	air changes per hour due to infiltrations	1/h
alpha_wall	wall’s radiation absorption coefficient	-
alpha_floor	floors’ radiation absorption coefficient	-
alpha_glass	windows’ radiation absorption coefficient	-
tau_glass	glass windows transmissivity	-
average_floor_height	average building floor height	m
capacitance_wall	wall specific heat capacitance	J/m².K
capacitance_roof	roof specific heat capacitance	J/m².K
capacitance_floor	floor specific heat capacitance	J/m².K
capacitance_ground	ground specific heat capacitance	J/m².K
capacitance_glass	glass specific heat capacitance	J/m².K

```
"Belgium": {
  "country": "Belgium",
  "residential_u_wall": "1.12",
  "residential_u_roof": "1.3",
  "residential_u_floor": "0.81",
  "residential_u_glass": "3.58",
  "non-residential_u_wall": "1.38",
  "non-residential_u_roof": "1.33",
  "non-residential_u_floor": "0.8",
  "non-residential_u_glass": "3.66",
  "u_exterior": "20",
  "u_interior": "2",
  "air_change_hour": "0.9",
  "alpha_wall": "0.5",
  "alpha_floor": "0.6",
  "alpha_glass": "0.08",
  "tau_glass": "0.675",
  "average_floor_height": "2.45",
  "capacitance_wall": "16615.5",
  "capacitance_roof": "189120",
  "capacitance_floor": "189120",
  "capacitance_ground": "189120",
  "capacitance_glass": "25200"
},
```

Figure 5 - Building KB example: Belgium buildings' parameters

c. Medium

In the KB, there is the necessary information to characterize all supply and excess heat streams by the different equipment, processes inflows and outflows. The database, as presented in Table 4 and Figure 6, initially stores the relevant thermophysical properties [19] [20] of the most common fluids utilized to transport thermal energy (water, steam, and thermal oil) and others with excess heat recovery potential (flue gas, air, etc). The user can add fluids or materials if they do not exist in the KB. For each fluid/material it is associated with its specific thermal capacity (cp), density (rho), latent heat. phase change temperature from liquid to vapor, and thermal conductivity.

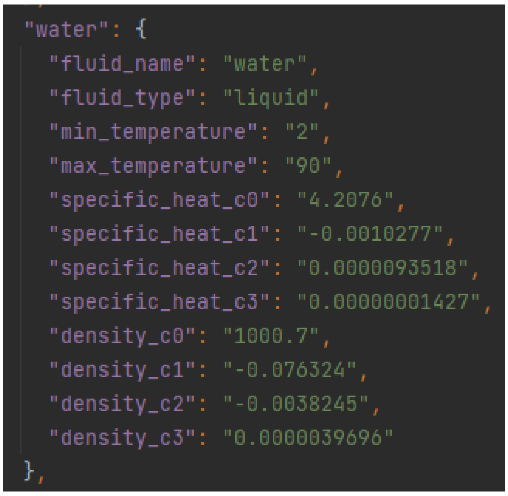
Table 4 - Medium KB; Parameters coefficients

Medium KB	
parameter	unit
fluid_name	-
fluid_type	liquid,steam,gas
min_temperature	ºC
max_temperature	ºC
specific_heat_c0	kJ/kg.K
specific_heat_c1	kJ/kg.K^2
specific_heat_c2	kJ/kg.K^3
specific_heat_c3	kJ/kg.K^4
density_c0	kg/m^3
density_c1	kg/m^3.K
density_c2	kg/m^3.K^2

density_c3	kg/m³.K³

The specific heat capacity and density of the fluids are given by the following equation,

$$Value = c_0 + c_1 * T + c_2 * T^2 + c_3 * T^3$$



```
"water": {
  "fluid_name": "water",
  "fluid_type": "liquid",
  "min_temperature": "2",
  "max_temperature": "90",
  "specific_heat_c0": "4.2076",
  "specific_heat_c1": "-0.0010277",
  "specific_heat_c2": "0.0000093518",
  "specific_heat_c3": "0.00000001427",
  "density_c0": "1000.7",
  "density_c1": "-0.076324",
  "density_c2": "-0.0038245",
  "density_c3": "0.0000039696"
},
```

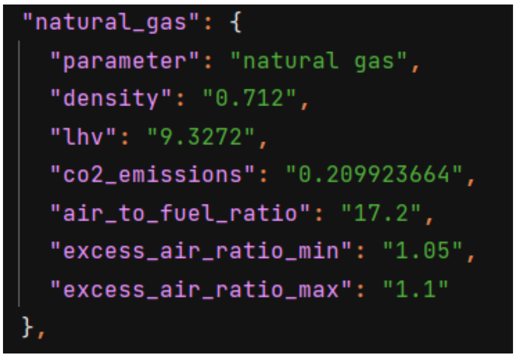
Figure 6 - Medium KB example: water parameters

d. Fuels

The fuels KB provides the necessary information to characterize equipment. e.g. equipment's efficiency. excess heat. estimate fuel/electricity consumption and its cost. and CO_2 emissions. The database has two different files for the fuels information. One file, presented in Table 5 and Figure 7, possesses the country independent parameters for each fuel, such as, its combustion characteristics and CO_2 emissions. The second file, presented in Table 6 and Figure 8, possesses country dependent parameters such as, fuels cost and electricity CO_2 emissions. Additionally, for the electricity and natural gas, the cost is given according to the country by the Eurostat API [20]. The user can add fuels if they do not exist in the KB, and modify the properties of the existent ones.

Table 5 - Fuel Properties KB; Parameters coefficients

Fuel Properties KB	
parameter	unit
density	kg/m³
lhv	kWh/m³
co2_emissions	kg CO_2/kWh
air_to_fuel_ratio	-
excess_air_ratio_min	-
excess_air_ratio_max	-



```
"natural_gas": {
  "parameter": "natural gas",
  "density": "0.712",
  "lhv": "9.3272",
  "co2_emissions": "0.209923664",
  "air_to_fuel_ratio": "17.2",
  "excess_air_ratio_min": "1.05",
  "excess_air_ratio_max": "1.1"
},
```

Figure 7 - Fuel Properties KB: natural gas

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

Table 6 - Fuel Costs and electricity emission per country KB; Parameters coefficients

Fuel Costs and electricity emission per country KB	
parameter	unit
country	-
electricity_emissions	g CO_2/kWh
gasoline_cost	euros/kWh
diesel_cost	euros/kWh
gasoil_cost	euros/kWh
fuel_oil_cost	euros/kWh
biomass_cost	euros/kWh

```
"Portugal": {
  "country": "Portugal",
  "electricity_emissions": "255",
  "gasoline_cost": "0.188451681",
  "diesel_cost": "0.143419946",
  "gasoil_cost": "0.124454323",
  "fuel_oil_cost": "0.065911153",
  "biomass_cost": "0.043199654"
},
```

Figure 8 - Fuel Costs and electricity emission per country KB: Portugal parameters

2.1.6.4 Source

The CF source submodule is divided into: characterization and simulation. The characterization is responsible for receiving the input data from the user, and, by performing several computations, to assess and estimate the available excess heat streams. The simulation aims to evaluate the recovery of the available excess heat internally, either by integrating it within the source's processes – pinch analysis - or by implementing an Organic Rankine Cycle (ORC); and externally, by converting it to the DHN. Figure 9 highlights the main features of the source submodule.

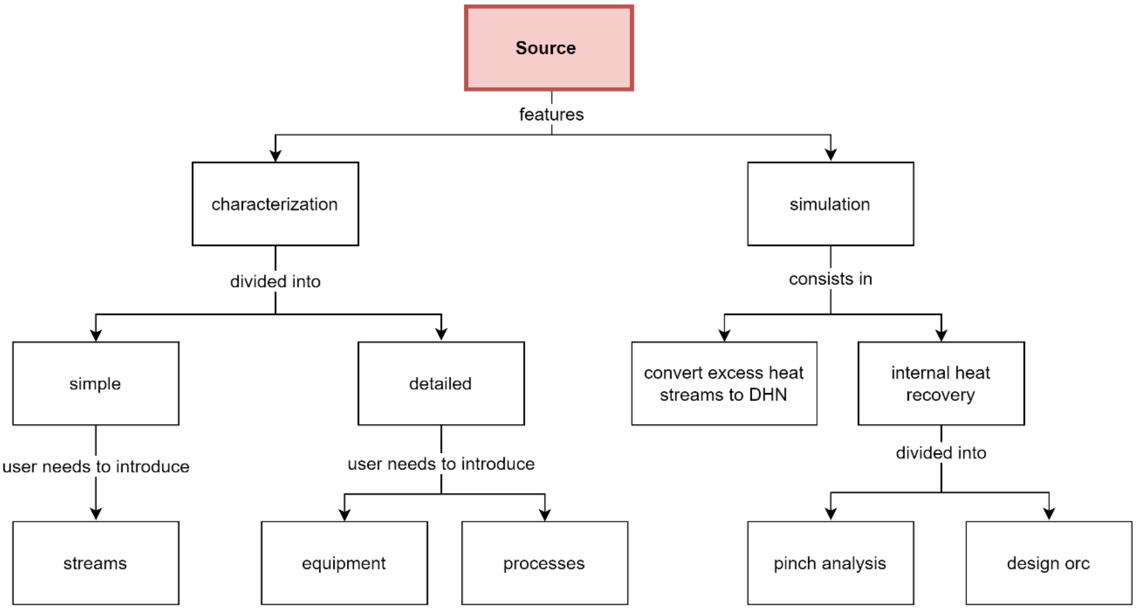


Figure 9- Source submodule overview

2.1.6.4.1 Characterization

The characterization is divided into **simple** - the user just needs to characterize their excess heat streams- and **detailed** - the user must describe all its equipment and processes in detail so that the algorithm can characterize the streams –, as described below. Source objects characterized in both detailed and simplified methods can, later on, perform the excess heat streams conversion to the DHN or ORC design. Only users who choose to carry out a detailed characterization can perform the pinch analysis, as presented in Figure 10.

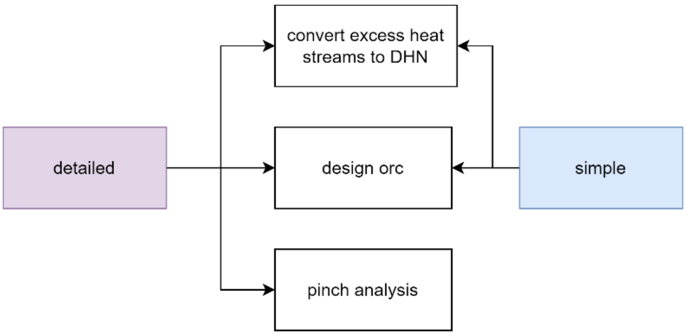


Figure 10 - Source characterization methods and respective simulation options

The characterization submodule has 6 main functions (scripts: *simple user*, *generate_process*, *generate_boiler*, *generate_burner*, *generate_cooling_equipment* and *generate_chp)*, and several auxiliary functions which are fully described in the

Github repository (github https://github.com/Emb3rs-Project/p-core-functionalities). Figure 11 presents a simplified relationship diagram of the characterization functionalities and scripts.

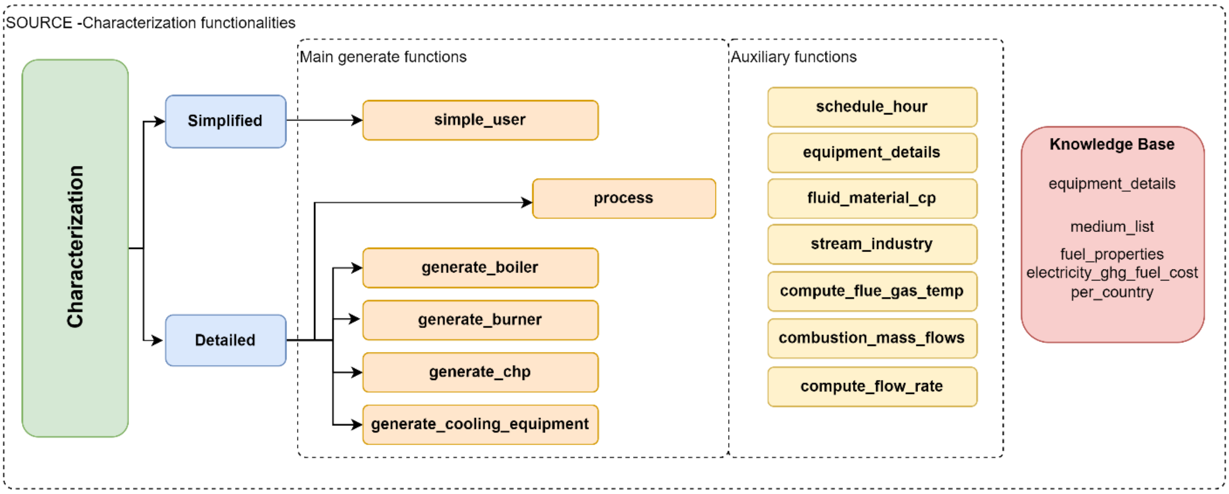


Figure 11 - Schematic relationship diagram of the CF source characterization submodule

a. Simple

At the simple characterization (script: *simple_user*), for each excess heat stream, the user must provide the following data: type of fluid, supply temperature T_{supply}, target temperature T_{target}, flowrate $\dot{m}_{stream}$, specific heat capacity cp_{stream}, and schedule information. As an example, flue gas supplied at 500ºC to be cooled to 20ºC with a flowrate of 3600 kg/h, available all year from 9h to 19h. The algorithm will then calculate for each stream the capacity P_{stream}, and a yearly hourly profile according to the schedule details provided by the user.

$$P_{stream}[kW] = \dot{m}_{stream} * c_{p_{stream}} * \left|T_{target} - T_{supply}\right|$$

The input needed for the simple characterization is presented in Table 7 and Table 8.

Table 7 – Source Simple Characterization input data

Mandatory Input				
var name	description	units	range	data type
type_of_object	if ‘source’ or ‘sink’	-	‘source’; ‘sink’	string
streams	streams data	-	-	dictionary

Table 8 - Source Simple Characterization input data - streams

streams (dictionary data)				
var name	description	units	range	data type
supply_temperature	stream supply/initial temperature	ºC	[0,inf[	float
target_temperature	stream target/final temperature	ºC	[0,inf[	float
fluid	fluid name	-	-	string
fluid_cp	specific heat capacity	kJ/kg.K	[0,inf[	float
flowrate	mass flowrate	kg/h	[0,inf[	float
saturday_on	if working -1- or not -0- on Saturday	-	0;1	integer

sunday_on	if working -1- or not -0- on Sunday	-	0;1	integer
shutdown_periods	array with shutdown periods	days	[0,365]; only integer	array
daily_periods	array with daily periods	hours	[0,24]; only integer	array
ref_system_fuel_type	Fuel type associated	-	natural_gas; electricity; biomass; fuel_oil; none	string
Optional Input				
real_hourly_capacity	Real hourly data - for each hour of the year	kWh	[0,inf[	array
real_daily_capacity	Real daily data - for each day of the year	kWh	[0,inf[	array
real_monthly_capacity	Real monthly data - for each month of the year	kWh	[0,inf[	array
real_yearly_capacity	Real yearly data - single value	kWh	[0,inf[	float

The output from the simple characterization routine, presented in Table 9 and Table 10, is similar to the input since the user is introducing directly the excess heat streams data. Nonetheless, this routine is essential to save the streams data in the platform in a standardized form to be later analyzed by other routines.

Table 9 - Source Simple Characterization output data

Output				
var name	**description**	**units**	**range**	**data type**
streams	streams data	-	-	dictionary

Table 10 – Source Simple Characterization output data - streams

streams (dictionary data)				
var name	**description**	**units**	**range**	**data type**
id	stream ID	-	[0,inf[	integer
object_type	defined as ‘stream’	-	‘stream’	string
object_id	object	-	[0,inf[	string
fluid	fluid name	-	-	string
stream_type	if an ‘inflow’,’outflow’,’supply_heat’,’excess_heat’ stream	-	‘inflow’,’outflow’,’supply_heat’,’excess_heat’	string
schedule	array with stream operating -1- or not -0- each hour	-	-	array
hourly_generation	array with stream hourly available capacity	kW	-	array
capacity	stream capacity	kW	[0,inf[	float
supply_temperature	stream supply/initial temperature	ºC	[0,inf[	float
target_temperature	stream target/final temperature	ºC	[0,inf[	float
monthly_generation	stream's monthly capacity	kWh	[0,inf[	array
fuel	Associated equipment fuel name	-	natural_gas; electricity;	string

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

			biomass; fuel_oil; none	
eff_equipment	Associated equipment efficiency	-	[0,1[	float

b. Detailed[1]

For a detailed characterization, the user must provide the processes and equipment data so that the CF module can obtain the key streams to be assessed. As an example, a cheese factory detailed characterization could be:

- a fermentation process with two streams, an inflow of milk at 20ºC heated until 80ºC, and an outflow of hot whey that is cooled from 80ºC to 5ºC;
- the fermentation process heating needs are supplied by a water boiler with a supply capacity of 3000 kW, running on natural gas.

Both equipment and processes must have a designated schedule. The CF module would obtain that the key streams from this cheese factory are: the fermentation process inflow and outflow streams, the boiler exhaust flue gas, and the supply fluid.

i. Equipment

When the user adds heating or cooling equipment objects, the respective implemented routine runs (see Figure 11), e.g. generate_boiler creates the user's water/steam boiler. The created equipment objects will be saved in the Knowledge Base (KB) with the attributes introduced by the user and estimated by the routines, e.g. equipment efficiency, supply capacity, among others. One of the most important attributes that will be added by the routines is 'streams', which contains the equipment streams as characterized by the user. The 'streams' attribute is the same as presented in Table 10. The heating equipment is characterized by having supply, excess heat, and inflow streams. The cooling equipment with supply and excess heat streams.

1. Heating Equipment

The following heating equipment can be added: boilers (script: *generate_boiler)*, burners (script: *generate_burner)* or Combined Heat and Power - CHP - units (script: *generate_chp)*. Each script aims at characterizing the equipment with key parameters, such as thermal supply capacity, global conversion efficiency, fuel type, and most importantly, the equipment streams data. The mandatory inputs common to all equipment are described in Table 11.

Table 11 – General equipment inputs

Mandatory Input

---

[1] The source detailed characterization has been implemented in past iterations of the platform and kept in the manual for reference, although it is not used. All of the detailed characterization takes place in the background, and is not visible to the user. Along the development process, it was verified that users know their industrial processes very well, and it wouldn't be needed to add such complexity to the simulation. As of M38, the user can add simple sources with multiple streams for waste heat recovery, which addresses the needs identified so far.

var name	description	units	range	data type
id	equipment ID	-	[0,inf[	integer
supply_temperature	working fluid supply temperature	ºC	[0,inf[	float
return_temperature	working fluid return temperature	ºC	[0,inf[	float
saturday_on	if working -1- or not -0- on Saturday	-	0;1	integer
sunday_on	if working -1- or not -0- on Sunday	-	0;1	integer
shutdown_periods	array with shutdown periods	days	[0,365]	array
daily_periods	array with daily periods	hours	[0,24]	array
fuel_type	equipment fuel	-	'natural_gas';'biomass';'fuel_oil';'electricity'	string

Table 12 was set apart from Table 11 to point out that the user can introduce directly the supply capacity of the equipment, if known, or get an indirect estimation of the supply capacity, provided by the routines. For the indirect estimation, the user must associate to the equipment, the processes to which it is providing heat. When processes are associated to an equipment, the yearly heat capacity needed for all process streams is summed and divided by the equipment working hours, to obtain an average equipment's supply capacity.

Table 12 - General equipment inputs – provide just one

Mandatory Input (one of these parameters)				
var name	**description**	**units**	**Range**	**Data type**
supply_capacity	equipment supply capacity	kW	[0,inf[	float
processes	array with processes streams which are provided by the equipment	-	-	array with dictionaries

Table 13 presents the additional boiler inputs. The user should define if the boiler works on open or closed loop. Moreover, the user can designate the boiler's efficiency, otherwise this value is obtained from the KB.

Table 13 – Boiler additional needed inputs

Mandatory Input				
var name	**description**	**units**	**range**	**data type**
open_closed_loop	if equipment works -1-, or not -0- on open loop/working fluid recirculation	-	0;1	integer
Optional Input				
global_conversion_efficiency	equipment efficiency	-	[0,1]	float
boiler_supply_flowrate	Equipment working fluid mass flowrate. Only for steam boilers	[kg/h]	[0,inf[	float

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

equipment_return_temperature	Equipment working fluid return temperature	[ºC]	[0,inf[	[0,inf[

The CHP can be characterized either by its thermal supply capacity or by its electrical generation, and respective conversion efficiencies, as presented in Table 14.

Table 14 – CHP additional needed inputs

Mandatory Input				
var name	**description**	**units**	**range**	**data type**
thermal_conversion_efficiency and supply_capacity	thermal_conversion_efficiency thermal supply capacity	- kW	[0;1] [0,inf[	float float
electrical_conversion_efficiency and electrical_generation	electrical_conversion_efficiency supply electrical capacity	- kWe	[0;1] [0,inf[	float float

The burners are commonly associated to ovens, furnaces and drying processes. It is not feasible to theorethically compute the excess heat recoverable of such equipment. Therefore, for the burner, the user must provide the excess heat streams data, such as supply and target temperatures, and flowrate, to properly characterize the stream (Table 15).

Table 15 - Burner additional needed inputs

Burner - Additional input data		
var name	**description**	**units**
burner_excess_heat_supply_temperature	Excess heat stream supply temperature	ºC
burner_equipment_sub_type	"indirect_burner" or "direct_burner"	-
burner_ excess_heat_flowrate	Excess heat stream mass flowrate	kJ/kg.K

All the input data described above is required so that the equipment streams can be characterized. The streams are namely: inflow stream, e.g. a water boiler requires inflow air for the combustion; excess heat stream, e.g. the flue gas due to combustion.

The inflow and excess heat streams are computed according to the combustion requirements. Some assumptions are considered to characterize these streams (script: *combustion_mass_flows*):

- the inflow stream supply temperature is around ambient temperature (20ºC), and the target temperature of the inflow is 80ºC, which will satisfy the majority of the various heating equipment technical requirements.
- the excess heat stream - flue gas - target temperature is 120ºC, since flue gas is not usually cooled below 120ºC due to the formation of condensates.

The parameters missing to fully characterize the streams, are both streams' mass flowrates, and the excess heat stream supply temperature. The method to estimate these parameters is described below.

First, it is estimated the fuel flowrate,

$$\dot{m}_{fuel}\left[\frac{kg}{h}\right] = \frac{\left(\frac{P_{equip}}{\eta_{equip}}\right)}{LHV_{fuel}}$$

,where $\dot{m}_{fuel}$ is the fuel mass flowrate going into the combustions chamber, P_{equip} is the equipment nominal supply capacity, and LHV_{fuel} is the lower heating value of the fuel.

The air flowrate is obtained by using common fuel properties for a proper combustion,

$$\dot{m}_{air}\left[\frac{kg}{h}\right] = \dot{m}_{fuel} \times AFR_{fuel} \times excess_{air-fuel}$$

,where AFR_{fuel} is the air to fuel ratio for a stoichiometric combustion, and $excess_{air-fuel}$ the excess air commonly applied to guarantee optimum combustion. Default values for these parameters are obtained from the KB.

The inflow stream is then fully characterized with all the parameters. The excess heat stream - flue gas – mass flowrate is simplified as the sum of the fuel and inflow air,

$$\dot{m}_{flue\ gas}\left[\frac{kg}{h}\right] = \dot{m}_{fuel} + \dot{m}_{air}$$

Then, the excess heat stream – flue gas - supply temperature is estimated in two steps. Firstly, an approximate estimate of the combustion chamber temperature is obtained,

$$T_{chamber}[K] = \frac{LHV_{fuel} * \dot{m}_{fuel}}{\dot{m}_{flue\ gas} * c_{p_{flue\ gas}}} + T_{supply_{mix}}$$

where, $T_{supply_{mix}} = 20ºC$, since it is considered that mix of fuel and inflow air is at ambient temperature.

Secondly, the excess heat stream supply temperature is estimated, by the following equation,

$$T_{supply_{flue\ gas}}\ [K] = -\frac{P_{equip}}{\dot{m}_{flue\ gas} * c_{p_{flue\ gas}}} + T_{chamber}$$

By performing these computations, it is possible to obtain an estimate of the excess heat and inflow stream characteristics. The user can modify the estimated values in the platform.

2. Cooling Equipment

When adding a cooling equipment (script: *generate_cooling_equipment*), the user must choose the equipment type (e.g. 'co2_chiller', 'cooling_tower', compression_chiller'), and characterize the equipment parameters, such as thermal supply capacity, global conversion efficiency (in this case, the COP - coefficient of performance), and working schedule. The inputs are similar to the ones presented in Table 11 and Table 12.

For all chillers, the excess heat stream is estimated theoritically. The following correlation is used to compute the excess heat capacity for the chillers, except the CO_2 chiller.

$$P_{excess\ heat} = P_{supply_{chiller}} * \left(1 - \frac{1}{COP}\right)$$

It is assumed by default that the excess heat supply and target temperature of the compression chiller is 45ºC and 35ºC, respectively, and for the cooling tower, 38ºC

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

and 33ºC. According to a representative manufacturer of CO_2 chillers [1] a simple estimate of the excess heat available is as below.

$$P_{excess\ heat} = P_{supply_{CO_2\ chiller}} * 1.5203$$

It is assumed that the excess heat supply and target temperature as, 90ºC and 60ºC respectively.The excess heat stream fluid is, by default, water.

ii. Process

The user can add multiple process objects for the same industry. For the processes script (script: *process*), the user must always provide the process schedule, the schedule type (e.g. batch or continuous), the operation temperature, and characterize the four types of streams possible: startup, maintenance, inflow, and outflow. There can be multiple streams of each stream type. For the startup, the user has to characterize the fluid, initial temperature, and mass. For both inflow and outflow the flow rate and the fluid type are required, as well as the supply temperature for the inflow and target temperature for the outflow streams. For the maintenance stream, the capacity needed to maintain a process at a certain temperature is requested. The streams are then characterized, as presented in Table 10 and Table 10, and added as an attribute to the object. Similarly to the equipment object, ‘streams’ is the most important attribute of the process object, since it will be used in other routines.

Table 16 – Process input data

Mandatory Input				
var name	**description**	**units**	**range**	**data type**
id	process ID	-	[0,inf[	integer
equipment	heat/cooling equipment ID the process is associated to	-	[0,inf[	integer
operation_temperature	process operation temperature	ºC	[0,inf[	float
saturday_on	if working -1- or not -0- on Saturday	-	0;1	integer
sunday_on	if working -1- or not -0- on Sunday -	-	0;1	integer
shutdown_periods	array with shutdown periods	days	[0,365]	array
daily_periods	array with daily periods	hours	[0,24]	array
schedule_type	0=continuous, 1=batch	-	0;1	integer
cycle_time_percentage	batch production time ratio for the startup	-	]0,1[	float
startup_data	array with dictionaries with startup streams characteristics	-	-	dictionary
maintenance_data	array with dictionaries with maintenance/evaporation streams characteristics	-	-	dictionary
inflow_data	array with dictionaries with inflow streams characteristics	-	-	dictionary
outflow_data	array with dictionaries with outflow streams characteristics	-	-	dictionary

Table 17 – Startup stream additional needed inputs

startup_data				
var name	**description**	**units**	**range**	**data type**
name	Name	-	-	string
fluid	fluid name	-	-	string
initial_temperature	medium initial temperature	[ºC]	[0,inf[	float

mass	medium mass	[kg]	[0,inf[	float

Table 18 - Maintenance stream additional needed inputs

maintenance_data				
var name	**description**	**units**	**range**	**data type**
name	Name	**-**	**-**	string
capacity	capacity given to a process to compensate for the thermal losses	kW	[0,inf[	float

Table 19 - Inflow stream additional needed inputs

inflow_data				
var name	**description**	**units**	**range**	**data type**
name	Name	-	-	string
fluid	fluid name	-	-	string
supply_temperature	stream supply temperature	ºC	[0,inf[	float
flowrate	stream flowrate	kg/h	[0,inf[	float

Table 20 - Outflow stream additional needed inputs

outflow_data				
var name	**description**	**units**	**range**	**data type**
name	Name	-	-	string
fluid	fluid name	-	-	string
target_temperature	stream target temperature	ºC	[0,inf[	float
flowrate	stream flowrate	kg/h	[0,inf[	float
Optional				
initial_temperature	Outflow initIal temperature	[ºC]	[0,inf[	float

2.1.6.4.2 Simulation

After performing the source characterization, the user is able to estimate the heat recovery potential for industrial sources. The simulation module has two main submodules: internal heat recovery analysis *(main scripts: convert_pinch and convert_orc)* and converting source's excess heat streams to the DHN *(main script: convert_sources)*. The main functions (presented in Figure 12) and all the auxiliary functions associated to the modules are fully documented in the Github repository (https://github.com/Emb3rs-Project/p-core-functionalities.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

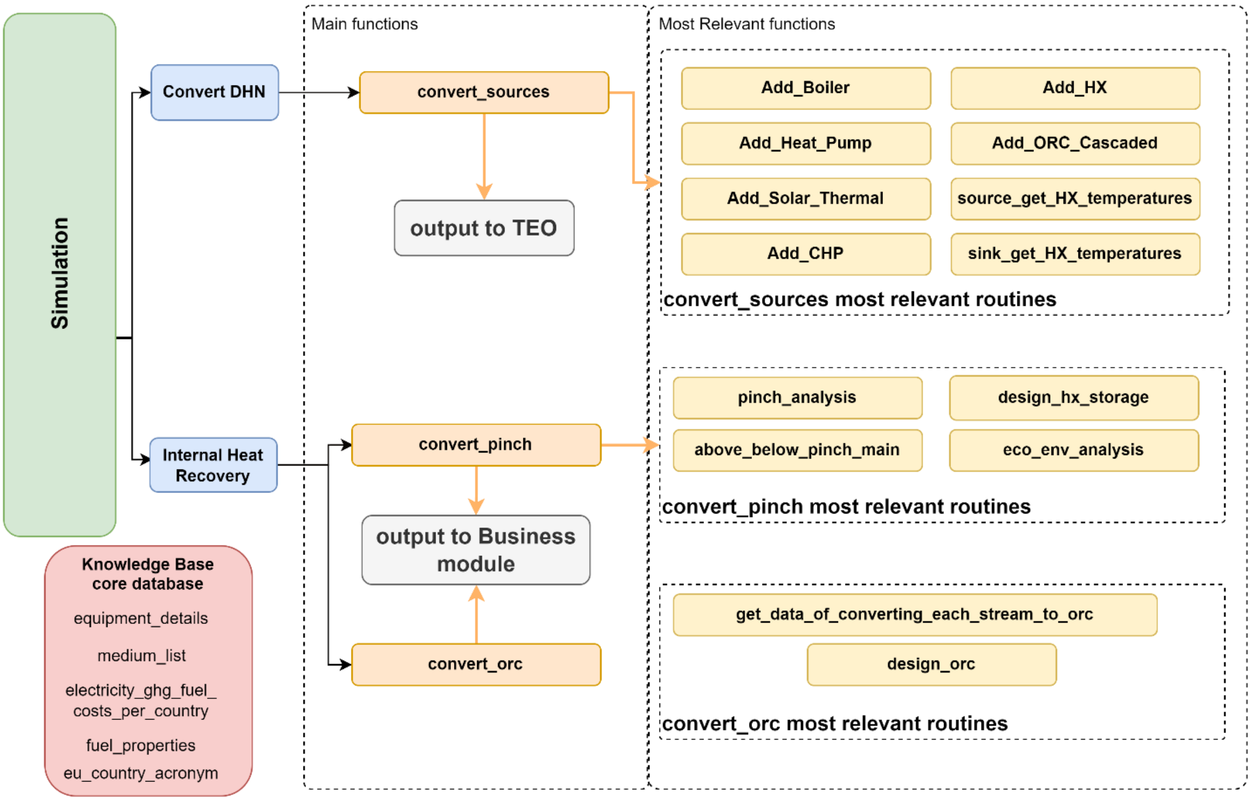


Figure 12 - Schematic relationship diagram of the CF source simulation submodule

a. Internal Heat Recovery

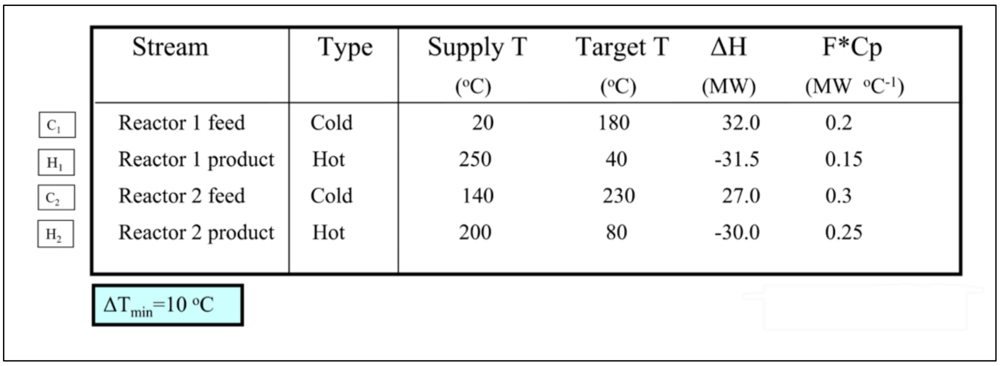
The internal heat recovery consists in recovering the heat within the industry by designing a network of heat exchangers – applying the pinch analysis - that promotes the exchange of heat within the processes, thus, minimizing heat/cooling supply by external equipment; by utilizing the excess heat to produce electricity – implementing an Organic Rankine Cycle (ORC).

i. Pinch Analysis

In brief, the pinch analysis is a theoretical method that, based on fundamental thermodynamics, analyzes the heat flow through the industry's processes with the aim of recovering heat within those processes by creating a heat exchanger - HX - network, reducing energy needs and CO_2 emissions (for more information see [2] [3]).

1. Theory

In order to provide a clear understanding of the pinch analysis, it is analyzed one example case [4]. Figure 13 presents the necessary data to perform the pinch analysis for the example case. The mandatory parameters are the streams' supply and target temperature, and heat capacity (commonly as $\dot{m}cp$, on the example nomenclature it is $F * Cp$). In this example, there are two hot streams ($T_{supply} > T_{target}$) and cold streams ($T_{target} > T_{supply}$) with a default $\Delta T_{min} = 10ºC$. The ΔT_{min} , which is the minimum heat exchanger allowed temperature difference, is set by the designer and defined according to cost and streams characteristic restrictions.



	Stream	Type	Supply T (ºC)	Target T (ºC)	ΔH (MW)	F*Cp (MW ºC^{-1})
C_1	Reactor 1 feed	Cold	20	180	32.0	0.2
H_1	Reactor 1 product	Hot	250	40	-31.5	0.15
C_2	Reactor 2 feed	Cold	140	230	27.0	0.3
H_2	Reactor 2 product	Hot	200	80	-30.0	0.25

ΔT_{min}=10 ºC

Figure 13 – Pinch analysis problem example [4]

After collecting the streams data, the heat cascade is computed [2]. The heat cascade is a table of the net heat flow from high to low temperatures divided up into temperature intervals. To ensure that within any interval hot and cold streams are at least ΔT_{min} apart, shifted temperatures are utilized. These are set at $\frac{1}{2}\Delta T_{min}$ (5°C in this example) below hot stream temperatures and $\frac{1}{2}\Delta T_{min}$ above cold stream temperatures. Looking at Figure 13, the shifted temperatures of the hot stream "Reactor 1 product" are $T_{\mathrm{supply}_{\mathrm{shifted}}} = 245^{\circ}C$ and $T_{\mathrm{return}_{\mathrm{shifted}}} = 35^{\circ}C$. Setting up the intervals with shifted temperatures guarantees that full heat inter-change within any interval is possible.

With all the temperatures shifted, of both hot and cold streams, it is possible to arrange the heat cascade table, as presented in Table 21. For each temperature interval it is computed the heat flow – dH- and evalauted if the heat cascade is feasible or not. The heat cascade is only thermodynamically feasible if it does not present negative heat flows.

Table 21 – Heat Cascade; First approach

Shift Temperature	**Interval**	**$T_{(i+1)}$-T_i**	**mCp$_{net}$**	**dH**	**Cascade**
°C		°C	MJ/s/K	MJ/s	**MW**
245					**0,0**
	1	10	0,15	1,5	
235					**1,5**
	2	40	-0,15	-6,0	
195					**-4,5**
	3	10	0,1	1,0	
185					**-3,5**
	4	40	-0,1	-4,0	
145					**-7,5**
	5	70	0,2	14,0	
75					**6,5**
	6	40	-0,05	-2,0	
35					**4,5**
	7	5	-0,4	-2,0	
25					**2,5**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

Looking at Table 21 , the heat cascade is not feasible since there are negative values. It is necessary to add a hot utility with equal power to the most negative heat flow (in this example, 7.5 MW) so that the heat cascade becomes feasible. A new and feasible heat cascade is presented in

Table 22.

Table 22 - Heat Cascade; Second approach

Shift Temperature	Interval	$T_{(i+1)}-T_i$	mCp_{net}	dH	Cascade
°C		°C	MJ/s/K	MJ/s	MW
245					**7,5**
	1	10	0,15	1,5	
235					**9,0**
	2	40	-0,15	-6,0	
195					**3,0**
	3	10	0,1	1,0	
185					**4,0**
	4	40	-0,1	-4,0	
145					**0,0**
	5	70	0,2	14,0	
75					**14,0**
	6	40	-0,05	-2,0	
35					**12,0**
	7	5	-0,4	-2,0	
25					**10,0**

Looking at
Table 22, it is possible to recognize the pinch at 145ºC, where the net heat flow is equal to 0, meaning there is no heat flow across this temperature. Moreover, from the heat table cascade, it is possible to obtain the minimum theoretical hot and cold utilities, in this case, 7.5 (red rectangle) and 10 MW (blue rectangle), respectively. At this point, the problem can be divided into two separate problems, above and below the pinch, as presented in Figure 14. Above the pinch, only hot utilities are required, and below, only cold utilities. The pinch temperature for hot streams will be the pinch temperature plus $\frac{1}{2}\Delta T_{min}$,and for the cold streams the pinch temperature minus $\frac{1}{2}\Delta T_{min}$.

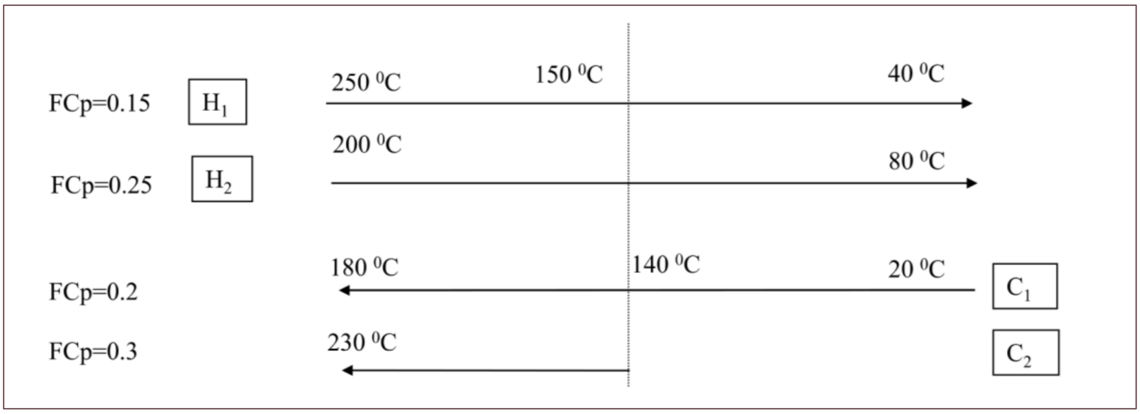


Figure 14 – Example problem divided into two sections: above and below pinch temperature [4]

Now, it is possible to design the various heat exchangers in order to obtain the optimal design solution. The following pinch design rules must be taken into consideration to guarantee maximum heat recovery :

- **Do not transfer heat across the pinch;**
- **Do not use cold utilities above the pinch.**
- **Do not use hot utilities below the pinch**

To perform the heat exchangers designs/matches, there is not a straightforward procedure that guarantees that by following specific rules an optimal design solution is immediately obtained. Some solutions may provide an excessive number of heat exchangers, others will not match the minimum required hot/cold utility and may need larger utilities. Thus, multiple designs should be considered and the designer should evaluate which solution is the optimal. Figure 15 presents a suggested algorithm from the literature [2], which aims to help the designer reach a pinch design.

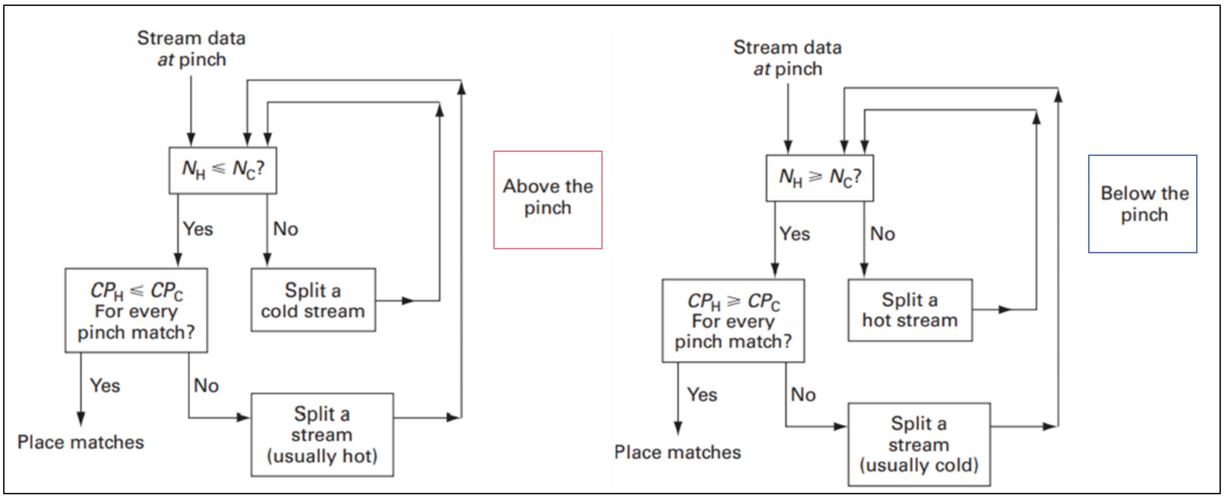
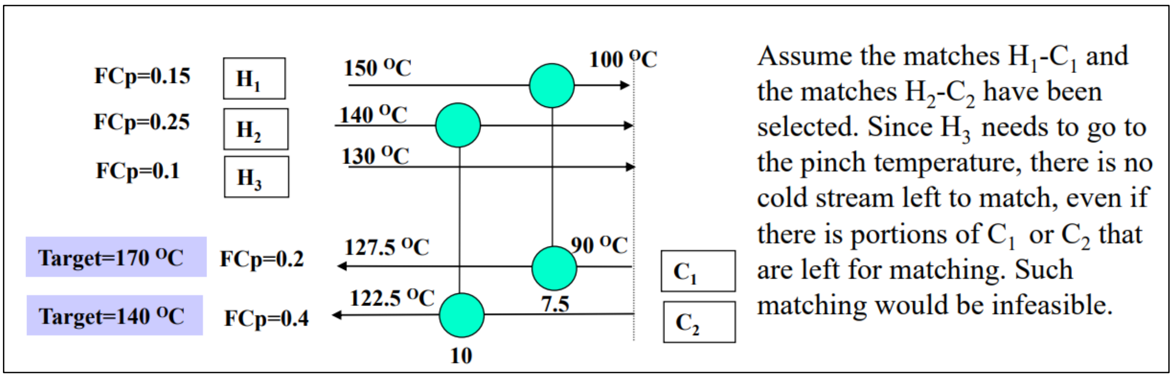


Figure 15 – Kemp Algorithm [2]

It is useful looking at some of the rules presented in Figure 15, before proceeding with the example that was being solved. After dividing the problem in above and below pinch, the designer should check if the number of streams going into the pinch – streams reaching pinch temperature, e.g. H_1, H_2, H_3 in Figure 16 - is smaller or equal to the number of streams going out of the pinch – streams leaving the pinch temperature, e.g. C_1 and C_2. If this requirement is not met, a stream going out of the pinch should be split, as presented in Figure 17Figure 17.



Assume the matches H_1-C_1 and the matches H_2-C_2 have been selected. Since H_3 needs to go to the pinch temperature, there is no cold stream left to match, even if there is portions of C_1 or C_2 that are left for matching. Such matching would be infeasible.

Figure 16 – Example with number of streams going into the pinch larger than the ones going out [4]

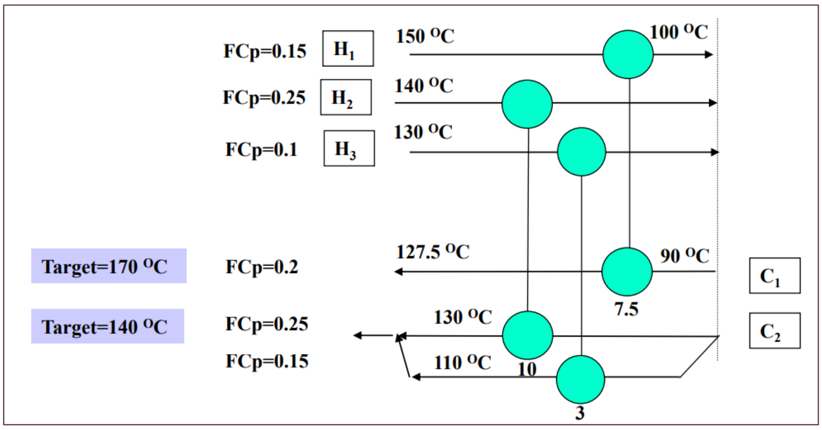


Figure 17 - Example of possible stream split [4]

If the number of streams rule is fulfilled, a cold and hot stream should be picked to match. All matches between streams must fulfil the mcp criteria, which states that, above the pinch, $mc_{p_{hot\,stream}} \geq mc_{p_{cold\,stream}}$, and, below the pinch, $mc_{p_{hot\,stream}} \leq mc_{p_{cold\,stream}}$. If not met, a split should be done to the stream going out of the pinch.

If it is met, the match can be done and the heat exchanger designed. The procedure should be repeated for the remaining streams, above and below the pinch.
Returning to the example, after following the procedure it is possible obtain the design solution presented in Figure 18 (for the full resolution please check [4]). For this simple example, we can obtain an optimal design where 5 heat exchangers are needed, and it is only needed to add the estimated theoretical minimum hot and cold utilities requirements.

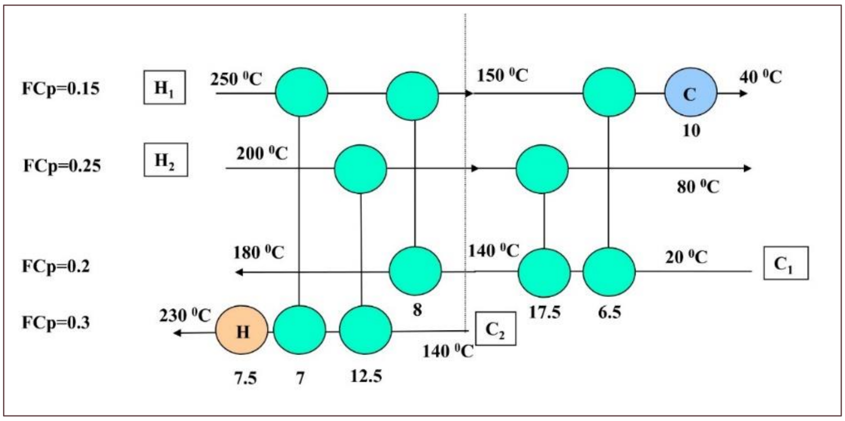


Figure 18 - Final design solution [4]

2. Implemented Routine

The pinch analysis submodule (main script: *convert_pinch)* assesses heat recovery alternatives within the industry by performing a series of pinch analysis to the process streams added by the user. The output of this routine is the 3 best pinch designs in terms of heat recovery (MW), lowest cost of heat (€/MWh) and highest CO_2 emission reductions (kgCO_2/year).

As it was observed in the example of the previous section, the pinch analysis is not a straightforward task and becomes more complex as more streams are added. To overcome various algorithm obstacles, a procedure that follows the pinch rules, with some simplifications, was implemented. Below you will find a step by step structure of the implemented pinch analysis algorithm, tested in different literature examples, that attempts to provide different design solutions.
This module runs when isolated streams (streams without equipment associated), equipment and processes streams, or only equipment's streams are provided.

Analyzing the options:
- When processes and equipment are provided, the heat recovery between processes is analyzed. The output is the three best solutions for the heat recovery, lowest cost of heat, and reduction of CO_2 emissions.
- When only an equipment is provided, its internal heat recovery (e.g. on a boiler, flue gas on the inflow air for the combustion) is analyzed. The output is only one solution, since it designs one heat exchanger for the equipment internal heat recovery – excess heat on supply fluid.
- When only isolated streams are provided, its heat recovery is analyzed. The output is the best solutions for the lowest cost of heat and maximum heat recovery. The analysis of minimum CO_2 emissions is only analyzed if these streams are associated to an equipment, otherwise CO_2 savings cannot be computed.

All the inputs required for the pinch analysis routine are described in Table 23. The parameter 'all_input_objects' refers to the streams that are going to be analyzed. The attribute 'streams', created in the characterization, for all process and equipment objects should be provided to this routine.

Table 23 - Convert_pinch inputs

Mandatory Input				
var name	**description**	**units**	**range**	**data type**
pinch_delta_T_min	delta temperature for pinch analysis	ºC	[0,inf[	float
all_input_objects	array with equipments/processes/isolated_stream dictionaries	-	-	array with dicts
fuels_data	Fuels price [€/kWh] and CO2 emission [kg CO2/kWh] for: natural gas, fuel oil, electricity, biomass.	-	-	dict
Optional input				
lifetime	technologies lifetime for economic analysis. (default: lifetime=20 years)	year	[0,inf[	integer
number_output_options	number of optimal output design options for each of the analysis (default: number_output_options=3)	-	[1, max_solutions]	integer
interest_rate	Interest rate considered for BM (default:0.04)		]0,1]	float

Table 24 - Equipment attribute required for convert_pinch

all_input_objects - equipment

var name	description	units	range	data type
id	equipment ID	-	[0,inf[	integer
object_type	type of object	-	“equipment”	string
streams	equipment streams (see section 0)	-	-	array with dicts
fuel_type	equipment fuel type	-	“natural_gas”,”electricity”,”biomass”,”fuel_oil”	string
global_conversion_efficiency	Equipment efficiency	-	]0,1[	float

Table 25 - Process attribute required for convert_pinch

all_input_objects - process				
var name	**description**	**units**	**range**	**data type**
id	process ID	-	[0,inf[	integer
object_type	type of object	-	“process”	string
equipment	equipment ID the process is associated to	-	[0,inf[	integer
streams	process streams	-	-	array with dicts

The output obtained from this routine, as mentioned before, are the best pinch designs - heat exchangers networks - in terms of the following optimization parameters: heat recovery (MWh), lowest cost of heat (€/MWh) and highest CO_2 emission reductions (kg CO_2/year).

Table 26 - Pinch output best_options categories

Output					
var name		**description**	**units**	**range**	**data type**
best_options	co2_optimization	array with dictionaries for the best designs with maximum CO_2 emissions savings	-	-	array
	energy_recovered_optimization	array with dictionaries for the best designs in maximum of energy recovered	-	-	array
	energy_investment_optimization	array with dictionaries for the best designs in max energy_recovered over turnkey	-	-	array
report		HTML report with Pinch Analysis report	-	-	string

For each of the optimization dictionaries, the best *n* pinch designs (depending on *number_output_options* given in Table 23) data, as presented in Table 27, is provided.

Table 27 - Pinch output - design attributes

dictionaries : co2_optimization/ energy_recovered_optimization/ energy_investment_optimization				
var name	**description**	**units**	**range**	**data type**
ID	designed solution ID	-	[0,inf[	integer
streams	streams in pinch design	-	-	array
capex	design turnkey	€	[0,inf[	float

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

om_fix	yearly O&M fixed costs	€/kW	[0,inf[	float
hot_utility	power of the hot utility needed	kW	[0,inf[	float
cold_utility	power of the cold utility needed	kW	[0,inf[	float
lifetime	considered lifetime	year	[0,inf[	integer
co2_savings	annualized CO$_2$ savings by implementing the pinch design	kgCO$_2$/kWh	[0,inf[	float
money_savings	annualized energy savings by implementing the pinch design	€/kWh	[0,inf[	float
energy_dispatch	yearly energy recovered by implementing the pinch design	kWh/year	[0,inf[	float
discount_rate	discount rate to be applied on the business analysis	-	[0,1]	float
pinch_temperature	design pinch temperature	ºC	[0,inf[	float
theo_minimum_hot_utility	theoretical power of the hot utility needed	kW	[0,inf[	float
theo_minimum_cold_utility	theoretical power of the cold utility needed	kW	[0,inf[	float
equipment_detailed_savings	array with dictionaries of each equipment savings when implementing the pinch design	-	-	array
pinch_hx_data	array with dictionaries of designed heat exchangers	-	-	array

There are two additional arrays, in

Table 27, that must be analyzed: *pinch_hx_data* and *equipment_detailed_savings*. The *pinch_hx_data* is an array which contains as many dictionaries as heat exchangers in the final pinch design solution. Each heat exchanger in this array possesses the attributes that are presented in Table 28.

Table 28- Pinch output - pinch data attributes

pinch_hx_data - example of a dictionary				
var name	**description**	**units**	**range**	**data type**
HX_Power	heat exchanger design power	kW	[0,inf[	float
HX_Hot_Stream	hot stream ID	-	[0,inf[	integer
HX_Cold_Stream	cold stream ID	-	[0,inf[	integer
HX_Original_Hot_Stream	original hot stream ID (it can be different of HX_Hot_Stream if a stream split occurs - a new id is given to the split)	-	[0,inf[	integer
HX_Original_Cold_Stream	original cold stream ID (it can be different of	-	[0,inf[	integer

	HX_Cold_Stream if a stream split occurs - a new id is given to the split)			
HX_Cold_Stream_flowrate	mass flowrate	kg/h	[0,inf[	float
HX_Hot_Stream_flowrate	mass flowrate	kg/h	[0,inf[	float
HX_Type	type of heat exchanger	-	"hx_plate", "hx_economizer"	string
HX_Turnkey_Cost	heat exchanger turnkey cost	€	[0,inf[	float
HX_OM_Fix_Cost	heat exchanger O&M cost	€/kW	[0,inf[	float
HX_Hot_Stream_T_Hot	hot stream hot temperature	ºC	[0,inf[	float
HX_Hot_Stream_T_Cold	hot stream cold temperature	ºC	[0,inf[	float
HX_Cold_Stream_T_Hot	cold stream hot temperature	ºC	[0,inf[	float
HX_Cold_Stream_T_Cold	cold stream cold temperature	ºC	[0,inf[	float
Storage	storage volume	m^3	[0,inf[	float
Storage_Turnkey_Cost	storage turnkey cost	€	[0,inf[	float
Total_Turnkey_Cost	whole package (heat exchanger + storage) turnkey cost	€	[0,inf[	float
Recovered_Energy	amount of yearly energy recovered	kWh	[0,inf[	float

The equipment_detailed_savings is an array that contains as attributes the energy, CO_2 emissions and monetary savings, as presented in the Table 29.

Table 29 - Pinch output - equipment details attributes

equipment_detailed_savings - example of a dictionary				
var name	**description**	**units**	**range**	**data type**
Equipment_ID	equipmet ID	-	[0,inf[	integer
CO2_Savings_Year	yearly CO_2 emissions saved	kg CO_2	[0,inf[	Float
Recovered_Energy	yearly energy saved	kWh	[0,inf[	Float
Savings_Year	yearly monetary savings	€	[0,inf[	float

3. Step by Step

The implemented routine, ***convert_pinch***, aims to generate different design combinations to increase the number of pinch designs, by trying to obtain various streams match possibilities, and thus provide the best solutions. This function designs various possible match combinations between hot and cold streams, above and below the pinch temperature, respecting the pinch rules. The pinch rules are implemented in the algorithm, with the exception that the mc_p criteria rule is not only applied at step 6 (see below). The matches are always done with the purpose of designing away from the pinch, meaning that the heat exchangers are designed from closer to the pinch temperature outwards.

In order to be able to provide a solution to the majority of the cases and different design solutions, the following methodology was implemented:

1. **Establish all possible combinations between hot and cold streams**
    a. **pick the streams combinations one by one**
    b. **get streams combination pinch temperature**
    c. **above and below pinch temperature analysis**
        i. **check if special cases exist**
        ii. **check if the number of $streams_{out} < streams_{in}$**
        iii. **perform first match of streams going into the pinch**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

   iv. **check if the number $streams_{out} < streams_{in}$**
   v. **match remaining streams according to power - without stream splits, and with the mc_p criteria**
   vi. **match remaining streams according to power - without stream splits, and without the mc_p criteria**
  d. **Design storage for each designed heat exchanger**
  e. **Make combinations between above and below pinch designs**
  f. **Return to step a) until all stream combinations are analyzed**

2. **Perform an economic and environmental analysis to all pinch designs**
3. **Return the three best heat recovery (kW), lowest cost of heat (€/kWh) and larger CO_2 emission reductions (kg CO_2) options.**

The following sections describe in detail the steps regarding the design of the heat exchangers network for a group of streams.

1. Establish all possible combinations between hot and cold streams

Make all possible combinations between cold and hot streams. Combinations which only possess hot or cold streams are automatically discarded.

a) pick the streams combinations one by one

At the beginning, one of combinations from all the possible streams combinations of streams is picked to be analyzed.

b) get streams combination pinch temperature

According to the streams present in the combination, the respective heat cascade and pinch temperature are found, as described in section 0.

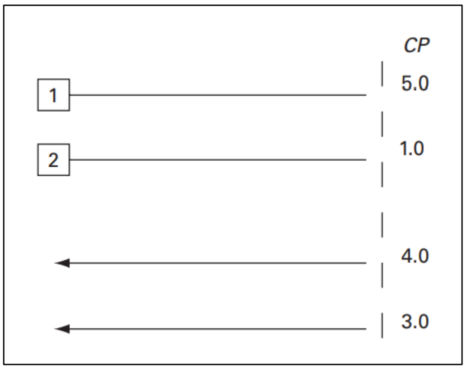
c. i) check if special cases exist

When testing the pinch algorithm, two special cases were found not to be correctly solved, leading to no pinch design solutions. Thus, specific scripts were coded to solve them. These cases occurred when the number of streams going in and out of the pinch was equal. Due to the streams characteristics it was not possible to design heat exchangers that would fulfil all streams, thus leading to no network design.

More specifically, these special cases were:

Case 1) when there is one $stream_{in}$ with larger mc_p than all $streams_{out}$ (see Figure 19)

Case 2) when there is a $stream_{out}$ with smaller mc_p than all $streams_{in}$ (see Figure 20)



This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

Figure 19 - Case 1 example [2]

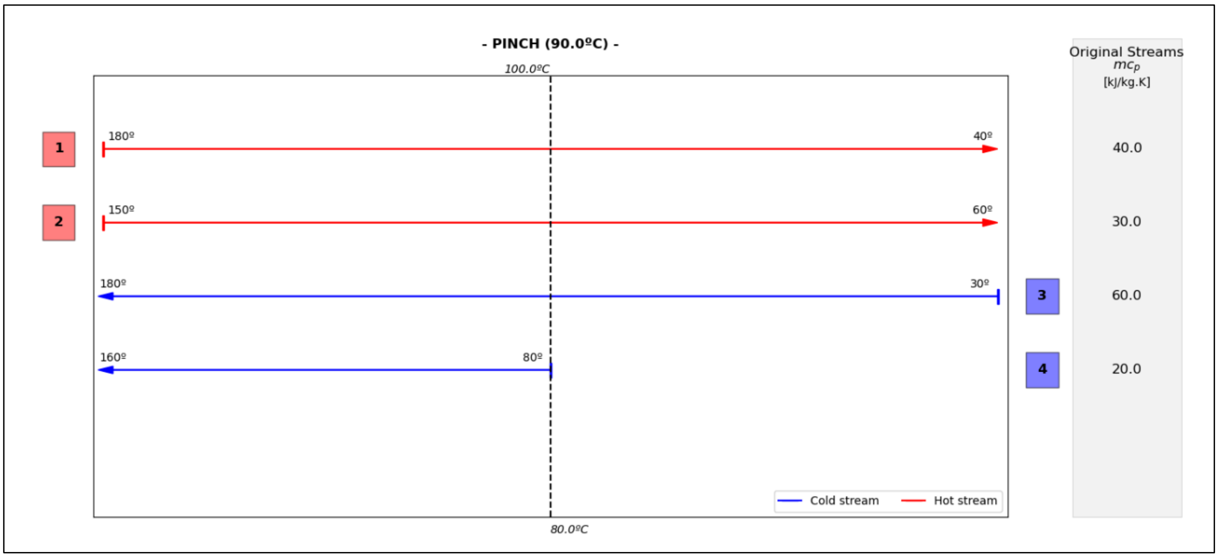


Figure 20 - Case 2 example [5]

To solve these special cases, the routines below were created so that the streams could be split and arranged in a way that there would be a feasible heat exchanger network design.

The designed solutions are:

Case 1) make all combinations between $streams_{in}$ and $streams_{out}$ with a smaller mc_p. Then, split the $stream_{in}$according to the maximum power available to be given to the $stream_{out}$.
Looking at the example presented in Figure 19, the code would generate two additional sets of $streams_{in}$, since Stream 1 would be split according to Stream 3 (Option 1) and Stream 4 (Option 2).

Table 30 – Special Case 1; Split of streams so that a pinch design solution is achieved.

Option 1	
Stream ID	**mcp**
1a	4
1b	1
2	1

Option 2	
Stream ID	**mcp**
1a	3
1b	2
2	1

Case 2) Similarly to Case 1, all combinations between $stream_{in}$ and $stream_{out}$ are analyzed (this time, without mc_p restrictions). Then, according to the mc_p of the streams, one of them is split. The $stream_{out}$ will be split if it has a larger mc_p than $stream_{in}$ and vice-versa. As a simplification, looking only at the mc_p of the streams of the example presented in Figure 20, without considering the stream temperatures - which would have an impact on splitting the streams, the code would generate the streams sets presented in Table 31.

Table 31 - Special Case 2, split of streams so that a pinch design solution is achieved.

Option 1		
Stream ID	**Type**	**mcp [kW/°C]**

1a	hot	20
1b	hot	20
2	hot	30.0
3	cold	60.0
4	cold	20.0
Option 2		
Stream ID	**Type**	**mcp [kW/°C]**
1	hot	40
2a	hot	20
2b	hot	10
3	cold	60.0
4	cold	20.0
Option 3		
Stream ID	**Type**	**mcp [kW/°C]**
1	hot	40
2	hot	20
3a	cold	40.0
3b	cold	20
4	cold	20.0
Option 4		
Stream ID	**Type**	**mcp [kW/°C]**
1	hot	40
2	hot	20
3a	cold	20
3b	cold	40
4	cold	20.0

c. ii) check if the number of $streams_{out} < streams_{in}$

This function aims to check one of the pinch rules, which states that during streams matching , the number of

$$streams_{in} \leq streams_{out}$$

If this does not occur, this function performs the necessary stream splitting to ensure the stream number rule. Only the splits are done, the matches are performed in the following routines. The routine makes all possible streams' splitting that satisfies the pinch rule, and saves the different streams arrangements to later make the matches.

Notice that on the step-by-step algorithm this step occurs twice, at **c.ii** and **c.iv**. At step **c.ii**, only streams reaching the pinch are evaluated. This is to ensure that at step **c.iii,** all $streams_{in}$ can be matched with a $stream_{out}$. At step **c.iv**, all streams are analyzed. Moreover, only splits which are not matched or are streams that do not result from a previous split can be matched.

Looking at the example in Figure 21, there is one surplus $stream_{in}$, thus it is expected that only one split is needed on the $streams_{out}$. The routine only splits $streams_{out}$ when matching with $streams_{in}$ with smaller mcp. Thus, the splits to be done are between C_1-H_1, C_1-H_3, C_2-H_1, C_2-H_2 and C_2-H_3. This will result in five new and different datasets of streams, each with 3 hot and cold streams, to be matched.

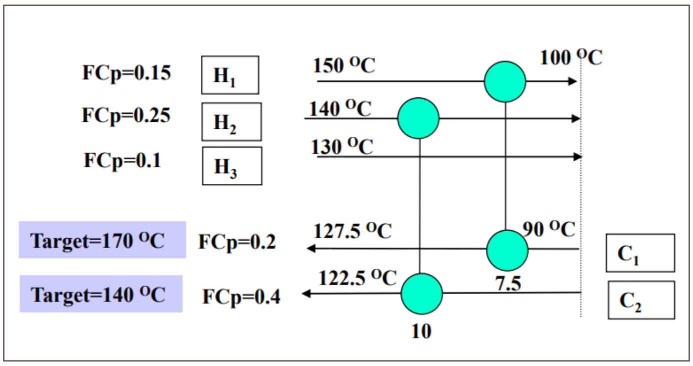


Figure 21 - Example of an analysis not respecting the streams' number rule

c iii) perform first match of streams going into the pinch

The pinch design is always made from the pinch temperature outwards. Thus, it is important to guarantee that all the streams reaching the pinch are matched. Otherwise, if a $stream_{in}$ is not matched and there are no $streams_{out}$ starting from the pinch, it is not possible to fulfil the $stream_{in}$. Figure 22 represents this situation. Since it was first performed the match between H_1 and C_1, stream H_2 cannot be matched with stream C_2 to reach the pinch temperature, leading to no network design.

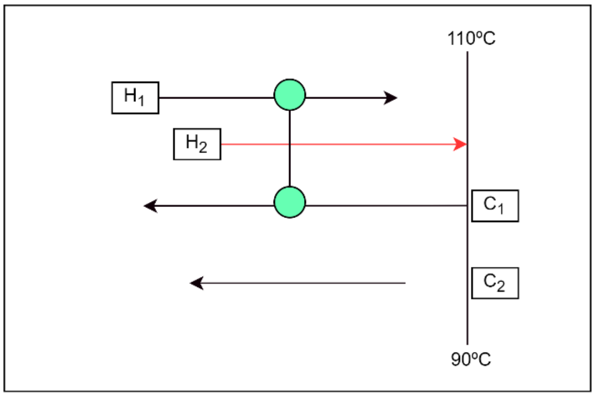


Figure 22 – Example of a $\boldsymbol{stream_{in}}$ – H_2 - which will not be fulfilled due to incorrect first match (H_1-C_1 match) .

The routine *first_match_reach_pinch* runs the recursive function *make_combinations,* which acts similarly to a decision tree. This recursive function initially picks a random stream and makes all possible match combinations of streams that reach pinch until all $streams_{in}$ have been matched. Notice that, the matches performed at each iteration affect the following matches to be executed. The recursive function maximizes pinch design variability by making all possible combinations between $streams_{in}$ and $streams_{out}$. Even though, for a large number of streams it can be time-consuming, it has the benefit of proposing a larger number of pinch designs.

To better visualize the implementation of *first_match_reach_pinch*, it is analyzed the example shown in Figure 23. Starting by the problem **above the pinch**, it can be seen that all streams reach pinch and therefore the routine will try to make all possible combinations between them. The following description, can be visualized in Figure 24. Only matching H_1-C_1 followed by H_2-C_2 will result in a valid pinch design, due to the fact that all $streams_{in}$ reaching pinch are matched. On the opposite, matching H_1-C_2 followed by H_2-C_1 will result in splitting H_2 in H_{2a} and H_{2b}, since mc_p of C_1 ($stream_{out}$) is smaller than of the H_2 ($stream_{in}$) - not respecting the mc_p criteria. The result is that stream H_{2b} will be left without a $stream_{out}$ to match. Since there are no more $streams_{out}$ reaching the pinch, it will be impossible to fulfil the stream H_{2b}. Thus, this HX network will not be further analyzed.

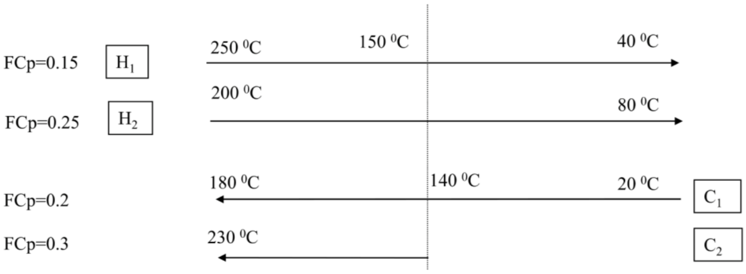


Figure 23 – Pinch problem [4]

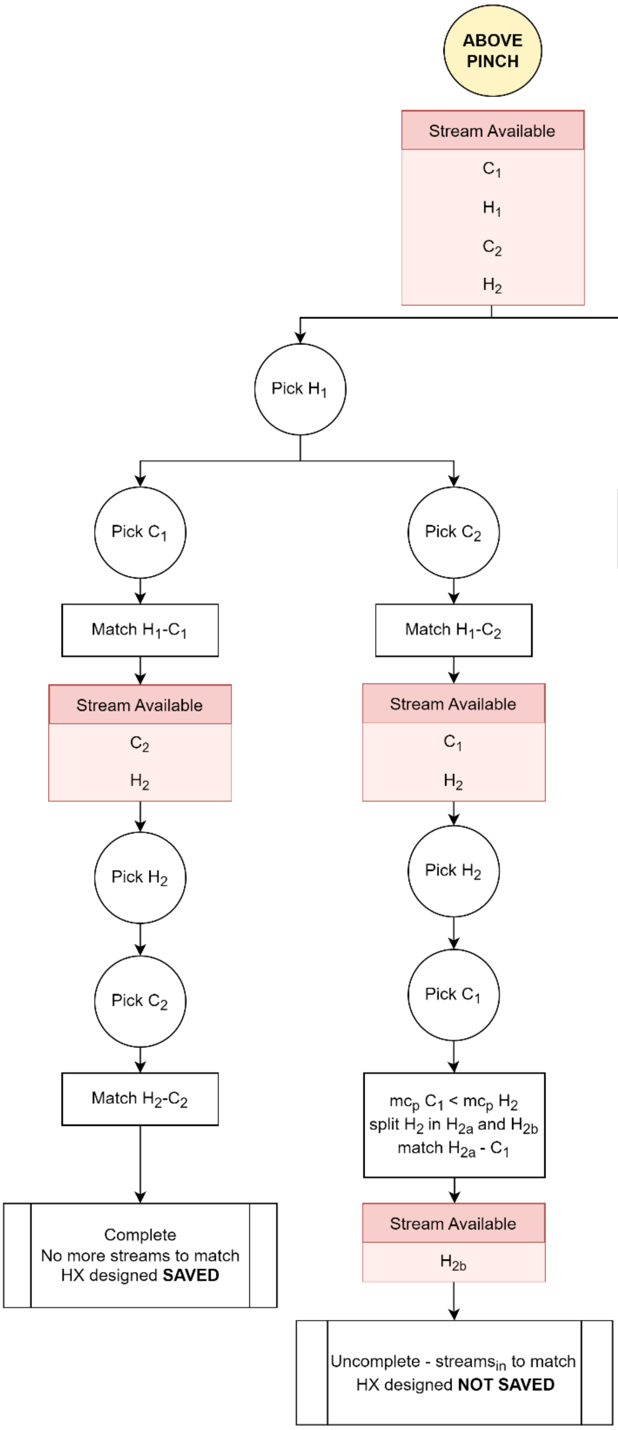


Figure 24 – Resolution of first_match_reach_pinch *on the example given in Figure 23; above pinch temperature*

Considering the problem below the pinch, in the methodology presented in Figure 25, is seen that despite resulting in different pinch design solutions, all the streams going into the pinch are fulfilled. Therefore, both are valid designs. As mentioned, by performing multiple iterations, even though it can be time and computing intensive, it may lead to more pinch designs.

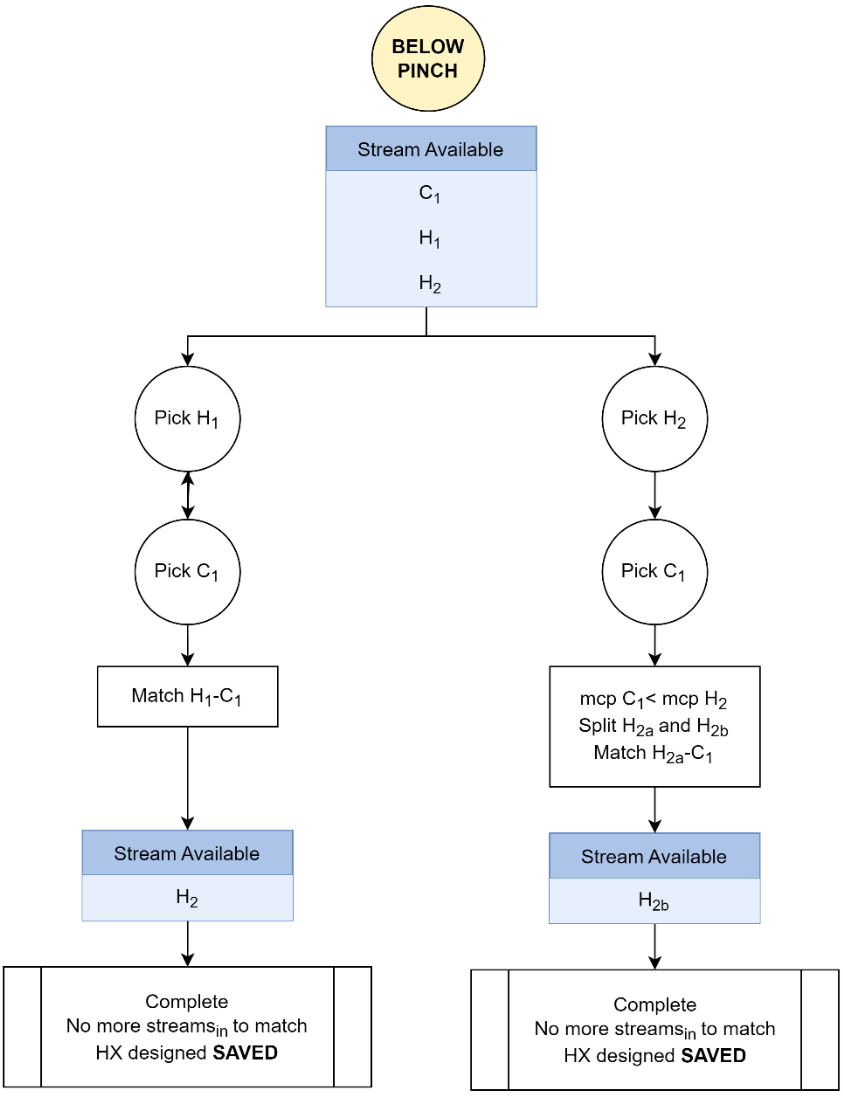


Figure 25 - Resolution of first_match_reach_pinch *on the example given in Figure 23; below pinch temperature*

c.iv) check if the number of $streams_{out} < streams_{in}$

The procedure at this step is similar to the one of step **c.ii**. As mentioned in step **c.ii**, all streams are analyzed in step **c.iv**, and not only the streams reaching the pinch.

c.v) match remaining streams according to power - without split and respecting mc_p criteria

A routine is implemented to design HXs for the streams which are no yet fulfilled – do not reach pinch. At each iteration all the HXs are designed and the most powerful HX is chosen. This approach is based on the tick off rule - heuristic of maximizing the heat load on an interchange by completely satisfying the heat load on one stream - [2] .To achieve this, every possible match between streams available was done and the respective HX designed. At the end, the HX with larger power was chosen and the streams updated. Due to the use of the"ticking off rule", when matching the streams, more complex cases may lead to the $streams_{in}$ not being fulfilled, which results in no pinch design.

c.vi) match remaining streams according to power - without split

The routine is the same as the one described at step **c.v**, however, no restriction respecting the mc_p criteria is applied. This is implemented since after testing with some pinch problem examples, it was observed that easing on this rule could lead to reducing the cases with no design solution.

If after this routine being run, there are still $streams_{in}$ not fulfilled, the pinch design solution will not be saved.

d. Design storage

This routine objective is to compute the storage needed for each HX according to the hot and cold stream schedules. Water or thermal oil storage is designed according to the working temperatures - water if the hot stream temperature is below 90ºC.

Three situations may occur:

1) coincident cold and hot streams schedules;
2) surplus hours of hot stream;
3) surplus hours of cold stream.

For the first situation, the cold and hot stream schedule is coincident therefore no storage is needed.

For the second situation, since there is a surplus of hot stream hours, it is designed the minimum storage required to satisfy 100% the cold stream needs. To design the storage, it is first set a storage with the maximum supply capacity to meet the total 20 days cold stream schedule. Through an iterative process storage capacity is subtracted until the minimum storage that fulfils the cold stream working alone hours.

For the third situation, since there is a surplus of cold stream hours, it is designed a storage that can recover maximum energy from the hot stream working alone hours. The storage is designed according to the longest period that there is a hot stream surplus.

2. Economic and environmental analysis

This analysis objective is to compute the monetary, energy, and CO_2 emissions savings from each equipment for the respective pinch design solution. To give an example, by implementing a HX recovering the heat within processes, it will reduce the heat supply needs from the equipment that feeds this process. The equipment supply capacity can be reduced, meaning that less fuel is needed and consequently less CO_2 emissions and fuel costs. **The full analysis is only performed when processes and equipment are given as an input to the main routine *convert_pinch***. If only isolated streams are provided, the CO_2 emissions analysis is not possible, since isolated streams do not have equipments associated.

Notice that the economic analysis takes only into account the estimated HXs capex and maintenance costs, and does not consider the ones of the circulation pumping or the hot/cold utilities.

ii. Design ORC

This routine (main script: *convert_orc*) assesses as an internal heat recovery alternative, the design of a organic rankine cycle – ORC - for the production of electricity for self-consumption or to sell to the grid. The routine performs a series of ORCs, and returns as output the 3 (default number) best ORC designs in terms of lowest cost of electrical generation (€/kW). The inputs presented in Table 32, and the outputs in Table 33 and Table 34, respectively.

Table 32 - ORC input

Mandatory Inputs				
var name	**description**	**units**	**range**	**data type**
location	[latitude,longitude]	º	[[-90,90], [-180,180]]	array
get_best_number	number of best conversion cases Default:3	-	[0,inf[	integer
streams	See Table 10	-	-	array with dictionaries
fuels_data	Fuels price [€/kWh] and CO2 emission [kg CO2/kWh] for: natural gas, fuel oil, electricity, biomass.	-	-	dict
Optional Inputs				
orc_T_evap	evaporator temperature	ºC	[0,inf[; higher than t_cond	float
orc_T_cond	condenser temperature	ºC	[0,inf[; lower than t_evap	float
interest_rate	Interest rate considered for BM (default:0.04)		]0,1]	float

Table 33- ORC output

Output				
var name	**description**	**units**	**range**	**data type**
best_options	array with the best designs dictionaries	-	-	array with dictionaries
report	HTML report for the ORC analysis	-	-	str

Table 34 - ORC output; best_options

best_options - dictionary				
var name	**description**	**units**	**range**	**data type**
ID	solution design ID	-	[0,inf[	integer
streams_id	array with the converted streams ID	-	-	array with integers
electrical_generation_nominal	nominal electrical supply capacity	[kW]	[0,inf[	float
electrical_generation_yearly	array with hourly electrical supply capacity	[kWh]	[0,inf[	array
excess_heat_supply_capacity	excess heat available supply capacity	[kW]	[0,inf[	float
conversion_efficiency	conversion efficiency of heat to electricity	-	[0,1]	float
turnkey	equipment turnkey	[€]	[0,inf[	float
om_fix	equipment turnkey O&M fix	[€/year]	[0,inf[	float
om_var	equipment turnkey O&M variable	[€]	[0,inf[	float
electrical_generation_yearly_turnkey	electrical production specific cost	[€/year]	[0,inf[	float

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

co2_savings	CO_2 emissions saving	[kg CO_2/kWh]	[0,inf[	float
money_savings	monetary savings	[€/kWh]	[0,inf[	Float
discount_rate	discount rate	-	[0,1]	Float
lifetime	equipment lifetime	years	[0,inf[	integer

The design of the ORC can be for a unique stream or an aggregated of multiple ones. If the excess streams are to be aggregated to enhance the power available to convert in the ORC, an intermediate circuit is designed, with a heat exchanger for each stream. To increase the number of ORC designs and obtain the optimum design, all possible combinations of streams aggregation are analyzed.

To design the ORC, based on ORC manufacturers and the literature [6], the following assumptions were considered:

- the ORC evaporator temperature $T_{evap,}$ by default, was assumed to be 110ºC
- the ORC condenser temperature, by default, was assumed to be $T_{cond} = 35ºC$
- the heat to electricity efficiency is the carnot efficiency multiplied by $factor_{carnot} = 0{,}44$

Various T_{evap} are provided as default, so that when converting the excess streams, the optimum conversion can be obtained. The evaporator and condenser temperatures given are based on the manufacturers ORC commercial equipment with a working fluid R245fa. If the user desires, it is possible to define a T_{evap} and T_{cond} as an input.

For the conversion, first it is computed the available thermal capacity $P_{thermal\ available}$, of the excess heat stream. The supply temperature of the streams to be converted must be superior to the $T_{evap,}$ plus a ΔT_{HX} according to the number of heat exchangers designed (when an intermediate circuit is implemented 2 HX are designed - $\Delta T_{HX} = 2 * 5$).

After estimating the thermal capacity that can be converted, to estimate the ORC electrical generation capacity, the Carnot efficiency is computed:

$$eff_{carnot} = 1 - \frac{T_{cond} + 273.15}{T_{evap} + 273.15}$$

Finally,

$$P_{electrical\ generation}\,[kW] = P_{thermal\ available} * eff_{carnot} * factor_{carnot}$$

The $factor_{carnot}$ is an estimate based on the literature [6] and commerical aplications of ORC provided by the manufacturers.

With the ORC electrical supply capacity and the heat exchangers capacity, it is possible to estimate the design turnkey costs. If the design is made for a combination of streams instead of a unique stream, it is estimated one single ORC for the total available excess heat and all the heat exchangers needed to convert each stream.
After collecting all ORC designs, the best *n* number of options (see Table 32 - ORC input), in terms of lowest cost of electrical generation (kW/€), are returned to the platform.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

iii. Convert Sources to DHN

This submodule will run if the user wishes to design and estimate the costs of converting the available heat of its excess heat streams to the District Heating Network - DHN. For each excess heat stream available the conversion technologies are designed, e.g. a heat exchanger to recover the heat from a hot stream and supply it to the DHN. The design may be done for each stream individually or it can be done to an aggregate of streams of a source (the user must provide their preference). This submodule will run together with the TEO and GIS modules in order to design the correct links and provide a more realistic estimate on the DHN.

1. Implemented Routine

Whenever the user selects a location to be analyzed, the CF *convert_sources* function calculates the conversion technologies from the source´s excess heat stream to supply heat/cold to the selected DHN temperature requirements. The output of *convert_sources* for each excess heat stream is: its available supply heat (kW), the hourly profile (kWh/h), the CO_2 emissions, and linearized costs for the turnkey (€), fixed (€/year) and variable (€/kWh) operation and maintenance costs, for the conversion technologies packages designed.

Table 35 - Convert Sources input

Mandatory Input				
var name	**description**	**units**	**range**	**data type**
group_of_sources	array with sources dictionaries	-	-	array with dictionaries
sink_group_grid_supply_temperature	DHN supply temperature	ºC	[0,120[	float
sink_group_grid_return_temperature	DHN return temperature	ºC	[0,120[	float
existing_grid_data	Existent grid connection point data	-	-	dict

Table 36 - Convert Sources input; group of sources dictionary

group_of_sources – example of a source dictionary				
var name	**description**	**units**	**range**	**data type**
id	source ID	-	[0,inf[	integer
location	[latitude,longitude]	º	[[-90,90], [-180,180]]	array
streams	Table 10	-	-	array with dictionaries
fuels_data	Fuels price [€/kWh] and CO2 emission [kg CO2/kWh] for: natural gas, fuel oil, electricity, biomass.	-	-	dict
existing_grid_data - dict				
id	Existent source or grid connection point ID	-	-	int
location	[latitude,longitude]	º	-	array
levelized_co2_emissions	Grid levelized CO2 emissions	CO2/kWh	[0,inf[	float
levelized_om_var	Grid levelized OM var	€/kWh	[0,inf[	float
levelized_om_fix	Grid levelized OM fix	€/kWh	[0,inf[	float

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

Table 37 - Convert Sources output

Output				
var name	**description**	**units**	**range**	**data type**
all_sources_info	array with multiple source dictionaries	-	-	array with dictionaries
teo_string	TEO output; default: "dhn"	-	"dhn"	string
n_supply_list	GIS output; array contains dictionaries with source and stream ID, stream capacity and source coordinates	-	-	array with dictionaries
input_fuel	TEO required string	-	“dh_water_supply”	string
output_fuel	TEO required string	-	“dh_water_demand”	string
output	TEO required value	-	1	integer
input	TEO required value	-	1	integer
teo_capacity_factor_group	array with dictionaries with the hourly capacity factor for each strem	-	-	array with dictionaries
ex_grid	TEO specific data; existent grid data	-	-	dict

Table 38 - Convert Sources output; n_supply_list dictionary

n_supply_list – example of a source dictionary				
var name	**description**	**units**	**range**	**data type**
id	source ID	-	[0,inf[	integer
stream_id	stream ID	-	[0,inf[	integer
coords	[latitude,longitude]	º	[[-90,90],[-180,180]]	array
cap	stream conversion capacity	[kW]	[0,inf[	float

Table 39 - Convert Sources output; all_sources_info

all_sources_info – dictionary example				
var name	**description**	**units**	**range**	**data type**
source_id	source ID	-	[0,inf[	integer
source_grid_supply_temperature	grid supply temperature the source has to meet	ºC	[0,120]	float
source_grid_return_temperature	grid return temperature	ºC	[0,120]	float
streams_converted	array with streams dictionaries	-	-	array with dictionaries

Table 40 - Convert Sources output; streams_converted

streams_converted – example of dictionary				
var name	**description**	**units**	**range**	**data type**
stream_id	stream ID	-	[0,inf[	integer

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

hourly_stream_capacity	array with hourly stream capacity	[kWh]	[0,inf[	integer
conversion_technologies	array with technologies dictionaries	[kWh]	-	array with dictionaries

Table 41 - Convert Sources output; conversion_technologies dictionary

conversion_technologies – example of dictionary				
var name	**description**	**units**	**range**	**data type**
teo_equipment_name	TEO output; names for technologies packages given by TEO (e.g. 'single_heat_exchanger' , 'multiple_heat_exchanger', …)	-	-	string
output	TEO output; default:1	-	1	integer
input_fuel	TEO output; default: ‘excess_heat’	-	‘excess_heat’	string
output_fuel	TEO output; default: **‘**dhn_water_source’	-	‘dhn_water_source’	string
equipment	array with technologies name (e.g. ['hx_plate', 'heat_pump','hx_plate'])	-	-	-
max_capacity	maximum power stream can provide	kW	[0,inf[	float
turnkey_a	turnkey a value (y=ax+b)	[€/kW]	[0,inf[	float
turnkey_b	turnkey b value (y=ax+b)	[€]	[0,inf[	float
conversion_efficiency	excess heat to usable heat conversion efficiency	-	[0,1]	float
om_fix	fix operation and maintenance costs	[€/year.kW]	[0,inf[	float
om_var	variable operation and maintenance costs	[€/kWh]	[0,inf[	float
emissions	CO_2 emissions per kWh	[kg.CO2/kWh]	[0,inf[	float
technologies	array with technologies designed dictionaries; each dictionary contains detailed info about the technology	-	-	array with dictionaries

2. Conversions Designed

When performing the conversion, three main designs options may occur:

Case 1) if the streams’ supply temperature is larger than the desired DHN supply temperature, a heat exchanger is designed. If the stream is flue gas or the supply temperature is larger than the defined safety temperature of 100ºC, an intermediate oil circuit between the stream and the grid is always designed - high temperatures streams require intermediate circuit for safety;

Case 2) if the streams supply temperature meets the ORC cascaded required evaporator temperature, one is designed. The ORC cascaded produces electricity and can provide heat to the DHN, since the condenser is at a large temperature;

Case 3) if the streams’ supply temperature is lower than the required DHN temperature,not only a heat exchanger is designed to conver the heat from the stream to the DHN but also a heating technology which will boost the DHN temperature to the required temperature value.

In brief, the possible design packages for the conversion are: HX, HX + intermediate circuit + HX, ORC cascaded, heating technology + HX. The respective circulation pumping for each source is designed. In addition, the intermediate circuits circulation pumping is also designed. For all design packages the data is collected and sent in a standardized form for the TEO module (script: *join_hx_and_technology*).

The following paragraphs will describe each of the cases in more detail.

CASE 1

For Case 1, the minimum required temperatures that the excess heat should meet are computed from the grid temperatures. It is then possible to estimate the excess heat stream available heat capacity, and design the correspondent heat exchanger.
As an example, considering that the DHN supplies hot water at 90ºC and returns at 50ºC, and a source has an excess heat stream with the following properties:

- fluid: flue gas
- $\boldsymbol{c_p = 1.3\ kJ/kg.K}$
- $\boldsymbol{T_{supply} = 350ºC}$
- $\boldsymbol{T_{target} = 25ºC}$ (released to the atmosphere)
- $\boldsymbol{\dot{m}_{stream} = 6000\ kg/h}$
- $\boldsymbol{P_{stream}}$ = 704 kW

Table 42 - Convert Source; Case 1 - HX design temperatures

	Source Stream	Intermediate Circuit Stream	Grid Stream
Fluid	Flue gas	Thermal oil	water
T_cold [ºC]	120	55	50
T_hot [ºC]	350	95	90

Since the stream temperature is superior to the DHN safety temperature of 100ºC and it is flue gas, an intermediate thermal oil circuit is designed. Since there is and intermediate circuit, two heat exchangers will be dimensioned. Thus, the minimum target temperature possible to recover heat from the stream, is

$$T_{target\ minimum} = T_{grid_{return}} + \Delta T_{HX} * number_{hx} = 50 + 5 * 2 = 60\ ºC$$

However the stream fluid is flue gas, hence it is only considered cooling until 120ºC and not 60ºC. Considering all technologies, the thermal capacity available to be recovered is

$$P_{HX} = \dot{m} * c_p * (\mathrm{T_{supply}} - \mathrm{T_{target}}) + hx_{eff}^{number_{hx}} = \frac{6000}{3600} * 1.3 * (350 - 120) * 0.95^2 \approx 450\ kW$$

The streams' available power to convert is 450kW, and not 704kW. The power that both heat exchangers must be designed for, as well as the hot and cold working fluid temperatures is now known. The HX_1 for 450kW, and the HX_2 for 428 kW (450 * hx_{eff}). The supply and target temperatures of the stream are 350ºC and 120ºC, and for the intermediate circuit of 95ºC and 55ºC. The proposed design is presented in Figure 26.

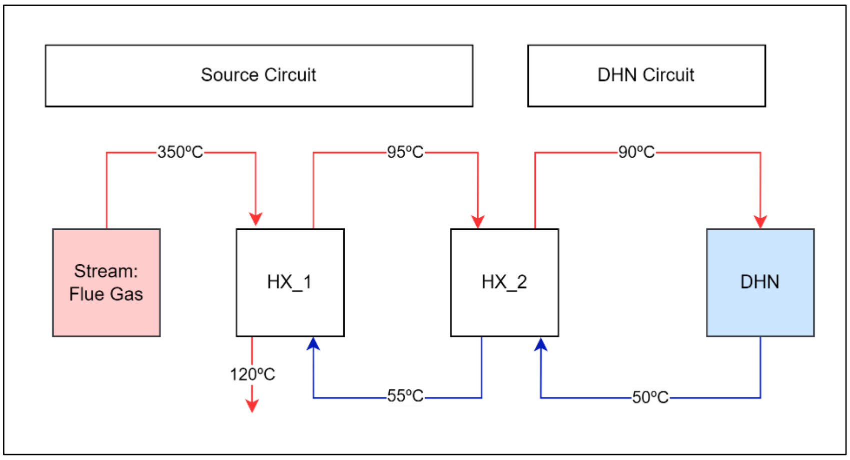


Figure 26 - Case 1 Convert Source; Technologies: HX diagram

CASE 2

For Case 2, from the DHN temperatures, the minimum required temperatures that the stream should meet are computed. It is then possible to obtain the available stream capacity, and design the ORC cascaded.
From the ORC cascaded manufacturers information ([7] and [8]), it was assumed that the minimum temperature difference between evaporator and condenser is 50ºC.

As an example, if the DHN supply temperature is of 90ºC, the ORC cascaded condenser will be designed to have a condenser temperature of 95 ºC (considering the ΔT_{HX}), and an evaporator temperature of 95º+45ºC=140ºC. If the stream supply temperature is not larger than the evaporator temperature, the ORC will not be designed. Otherwise, the routine described in ii will compute the ORC electrical generation and the supplied thermal capacity.

Looking at the example described in Case 1 and the temperatures computed, it is observed that the stream supply temperature is higher than the ORC evaporator temperature (350ºC>140ºC). Thus, it is possible to design it. The available heat capacity to be converted from the stream considers the heat between the streams' supply temperature and the ORC evaporator temperature. With this data, it is possible to obtain the ORC thermal and electrical supply capacity. The proposed design is presented in Figure 27.

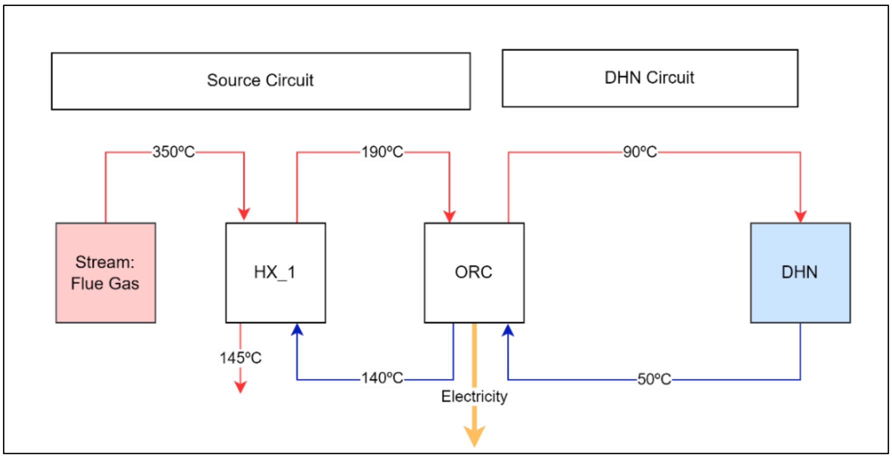


Figure 27 - Case 2 Convert Source; Technologies: HX + ORC diagram

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

CASE 3

Case 3 occurs when the stream's supply temperature does not meet the required the minimum required temperature for the DHN. A large number of these streams may be of great interest to convert because they may have a large thermal capacity. In order to make the stream convertible, heating technologies are designed so that the DHN reaches the desired temperature (see Figure 28 - Case 2). The heating technologies that can be designed to increase the DHN temperature are: boiler, heat pump, CHP and solar thermal.
When implementing a boiler or a CHP, first it is computed the grid temperature increase that the excess heat stream can provide, and then it is computed the capacity of the technology needed to meet the grid temperature, as it is presented in Figure 28 - Case 2 and Figure 29.

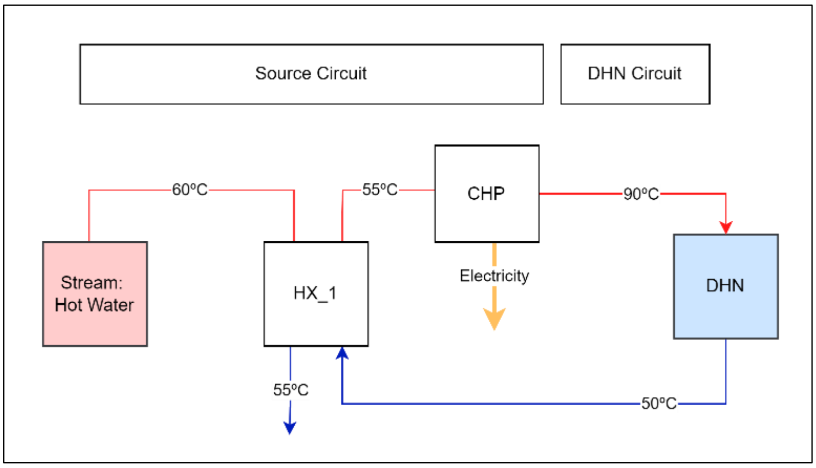


Figure 28 - Case 2 Convert Source; Technologies: HX + CHP diagram

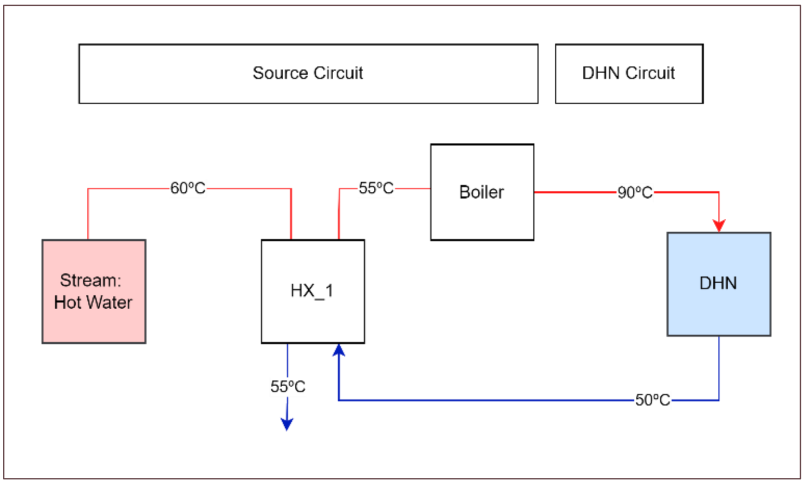


Figure 29 - Case 2 Convert Source; Technologies: HX + Boiler diagram

When implementing a heat pump (see Figure 309), it is computed the capacity provided by the excess heat stream to the heat pump evaporator. It is considered the evaporator temperature is at the excess heat stream target temperature minus the heat exchanger temperature difference. The condenser temperature is set equal to the grid supply temperature plus the heat exchanger temperature difference. With the evaporator and condenser temperature it is possible to compute the coefficient of performance COP and then apply a correction factor [9]

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

$$COP = 0.55 * COP_{Carnot} = 0.55 * \frac{(T_{cond} + 273)}{T_{evap} - T_{cond}}$$

Knowing the COP and the evaporator capacity, it is computed the heat pump supply capacity to the DHN as,

$$Q_{cond} = Q_{evap} + W <=> Q_{cond} = Q_{evap} + \frac{Q_{cond}}{COP} <=> Q_{cond} = \frac{Q_{evap}}{1 - \frac{1}{COP}}$$

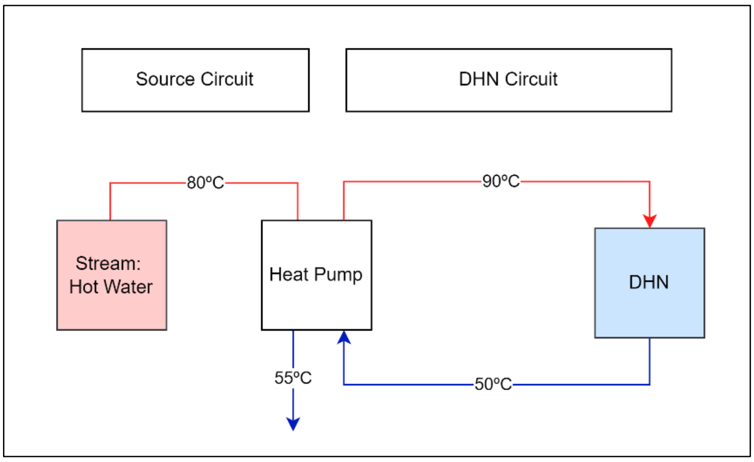


Figure 30 - Case 2 Convert Source; Technologies: Heat Pump diagram

The solar thermal is dependent on the climate conditions and therefore a different design approach is required. A boiler or a heat pump is always designed as backup because, according to the climate conditions, the solar thermal may not provide the required capacity. To design the solar thermal (see Figure 31), first, it is obtained the nominal produciton capacity in the whole year. For the defined location, it is obtained from the *pvlib* – python library - the incident solar radiation $[W/m^2]$ and ambient temperatures. According to the solar thermal properties and the temperature difference between equipment and ambient, as well as the incident radiation, the solar thermal efficiency is computed,

$$\Delta T_{solar\ colllector-exterior}[ºC] = \frac{T_{supply} + T_{return}}{\mathbf{2}} - T_{exterior}$$

$$eff_{solar\ collector} = C_0 - C_1 * \frac{\Delta T_{solar\ colllector-exterior}}{Q_{rad}} - C_2 * \frac{(\Delta T_{solar\ colllector-exterior})^2}{Q_{rad}}$$

Where, C_0, C_1 and C_2 are performance coefficients of the solar panel, these will vary according to the solar thermal collector type. The power for each hour is then,

$$P_{nominal_{solar\ collector}}\ [W/m^2] = Q_{rad} * eff_{solar\ collector}$$

Taking into account all the productive hours, the nominal solar thermal production $P_{nominal_{solar\ collector}}$, is obtained. Knowing the nominal capacity needed by the stream, the solar thermal area is computed,

$$A = \frac{P_{stream}}{P_{nominal_{solar\ collector}}}$$

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

The CAPEX of the whole design package is the sum of both solar thermal and backup heating technology – boiler or heat pump - CAPEX. The variable operation and maintenance - O&M - costs as well as the CO_2 emissions of the whole designed solution are computed according to the estimated solar thermal production and the required backup capacity. To perform this estimate, it is computed the fraction of total energy the solar thermal provides to the stream,

$$coef_{solar\ thermal} = \frac{P_{yearly_{solar\ thermal}}}{P_{yearly_{stream}}}$$

The $P_{yearly_{solar\ thermal}}$ is the power the solar thermal provides when the stream is also operating. As an example, if at a certain day the solar thermal works from 9h to 17h and the stream is operating between 15h and 24h, only the period between 15h and 17h is considered. The solar thermal is only considered as an option if it satisfies at least 30% of the stream needs when they are both operating.

The variable O&M costs as well as the CO_2 can then be corrected by applying the following computation,

$$om_{var} = om_{var_{heating\ technology}} * \left(1 - coef_{solar\ thermal}\right) + om_{var_{solar\ thermal}} * coef_{solar\ thermal}$$

$$emissions = emissions_{heating\ technology} * \left(1 - coef_{solar\ thermal}\right) + emissions_{solar\ thermal} * coef_{solar\ thermal}$$

where, om_{var} is the O&M variable costs in €/kWh, and the $emissions$ is the CO_2 emissions in kg.CO_2/kWh.

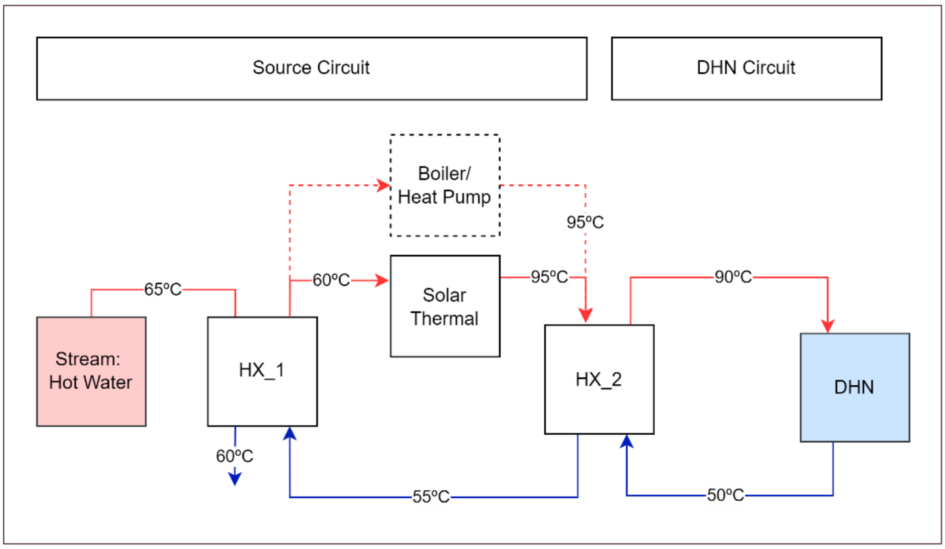


Figure 31 - Case 3 Convert Source; Technologies: Solar Thermal diagram

3. Grid Losses

To take into account a rough estimate on the grid losses, the technologies on the sources were designed for a higher temperature that the one desired by the grid. It

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

was assumed a $delta_{T_{buffer}} = 6ºC$, meaning that if the grid supply and return temperature were 90ºC and 50ºC, the technologies were designed for 96ºC and 44ºC.

2.1.6.5 Sink

The CF sink submodule is divided into: characterization and simulation. The characterization is responsible for receiving the input data from the user and estimate the sink heating and cooling needs. The simulation aims to evaluate and design the technologies needed to convert the DHN heat into the sinks needs. Figure 32 highlights the main features of the sink submodule.

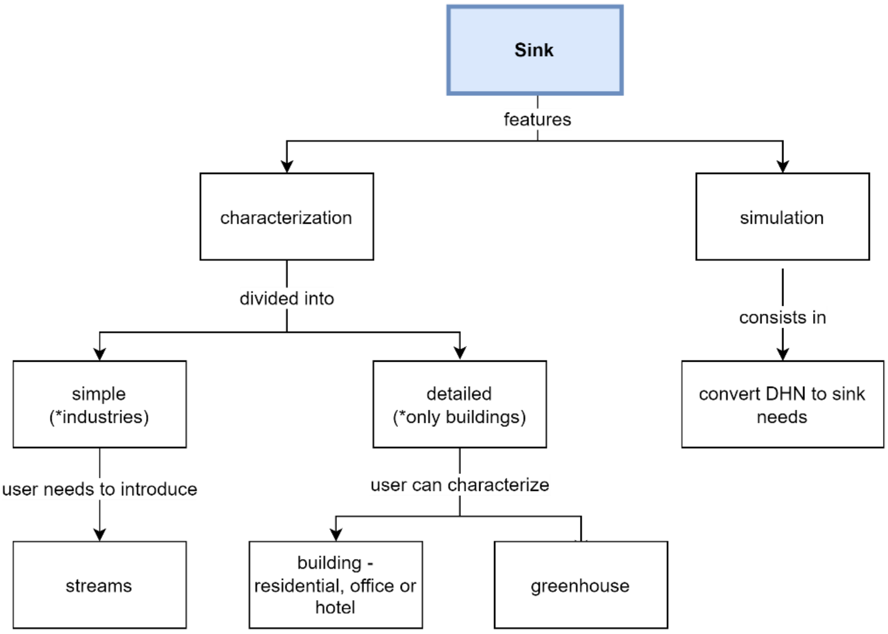


Figure 32 - Sink submodule overview

2.1.6.5.1 Characterization

Similarly to the Source, the characterization is divided into **simple** and **detailed**. For the **simple** characterization the industry user needs to characterize a hot water, steam and/or cold water streams. A user that performs the **detailed** characterization can analyze a building heat and cooling needs by introducing the building characteristics. Sink objects characterized in both detail or simplified forms, can later on be handled to perform the simulation. The characterization has 3 main functions (scripts: *building, greenhouse and simple_user)* and auxiliary functions which are fully described in the Github repository (https://github.com/Emb3rs-Project/p-core-functionalities/tree/master/src/Sink/characterization). Figure 33 presents a simplified relationship diagram between the characterization functionalities.

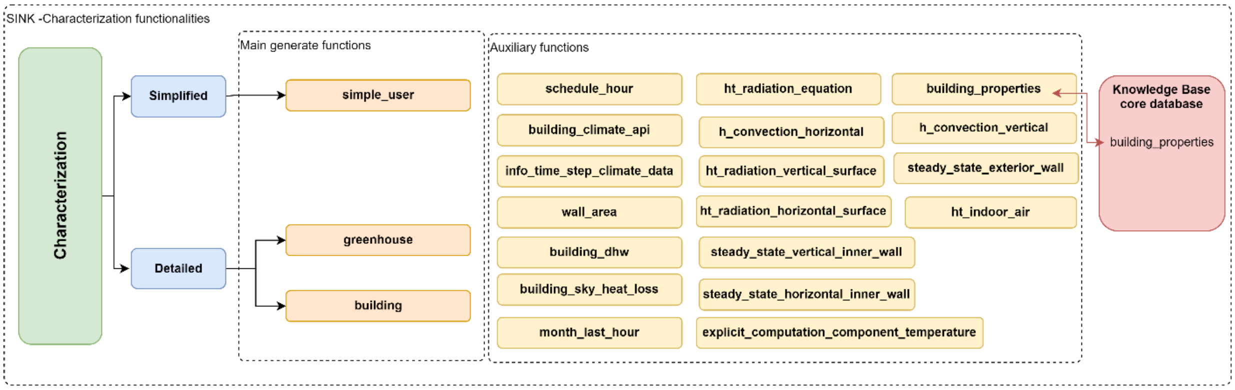


Figure 33 - Schematic relationship diagram of the CF sink characterization submodule

a. Simple

The simple characterization is performed as described in section 1.1.1.1. However, instead of characterizing the excess heat streams, the industry user needs to characterize 3 types of streams – hot water, steam or cold water – and provide reference values of the technologies it possesses (if existent). Industries, usually, possess combinations – 1, 2 or all of them - of these 3 types working fluids that supply to the different processes. If the user posseses for example two hot water boilers, the stream’s parameters provided should be according to the largest temperature hot water boiler. The same should be applied to the steam and cold water streams (smallest temperature in this case).

b. Detailed

The building and greenhouse routines are for users that intend to simulate a climate dependent heating/cooling demand. The functions will generate a quick estimate on the hourly heating/cooling demand profile for a full year based on climate data and buildings’ indoor temperature requirements. The current implementation can generate demand for three types of buildings: residential, offices and hotels, as well as greenhouses. Further typologies and more accurate will be implemented in future releases. Further information on the individual functionalities can be found on the Github repository (https://github.com/Emb3rs-Project/p-core-functionalities/tree/master/src/Sink/characterization).

i. Buildings – Residential, Hotel and Office

This routine's objective is to estimate a building’s heat and cooling consumptions over the year, with the input parameters provided by the user. These inputs, which are presented in Table 43, are divided into mandatory and optional inputs. The user must introduce the mandatory inputs in order to run the simulation. These are simple parameters that any type of user (from basic to advanced) understands and can introduce to run the simulation. The optional values are advanced building parameters, which have default values so that the simulation runs, and should only be modified by advanced users. The user that desires a more accurate building characterization can modify these default optional valuesTable 43 - Building Mandatory/Optional inputs. In addition to the input building parameters, the local climate profile also influences the buildings’ heating and cooling needs. For this reason, for each location the respective weather data is obtained through the *pvlib* package. The reasoning on the buildings thermal modeling approach is based on [10].

Table 43 - Building Mandatory/Optional inputs

Mandatory Input				
var name	**description**	**units**	**range**	**data type**
location	building location [lat,long]	º	[-90,90], [-180,180]	array
number_floor	number of floors	-	[1-200]	integer
width_floor	floor’s width	m	]0-inf[	float
length_floor	floor’s length	m	]0-inf[	float
height_floor	floor’s height	m	]0-inf]	float
ratio_wall_N	percentage of northe wall area in total north facade area (wall and glass)	-	[0-1]	float
ratio_wall_S	percentage of south wall area in total south facade area (wall and glass)	-	[0-1]	float
ratio_wall_E	percentage of east wall area in total east facade area (wall and glass)	-	[0-1]	float
ratio_wall_W	percentage of west wall area in total west facade area (wall and glass)	-	[0-1]	float
saturday_on	if working -1- or not -0- on Saturday	-	0 or 1	integer
sunday_on	if working -1- or not -0- on Sunday	-	0 or 1	integer
shutdown_periods	periods of days the building is not occupied	days	[1,365]	array with arrays
daily_periods	daily hourly period sbuilding is occupied	hour	[0-24]	array with arrays
building_type	type of building	-	'office', 'residential' or ' hotel'	string
'residential' -> mandatory input -> number_person_per_floor	number of persons per floor; mandatory input for residential buildings	-	[0-inf[	integer
building_type = 'hotel' -> mandatory input -> number_rooms	number of rooms per floor; mandatory input for hotel buildings	-	[0-inf[	integer
space_heating_type	Space heating type: 1) 0 = Conventional; heaters working fluid supply temperature of 75ºC, heaters working fluid return temperature of 45ºC) 2) 1 = Low temperature; heaters working fluid supply temperature of 50 ºC, heaters working fluid return temperature of 30ºC)	-	0 or 1	integer
building_orientation	building’s main facade orientation	-	'N','S','E' or 'W'	string
ref_system_fuel_type_heating	Fuel type associated; e.g. "natural_gas","electricity","bio mass","fuel_oil","none"	-	-	string
ref_system_fuel_type_cooling	Fuel type associated; e.g. "natural_gas","electricity","bio mass","fuel_oil","none"	-	-	string
Optional Input				

number_person_per_floor	number of persons per floor	-	-	integer
supply_temperature_heat	heaters working fluid supply temperature	ºC	]0,100[; must be lower than target_temperature_heat	float
target_temperature_heat	heaters working fluid target temperature	ºC	]0,100[; must be higher than supply_temperature_heat	float
supply_temperature_cool	cooling working fluid supply temperature;	ºC	]0,30[; must be higher than target_temperature_cool	float
target_temperature_cool	cooling working fluid target temperature	ºC	]0,30[; must be lower than supply_temperature_cool	float
T_cool_on	maximum temperature in a room during occupied hours; cooling is turned on.	ºC	]0,50[; must be lower than T_heat_on	float
T_heat_on	minimum temperature in a room during occupied hours; heating is turned on.	ºC	]0,50[; must be larger than T_cool_on	float
T_off_min	minimum temperature in a room during unoccupied hours; heating is turned on.	ºC	]0,50[; must be lower than T_heat_on	float
T_off_max	maximum temperature in a room during unoccupied hours; cooling is turned on.	ºC	]0,50[; must be higher than T_heat_on	float
tau_glass	glass windows transmissivity	-	]0,1]	float
alpha_wall	wall’s radiation absorption coefficient	-	]0,1]	float
alpha_floor	floors’ radiation absorption coefficient	-	]0,1]	float
alpha_glass	windows’ radiation absorption coefficient	-	]0,1]	float
u_wall	wall U value	[W/m2.K]	]0,20]	float
u_roof	roof U value	[W/m2.K]	]0,20]	float
u_floor	floor U value	[W/m2.K]	]0,20]	float
u_glass	glass U value	[W/m2.K]	]0,20]	float
cp_roof	roof specific heat capacitance	[J/m2.K]	]50000,250000[	float
cp_wall	wall specific heat capacitance	[J/m2.K]	]5000,90000[	float
air_change_hour	air changes per hour due to infiltrations	[1/h]	[0,20]	float
renewal_air_per_person	fresh air changer per person	[m3/s.person]	[0, 0.05]	float
vol_dhw_set	volume of daily water consumption	[m3]	[0, inf[	float
Q_gain_per_floor	heat gains due to miscellaneous equipment (e.g. lamps, computers...)	W	[0, inf[	float
emissivity_wall	walls radiation emissivity coefficient	-	]0,1]	float
emissivity_glass	windows’ radiation emissivity coefficient	-	]0,1]	float
ref_system_eff_equipment_heating	Efficiency of the heating equipment	kWh	]0,1]	float
ref_system_eff_equipment_cooling	COP of the cooling equipment	kWh	]0,1]	float

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

real_heating_monthly_capacity	Real monthly data - for each month of the year	kWh	[0, inf[	array
real_heating_yearly_capacity	Real yearly data - single value	kWh	[0, inf[	float
real_cooling_monthly_capacity	Real monthly data - for each month of the year	kWh	[0, inf[	array
real_cooling_yearly_capacity	Real yearly data - single value	kWh	[0, inf[	float

1. Step by step

The thermal model to characterize the building's heating and cooling needs includes solving heat balance equations based on heat transfer mechanisms, which are affected by the climate conditions and the building parameters. At each time step, heat balances are computed to the walls and windows, as well as the floor and ceiling of each of the floors in order to estimate its temperature, as presented in Figure 34. The thermal model step by step methodology implemented is as follows:

1. **Get building parameters**
2. **For each time step**
    a. **Get climate data**
    b. **Climate data treatment**
    c. **Compute incident solar radiation**
    d. **Compute inner radiation heat transfer**
    e. **Compute sky radiation heat transfer**
    f. **Compute convection heat transfer**
    g. **Compute infiltrations**
    h. **Compute internal heat gains**
    i. **Compute space heating/cooling actuation**
    j. **Compute hot water consumptions needs**
    k. **Solve surfaces heat balance**
    l. **Update surfaces temperatures**
    m. **Save heating/cooling space actuation need**
    n. **Return to step a) if not yet in the last time step**
3. **Return monthly and hourly heating/cooling needs**

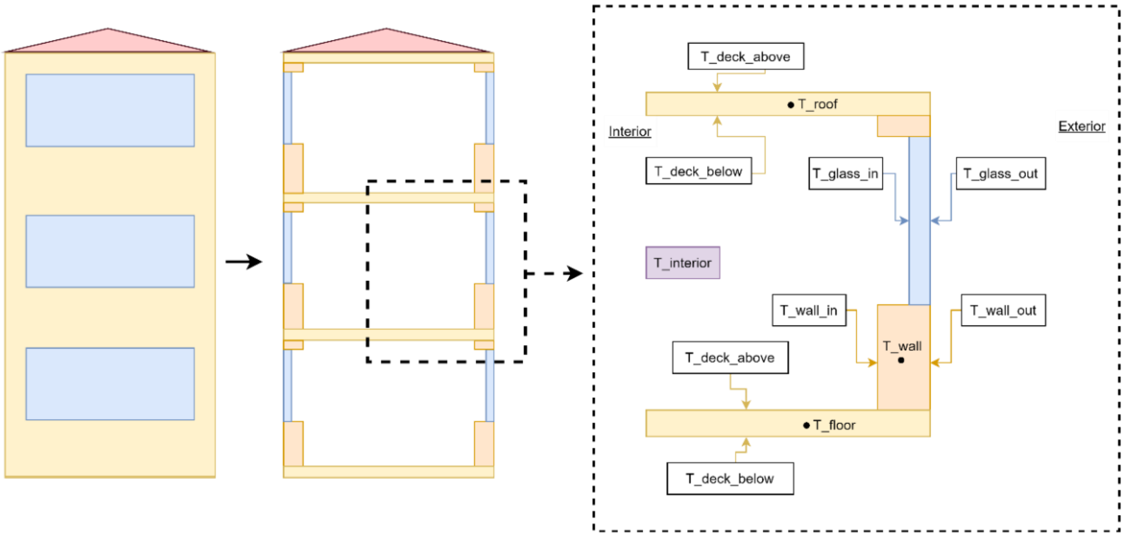


Figure 34 - Building schematic representation with detailed view of the considered surfaces in the thermal model

Climate data

For each time step, the *pvlib* package provides the location climatic conditions (script: *building_climate_api),* such as, total diffuse solar radiation and direct incident solar radiation on a horizontal surface [W/m^2], ambient and sky temperatures [ºC], as well as wind speed at 10m.

The raw data must be treated. Starting with the indicent solar radiation, the building's faces are considered to be vertical, thus the horizontal incident solar radiation must be corrected to the one incident on a vertical wall. By computing the sun's position for each time step, the incident solar radiation on the vertical facades is estimated,

$$\theta_{wall} = cos(sin(\alpha_{sun}) * cos(\beta_{wall}) + cos(\alpha_{sun}) * sin(\beta_{wall}) * cos(\phi_{wall} - \phi_{sun})$$

where, θ_{wall} is the incidence angle of solar radiation in te wall, ϕ_{wall} the wall azimuth, β_{wall} the wall inclination, α_{sun} is the solar altitude angle and ϕ_{sun} the solar azimuth angle.

After computing the incidence angle, it is estimated the total solar incident radiation $G_{sun_{facade}}$, on each building's facade,

$$G_{sun_{facade}} = G_{beam} * cos(\theta_{wall}\ ) + \frac{180\ - \beta_{wall}}{180} * G_{diffuse}$$

Where, G_{beam} and $\mathrm{G}_{\mathrm{diffuse}}$, are the direct and diffuse solar radiation, respectively.
The wind speed is obtained for a 10m height. As an approximation, it is corrected the wind speed for half of the building's height to get a representative value of wind speed to the whole building. Applying the logarithmic wind profile [11] and assuming a surface roughness of $z_0 = 0.01$ (open flat terrain surface roughness), the wind speed is obtained as,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

$$v_{wind_{average}} = v_{wind_{10m}} * \frac{\log\left(\frac{height_{floor} * \frac{number_{floor}}{2}}{z_0}\right)}{\log\left(\frac{10}{z_0}\right)}$$

Heat Transfer

This section analyses all the heat transfer items that have an impact on the building thermal model.

1. Solar Radiation

For the considered buildings, there is incident solar radiation on its exterior vertical facades and its roof (assumed as an horizontal surface), and the interior floors surface. The absorbed solar radiation on a specific surface depends on its absorption coefficient $\alpha_{surface}$.
For the building exterior surfaces,

$$Q_{sun_{surface}}[W] = \alpha_{surface} * G_{sun_{surface}} * A_{surface}$$

For interior surfaces the transmissivity of the windows glass, τ_{glass}, will affect the amount of incident solar radiation,

$$Q_{sun_{surface}}[W] = \alpha_{surface} * G_{sun_{surface}} * A_{surface} * \tau_{glass}$$

2. Outside Convection

Exterior surfaces exchange heat with the ambient air by convection. To compute the exchanged heat, the heat transfer coefficient [12] is computed as,

$$h_{c,ext} = (5.8 + 3.94 * v_{wind_{average}})$$

and the total heat transfer,

$$Q_{conv_{ext}} = h_{c,ext} * A_{surface} * \Delta T$$

Where, ΔT is the temperature difference between the facade and the ambient air.

3. Outside Radiation

For exterior surfaces, it is computed the heat exchanged by radiation with the sky and ground. Firstly, the view factors between surfaces are computed.

$$F_{sky} = 0.5 * \left(1 + cos\left(\theta_{surface}\right)\right)$$
$$F_{ground} = 0.5 * \left(1 - cos\left(\theta_{surface}\right)\right)$$

Here, $\theta_{surface}$ is the surface inclination. The view factor to the sky is further split between sky and air radiation by,

$$\beta = \sqrt{\left(0.5 * \left(1 + cos\left(\theta_{surface}\right)\right)\right)}$$

The total amount of heat exchanged by radiation is,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

$$Q_{rad_{outside}} = Q_{rad_{sky}} + Q_{rad_{air}} + Q_{rad_{ground}}$$

Looking at each term separately,

$$Q_{rad_{sky}} = \varepsilon_{surface} * \sigma_{Boltzmann} * \left(\left(T_{sky} + 273\right)^4 - \left(T_{surface} + 273\right)^4\right) * F_{sky} * \beta$$

$$Q_{rad_{air}} = \varepsilon_{surface} * \sigma_{Boltzmann} * \left((T_{exterior} + 273)^4 - \left(T_{surface} + 273\right)^4\right) * F_{sky} * (1 - \beta)$$

$$Q_{rad_{ground}} = \varepsilon_{surface} * \sigma_{Boltzmann} * \left((T_{exterior} + 273)^4 - \left(T_{surface} + 273\right)^4\right) * F_{ground}$$

where, $\varepsilon_{surface}$ is the surface's emissivity, and $\sigma_{Boltzmann}$ the stefan boltzmann constant ($\sigma_{Boltzmann} = 5.6704 \times 10^{-8}\ W/m^2.K$).

4. Inside Convection

Inside the building, heat transfer by convection occurs between walls, glass, ceiling and the floor with the interior air, T_{air}, due to the temperature difference. According to the surface position – vertical or horizontal – a heat transfer coefficient, h_c, is applied. For vertical surfaces [10],

$$h_c\left[\frac{W}{m^2.K}\right] = 1.31 * \Delta T^{\frac{1}{3}}$$

For horizontal surfaces [10] if $T_{surface} > T_{air}$ downward facing surface or $T_{surface} < T_{air}$ and upward facing surface,

$$h_c\left[\frac{W}{m^2.K}\right] = 9.482 * \frac{\Delta T^{\frac{1}{3}}}{7.283 - \left|\cos(\theta_{surface})\right|}$$

If $T_{surface} < T_{air}$ downward facing surface or $T_{surface} > T_{air}$ and upward facing surface,

$$h_c\left[\frac{W}{m^2.K}\right] = 1.81 * \frac{\Delta T^{\frac{1}{3}}}{1.382 + \left|\cos(\theta_{surface})\right|}$$

The total amount of heat exchanged between the surfaces and interior air is similar to the one given by

$$Q_{conv_{int}} = h_{c,int} * A_{surface} * \Delta T$$

5. Inside Radiation

To compute the heat exchange by radiation between inner surfaces the view factor between them is needed [10]. Approximately, the view factor of surface 1 to surface 2 is the ratio of the area of surface 2 to the total area "seen" by surface 1.

$$F_{1-2} = \frac{A_{surface\ 2}}{A_{seen\ by\ surface\ 1}}$$

Thus, the heat exchanged is

$$Q_{rad}[W] = \varepsilon_{surface} * \sigma_{Boltzmann} * A_{surface\ 1} * \left(\left(T_{surface\ 2} + 273\right) ** 4 - \left(T_{surface\ 1} + 273\right)^4\right) * F_{1-2}$$

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

6. Conduction

Heat transfer across a solid material occurs if there is a temperature difference between its surfaces. The amount of exchanged heat varies according to its thermal conductivity, u. The heat exchanged by conduction is the following,

$$Q[W] = uA\Delta T$$

7. Internal Gains

The occupants of a building as well as the equipments, are important internal gains that affect the heating/cooling needs. The internal gains vary according to the type of building assigned:

1) For residential buildings and hotels, the heat gain due to occupancy and appliances is

$$Q_{gain}[W] = coef_{equipment} * A_{floor}$$

where, $coef_{equipment} = 4\ W/m^2$ [13].

If it is chosen an office building,

$$Q_{gain}[W] = number_{person\ per\ floor} * coef_{person} + coef_{equipment} * A_{floor}$$

where, $coef_{equipment} = equipment + ligths = 18\ W/m^2$ [13] and $coef_{person} = 108\ W/person$ [14].

8. Air Renewal

Office buildings, which are occupied for long periods of time, require minimum renewal of air to ensure a good air quality. The value assumed [14] for constant renewal air per person is:

$$renewal\ air_{per\ person}\ [m^3/s] = 10 * 10^{-3}$$

The heat transfer due to air renewal is the following,

$$Q_{renewal\ air} = (\rho_{air} * c_{p_{air}} * renewal\ air_{per\ person} * number_{person\ per\ floor}) * (T_{exterior} - T_{interior})$$

9. Infiltrations

Infiltrations are an introduction of outside air into a building envelope usually caused by wind or negative building pressurization. The better the building construction, the better it is protected from infiltrations. The heat exchange due to infiltrations is,

$$Q_{infiltration}\ [W] = \rho_{air} * c_{p_{air}} * \frac{ACH}{3600} * (T_{exterior} - T_{interior})$$

where, ACH [1/h] is the air change per hour rate due to infiltrations.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

Heat Balances

This section analyses all the heat balances that are solved on building thermal model. Both, steady state and dynamic equations are applied. The objective of the equations is to estimate the surfaces' temperature at each time step. To all the steady state heat balances applied, the temperature value obtained for that time step is interpolated with the value of the previous time step. The non-interpolation could lead to unrealistic large variations of temperature since the body does not have thermal inertia.

1. Outer Glass

Since the window has low thermal inertia, a steady state balance is applied to both inner and outer window surfaces. For the outer window (see Figure 35 – Window outer surface's exchanged heat schematic representation),

$$Q_{cond} + Q_{rad_{sky}} + Q_{conv_{ext}} = Q_{sun}$$

Manipulating the equation to obtain at the end of the time step the outer glass temperature,

$$T_{glass_{outer}} = \frac{Q_{sun} * \alpha_{glass} + Q_{rad_{sky}} + h_{exterior} * T_{exterior} + u_{glass} * T_{glass_{inner}}}{h_{exterior} + u_{glass}}$$

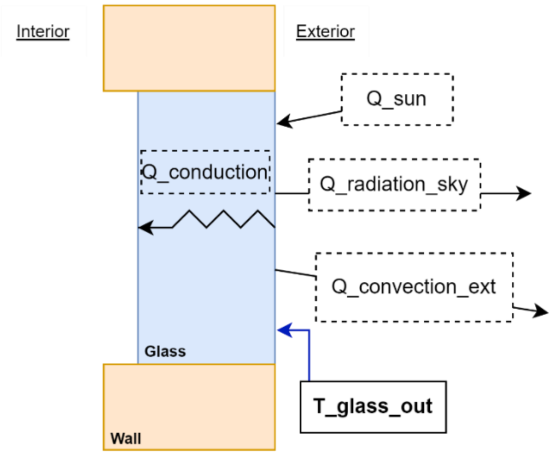


Figure 35 – Window outer surface's exchanged heat schematic representation

2. Inner Glass

For the inner glass (see Figure 36),

$$Q_{conv_{int}} + Q_{rad_{int}} = Q_{sun} + Q_{cond}$$

Manipulating the equation to obtain at the end of the time step the inner glass temperature,

$$T_{glass_{inner}} = \frac{Q_{rad_{int}} + h_{vertical} * T_{interior} + u_{glass} * T_{glass_{outer}}}{h_{vertical} + u_{glass}}$$

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

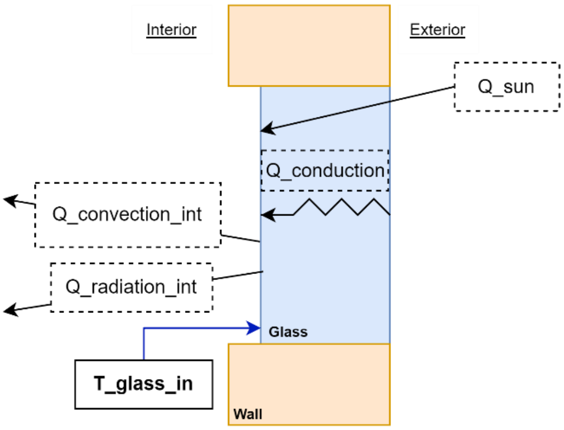


Figure 36 - Glass inner surface's exchanged heat schematic representation

3. Walls

For the walls, the heat balance equations that occur to the inner and outer wall are similar as the one occurring for the inner and outer glass. Except for the fact that the heat conduction term will be between the wall surfaces and the wall itself, and not directly between surfaces. An additional transient heat balance is applied for the wall body, so that it is considered the wall thermal inertia (see Figure 37). For this additional heat balance the equation is the following,

$$\left(\frac{mc_pT}{dt}\right)_{wall} = Q_{cond_{inner}} + Q_{cond_{outer}}$$

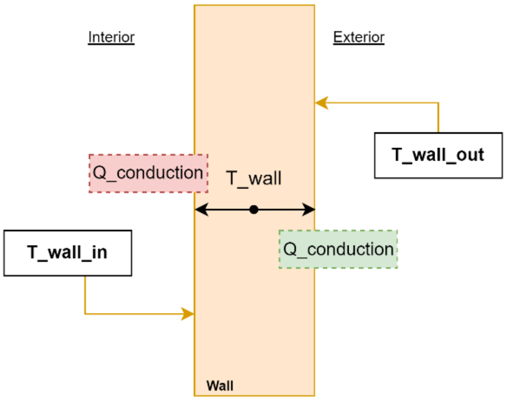


Figure 37 - Wall schematic representation

Roof/Floor/ Top and Bottom Deck

As nomenclature, the body between the ground and the zero floor is designated as floor. Between each floor there is the deck. The roof is the last deck, and its top surface is in contact with the exterior. For each body (see Figure 387), a similar heat balance to the wall is the applied, meaning that a heat balance for the bottom and upper surface of the body, as well as for the body itself.

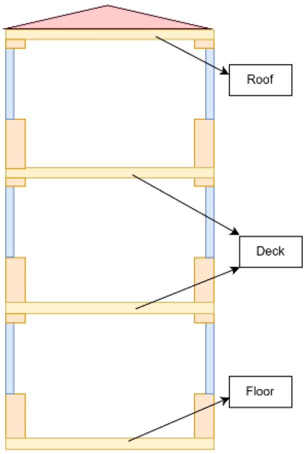


Figure 38 - Building schematic representation

Floor Top/ Deck Top

The floor and deck top are the upwards surface of the floor and deck, respectively. These horizontal surfaces facing upwards are considered to absorb the incident solar radiation. For the floor and deck top's this refers to the radiation incoming from the outside and transmitted by the windows,

$$Q_{sun_{floor}}[W] = \left(Q_{sun_{facade_N}} * A_{window_N} + Q_{sun_{facade_S}} * A_{window_S} + Q_{sun_{facade_E}} * A_{window_E} + Q_{sun_{facade_W}} * A_{window_W}\right) * \tau_{glass}$$

For the roof top, it is considered all the incident solar radiation on a horizontal surface. The steady state equation applied to the floor top surface is similar to the Inner Glass.

Deck Bottom/ Roof Bottom

The deck and roof bottom are the downwards sufaces of the deck and roof, respectively. The heat balance equation differs from the Inner glass one, because it is not considered that this surface absorbs solar radiation.
Thus,

$$Q_{conv_{int}} + Q_{radiation_{int}} = Q_{cond}$$

Roof Top

This surface is a exterior surface, and therefore the correspondent heat balance equation is similar to the Outer glass one.

Roof/Deck/Floor Body

As applied on the walls, an additional transient heat balance equation is applied for the roof, decks and floor, so that its thermal inertia is considered. The bodies will exchange heat by conduction with the correspondent surfaces. Notice that for the floor, the downwards surface is the ground, and for it a constant temperature of 15ºC was assumed.

4. Interior air

The interior air heat balance takes into account the heat exchanged between the air and all the surfaces - horizontal and vertical - it is in contact with. Moreover, the heat balance is also affected by the air renewal, infiltrations and internal gains. The heat balance is then the following,

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

$$\left(\frac{mc_pT}{dt}\right)_{interior\ air} = Q_{building\ floor} + Q_{heat\ required}$$

Where, $Q_{building\ floor}$ is an average value for a single floor, of the heat exchanged between the interior air and the surfaces it is in contact with, $Q_{conv_{average\ floor}}$, infiltrations, $Q_{infiltration}$, internal gains, $Q_{internal}$, and renewal air, $Q_{renewal\ air}$, that occur

$$Q_{building\ floor} = Q_{renewal\ air} + Q_{infiltration} + Q_{internal} + Q_{conv_{average\ floor}}$$

All the terms are considered constant for each floor, except $Q_{conv_{average\ floor}}$, which is an average of the convective heat exchange of all the floors of the building,

$$Q_{conv_{average\ floor}} = \frac{\left(Q_{conv_{top\ floor}} + Q_{conv_{middle\ floor}} * (number_{floor} - 2) + Q_{conv_{bottom\ floor}}\right)}{number_{floor}}$$

The $Q_{heat\ required}$ is related to space heating and cooling actuation, the heat or cooling needs to maintain the building's interior air temperature on the temperature intervals defined by the user. This term is described in detail in the next section.

Stream Characterization

This section focuses on the characterization of the heating and cooling the streams of the building. The streams are characterized hourly and monthly, and are used in the DHN simulation to link the building - which is a sink- to the grid. Notice that the user can confirm if the estimated monthly values are close to the building's consumption and correct them if the values are not accurate, by introducing the monthly consumption values. It is applied a coefficient for the hourly profiles according to the estimated monthly consumptions and the values given by the user.

Space Heating/Cooling Actuation

The room temperature $T_{interior}$, is meant to be restricted between set points given by the user. Usually these set points vary depending whether the buildings are occupied or not, leading to more restric temperature range or less restrictive, respectively. The implemented routine, for each time step, will estimate the needed heat/cool capacity, $Q_{heat\ required}$, in order to maintain the temperature at the desired temperature ranges. From the $Q_{heat\ required}$ is obtained the heating and cooling streams profiles that can be later utilized in the simulations is obtained .
To estimate the $Q_{heat\ required}$, the interior air heat equation balance is first solved to analyze how the temperature will shift.

$$\left(\frac{mc_pT}{dt}\right)_{interior\ air} = Q_{building\ floor}$$

The following situations may occur, considering that initially that the room temperature is within the temperature range:

1) If the computed $\mathbf{T_{interior\ air}}$ for the time step is within the defined temperature range there is no need of heating/cooling, $\mathbf{Q_{heat\ required}} = \mathbf{0}$;
2) If the computed $\mathbf{T_{interior\ air}}$ goes below the the defined temperature range, space heating is activated to provide the heating needed to maintain the temperature within the set points, $\mathbf{Q_{heat\ required}} > \mathbf{0}$;
3) If the computed $\mathbf{T_{interior\ air}}$ goes above the the defined temperature range, space cooling is activated to provide the cooling needed to maintain the temperature within the set points, $\mathbf{Q_{heat\ required}} < \mathbf{0}$;

As an example, for an occupied office building with set points of $T_{min} = 18ºC$ and $T_{max} = 24ºC$ during occupied hours, if $T_{interior\ air}$ is near 24ºC and the internal and exterior gains are such that the temperature would rise past the 24ºC, the required cooling capacity to keep it at 24ºC is computed. Similarly, if $T_{interior\ air}$ were to drop below 18ºC, the heating would actuate to keep the interior room at a minimum of 18ºC. Outside the occupied hours, the same computation occurs, however, the temperature set points are, usually less restrict, e.g. minimum of 15ºC and maximum of 28ºC, or do not exist (the user can select no set points in non-occupied hours).

If initially $T_{interior\ air}$ is outside of the temperature set points, heating or cooling will be provided if it is below or above the desired temperature range, respectively. This could lead to a large amount of heat provided in a small time step so that the temperature would meet the temperature set points. Therefore, it was added a restricition that constraints the maximum cooling/heating actuation such that the $T_{interior\ air}$ would never increase/decrease more than 1ºC per minute.

1. Domestic Hot Water

The streams characterized in the space heating/cooling actuation are complemented with the domestic hot water needs. To perform this, an expected hot water volume consumption per day is computed.

$$V_{dhw_{set}} = coef_{water} * number_{person\ per\ floor}$$

Here, $coef_{water}$ is the daily dwelling of water per person. For residential buildings and hotels $coef_{water} = 0.03$, and for offices $coef_{water} = 0.003$ [15]. Then, a defined value of domestic hot water flowrate $[m^3/s]$ for each time step is estimated,

$$\dot{m}_{dhw_{set}} = \frac{6}{60000} * number_{person\ per\ floor} * 0.5$$

And, for each time step the volume of hot water already consumed is,

$$(V_{dhw})_{new} = (V_{dhw})_{old} + \dot{m}_{dhw} * \Delta t$$

The hot water consumption will not occur in a continuous form, but as represented in the fixed intervals:

1) At 8h - 9h it is allowed consumption until a maximum of 40% of $\boldsymbol{V_{dhw_{set}}}$
2) At 12h - 13h it is allowed consumption until a maximum of 60% of $\boldsymbol{V_{dhw_{set}}}$

3) At 19h - 20h it is allowed consumption until a maximum 100% of $\boldsymbol{V_{dhw_{set}}}$
4) Outside the previous time range, no consumption of hot water is assumed

The heat needed for the hot water consumption computed for each time step is,

$$Q_{dwh} = \rho_{water} * c_{p_{water}} * \dot{m}_{dhw} * (T_{dhw} - T_{net})$$

where, T_{net} is the temperature supplied from the water distribution system.

ii. Greenhouse

This routine's objective is to apply a thermal model to simulate the hourly heat consumption over the year, based on the greenhouse parameters (see Table 44). The cooling needs are not computed since it is considered that when cooling is required the greenhouse cover is open to climatize with outside air temperature. Similar to the building routine, for each location, with the *pvlib* package, the respective climate weather data is obtained. The reasoning on the greenhouse thermal modeling approach is based on [16].

Table 44 - Greenhouse input parameters

Mandatory Input				
var name	**description**	**units**	**Range**	**Data type**
location	building location [lat,long]	º	[-90,90], [-180,180]	array
width	width of greenhouse main facade	m	[0, inf[	float
length	greenhouse length	m	[0, inf[	float
height	greenhouse height	m	[0, inf[	float
saturday_on	if working -1- or not -0- on saturday	-	0 or 1	integer
sunday_on	if working -1- or not -0- on Sunday	-	0 or 1	integer
shutdown_periods	periods of days during the year the greenhouse heating actuation is deactivated	days	[1,365]	array with arrays
daily_periods	daily hourly period the greenhouse heating actuation is activated	hour	[0-24]	array with arrays
greenhouse_orientation	greenhouse's main facade orientation	-	'N','S','E' or 'W'	string
artificial_lights_system	if the greenhouse has artificial lighting system or not	-	0;1	integer
hours_lights_needed	hours of light the plant needs (accounting with daily solar hours); only if lights_on =1	hours	[0,24]	integer

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

greenhouse _efficiency	greenhouse air infiltration tightness; 1- tight cover with low infiltrations; 2 - medium sealing; 3 - leaky cover	-	1;2;3	integer
ref_system_fuel_type	Fuel type associated; e.g. "natural_gas","electricity","bio mass","fuel_oil","none"	-	-	string
Optional input				
f_c	characterization of tightness of the cover to air infiltration [17]	-	[0, 20 * 10^(-4)]	float
T_heat_on	minimum allowed greenhouse interior air temperature for which the space heating starts actuating	ºC	[0,inf[	float
supply_temperature_heat	heaters working fluid temperature; supply temperature < target temperature (supply temperature to the DHN heat exchanger)	ºC	]0,100[; must be lower than target_temperature_heat	float
target_temperature_heat	heaters working fluid temperature; target temperature < supply temperature (temperature obtained from the DHN heat exchanger)	ºC	]0,100[; must be higher than supply_temperature_heat	float
leaf_area_index	average leaf area index of a plant	-	[0,inf[	float
rh_air	relative humidity	-	]0,1[	float
u_cover	cover thermal conductivity	W/m.K	[0,inf[	float
indoor_air_speed	indoor air velocity	m/s	[0,inf[	float
leaf_length	characteristic length of a plant leaf	m	[0,inf[	float
tau_cover_long_wave_radiation	Cover transmissivity coefficient to long-wave radiation	-	[0,1]	float
emissivity_cover_long_wave_radiation	Cover emissivity coefficient to long-wave radiation	-	[0,1]	float
tau_cover_solar_radiation	transmissivity coefficient to solar radiation	-	[0,1]	float
power_lights	light power per square meter	W/m^2	[0,inf[	float
real_monthly_capacity	Real monthly data - for each month of the year	kWh	[0,inf[	array
real_yearly_capacity	Real yearly data - single value	kWh	[0,inf[	float

1. Step by Step

The thermal model includes solving heat balance equations, based on heat transfer mechanisms which are affected by the climate conditions and greenhouse parameters. At each time step the heating needs for the greenhouse interior air are computed. The step by step methodology implemented is:

1. **Get greenhouse parameters**
2. **For each time step**
    a. **Get climate data**
    b. **Climate data treatment**
    c. **Compute incident solar radiation**
    d. **Compute evapotranspiration heat**
    e. **Compute interior radiation heat transfer**
    f. **Compute sky radiation heat transfer**
    g. **Compute interior convection heat transfer**
    h. **Compute ground losses**
    i. **Compute infiltrations**
    j. **Compute internal heat gains**
    k. **Compute space heating actuation**
    l. **Solve greenhouse interior air heat balance**
    m. **Save heating actuation power**
    n. **Return to step a) if not yet in the last time step**
3. **Return monthly and hourly heating needs**

Heat Transfer

This section analyses all the heat transfer items that have an impact on the greenhouse thermal simulation.

1. Solar Radiation

Similarly to the buildings the absorption of incident solar radiation by greenhouses is given by,

$$Q_{sun_{greenhouse}} = \left(Q_{sun_{wall_N}} * A_{wall_N} + Q_{sun_{wall_S}} * A_{wall_S} + Q_{sun_{wall_E}} * A_{wall_E} + Q_{sun_{wall_W}} * A_{wall_W} + Q_{sun_{roof}} * A_{floor}\right) * \tau_{cover} * \alpha_{greenhouse}$$

where, the greenhouse average solar radiation absorption coefficient, $\alpha_{greenhouse}$, is assumed to be 0.75 [16].

2. Inner/Outside Convection

The heat lost through the cover is due to: convection heat between interior air and cover, conduction heat on the cover, and outer convection between cover and exterior. The overall heat transfer coefficient can be computed as,

$$u_{total}\left[\frac{W}{m^2.K}\right] = \left(\frac{1}{h_{interior}} + \frac{1}{u_{cover}} + \frac{1}{h_{exterior}}\right)^{(-1)}$$

The total heat transfer between interior air and exterior is,

$$Q_{lost_{exterior}} = u_{total} * A_{cover} * (T_{exterior} - T_{interior})$$

3. Long-wave Radiation

This transmitted radiative heat loss from a greenhouse can be expressed as follows,

$$Q_{rad_{lost}} = Q_{rad_{sky}} + Q_{rad_{ground}}$$

where,

$$Q_{rad_{greenhouse-sky}} = \sigma_{Boltzmann} * \varepsilon_{cover} * F_{cover-sky} * (T_{interior}^4 - T_{cover}^4)$$

$$Q_{rad_{ground-cover}} = \sigma_{Boltzmann} * \varepsilon_{plants} * F_{ground-cover} * (T_{interior}^4 - T_{cover}^4)$$

The cover temperature is typically a linear function of the indoor and the outdoor temperature. The following linear function can be used to solve cover temperature [16]

$$T_{cover} = \frac{2}{3} T_{exterior} + T_{interior}^{\frac{1}{3}}$$

It was considered that the view factor between greenhouse and sky, and ground and cover is approximately 1.

4. Ground conduction

The ground surface, considered to be at interior air temperature, loses heat by conduction to the ground under it.

$$Q_{lost_{ground}} = u_{ground} * A_{floor} * (T_{ground} - T_{interior})$$

, T_{ground} is considered to be 15ºC.

5. Evapotranspiration

Evapotranspiration represents the evaporation of water from the plants, and it is responsible for a significant amount of heat loss in greenhouses. A simplified approach is done to estimate the heat exchanged due to plant evapotranspiration Q_{plants},

$$Q_{plants} = \dot{m}_{evap} * \Delta H_{water}$$

where, ΔH_{water} is the water latent heat and $\dot{m}_{evap}$ the mass flowrate of evaporation, which is given by

$$\dot{m}_{evap} = A_{plants} * \rho_{air} * \frac{w_{plant} - w_{air}}{R_{aerodynamic} + R_{stomatal}}$$

where, A_{plants} is the plant's leaf surface area, w_{plant} the humidity ratio of plants and w_{air} of indoor air, $R_{aerodynamic}$ the aerodynamic resistance and $R_{stomatal}$ the stomatal resistance.

For the humidity ratios,

$$w_{plant} = 0.6219 * \frac{p_{v_{plants}}}{P_a - p_{v_{plants}}}$$

$$w_{air} = 0.6219 * \frac{p_{v_{air}}}{P_a - p_{v_{air}}}$$

where, p_v is the saturated vapour pressure and P_a the atmospheric pressure. It is considered that plants and indoor air are at the same temperature, and that the relative humidity of the air is given by the user or it is obtained a default value from the KB. The A_{plants} is obtained as,

$$A_{plants} = A_{floor} * LAI$$

where, LAI is leaf area index of the plants in the greenhouse.

The aerodynamic resistance and stomatal resistance are computed as,

$$R_{aerodynamic} = 220 * \frac{L_f^{0.2}}{v_{interior}^{0.8}}$$

$$R_{stomatal} = 200 * \left(1 + \frac{1}{e^{\left(0.05 * \left(Q_{sun_{roof}} * \tau_{cover} - 50\right)\right)}}\right)$$

where, L_f is the plant's leaf characteristic length and $v_{interior}$ the indoor air velocity.

6. Lights

Lights may be used in the greenhouse to guarantee a minimum number of lighting hours for the plants. A significant amount of heat can be added by lights and therefore it must be computed,

$$Q_{lights}\,[W] = P_{lights} * A_{floor}$$

where, P_{lights} the power of lights per square meter.

For each day, it is analysed the number of hours with incident solar radiation larger than zero, which means that there is natural lighting. If the number of lighting hours desired by the user is larger than the number of hours there is incident solar radiation, it is considered that the lighting equipment operates after the last time step the incident solar radiation is zero until the number of total lighting hours is met.

7. Infiltrations

The model used for computing infiltrations is a modification of building air leakage model [17]. The heat exchanged due to infiltrations, is,

$$Q_{infiltrations}\,[W] = \rho_{air} * c_{p_{air}} * ACS * \left(T_{exterior} - T_{interior}\right)$$

where, ACS is the air change per second, and computed as,

$$ACS\,[1/s] = A_{infiltration} * \sqrt{\left(c_w^2 * wind_{speed} + f_t^2 * \left(|T_{interior} - T_{exterior}|\right)\right)}$$

And c_w is the average wind pressure coefficient (assumed c_w =0.22) and f_t is the temperature difference factor (assume f_t = 0.16). The $wind_{speed}$ is obtained from the climate data for each time step.

Finally,

$$A_{infiltration} = A_{total\ cover} * f_c$$

The f_c is the characterization of tightness of the cover to air infiltration. The largest it is, the less tight and the more exterior air leakages the greenhouse has.

Heat Balances

The interior air heat balance considers the heat exchanged between the air and all the surfaces - horizontal and vertical - it is in contact with. Moreover, the heat balance is also affected by infiltrations, internal and solar gains, heat exchanged by radiation, evapotranspiration, and artificial lighting, if existent. The heat balance is the following,

$$\left(\frac{mc_pT}{dt}\right)_{interior\ air} = Q_{greenhouse} + Q_{heat\ required}$$

where,

$$Q_{greenhouse} = Q_{lights} + Q_{infiltrations} + Q_{conv_{total}} + Q_{lost_{ground}} + Q_{lost_{exterior}} + Q_{sun_{greenhouse}} * \alpha_{greenhouse} - Q_{plants} + Q_{rad_{lost}}$$

Stream Characterization

This section focuses on the characterization of the heating stream of the greenhouse. The stream is characterized hourly and monthly, and is used in the DHN simulation to link the greenhouse - which is a sink- to the grid. Similarly, to the building, the monthly heat consumption, if known, can be given by the user to adjust the estimated hourly heating needs.

A similar routine as described in section i (Buildings – Residential, Hotel and Office) was implemented0. As mentioned at the beginning of the greenhouse section, only the space heating actuation is evaluated, and not the cooling.

2.1.6.5.2 Simulation

a. Convert DHN

This routine designs the conversion technologies necessary so that the DHN can meet the sinks' heating/cooling needs. For a given group of sinks, the DHN supply and return temperatures are set by the sink which demands the largest temperature. Initially, grid specific technologies are designed to meet the heating/cooling requirements of the group of sinks. The DHN specific technologies are technologies designed to cover the heating and cooling needs of whole group of sinks. Then, for each sink stream the conversion technologies are designed.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

i. Implemented Routine

Whenever the user selects an area to be analyzed, the group of sinks is provided to the CF *convert_sink* function to design the technologies that will convert the heat from the DHN to each sink (see Table 45 and Table 46). The output of *convert_sink* is the group grid supply and return temperatures for each excess heat stream its supply heat needed [kW], the consumption profile [kWh/h], the CO_2 emissions and linearized costs of the turnkey [€], fixed [€/year] and variable [€/MWh] operation and maintenance costs, for the conversion technologies designed (see Table 47 to Table 52).

Table 45 - Convert Sinks input

Mandatory Input				
var name	**description**	**units**	**range**	**data type**
group_of_sinks	array with the sinks to be converted	-	-	array with dictionary
grid_supply_temperature	Grid supply temperature	ºC	[5,110]	float
grid_return_temperature	Grid return temperature	ºC	[5,110]	float

Table 46 - Convert Sinks input; group of sinks

group_of_sinks – example of sink				
var name	**description**	**units**	**range**	**data type**
id	sink ID	-	-	integer
location	sink location as [latitude,longitude]	º	[[-90,90],[-180,180]]	array
streams	(see Table 10)	-	-	array with dictionaries
fuels_data	Fuels price [€/kWh] and CO2 emission [kg CO2/kWh] for: natural gas, fuel oil, electricity, biomass.	-	-	dict

Table 47 - Convert Sinks output

Output				
var name	**description**	**units**	**range**	**data type**
sink_group_grid_supply_temperature	DHN supply temperature	ºC	[0,120]	float
sink_group_grid_return_temperature	DHN return temperature	ºC	[0,120]	float
grid_specific	dictionary with heating and cooling technologies	-	-	dictionary
sinks	array with sinks and respective conversions	-	-	dictionary

Table 48 - Convert Sinks output; grid specific dictionary

grid_specific				
var name	**description**	**units**	**range**	**data type**
heating	array with heating technologies for the sink group	-	-	array with dictionaries
cooling	array with cooling technologies for the sink group	-	-	array with dictionaries

Table 49 - Convert Sinks output; heating/cooling grid specific dictionaries

grid specific - Heating/cooling dictionaries				
var name	**description**	**units**	**range**	**data type**
teo_equipment_name	TEO output; names for technologies packages given by TEO (e.g. 'single_heat_exchanger' , 'multiple_heat_exchanger',…)	-	-	string
output	TEO output; default:1	-	1	integer
input_fuel	TEO output; default: 'excess_heat'	-	'excess_heat'	string
output_fuel	TEO output; default: 'dhn_water_source'	-	'dhn_water_source'	string
equipment	array with technologies name (e.g. ['hx_plate', 'heat_pump','hx_plate'])	-	-	-
max_capacity	maximum power stream can provide	kW	[0,inf[	float
turnkey_a	turnkey a value (y=ax+b)	€/kW	[0,inf[	float
turnkey_b	turnkey b value (y=ax+b)	€	[0,inf[	float
conversion_efficiency	excess heat to usable heat conversion efficiency	-	[0,1]	float

Table 50 - Convert Sinks output; sinks

sinks				
var name	**description**	**units**	**range**	**data type**
sink_id	sink ID	-	-	integer
streams	array with sink's streams	-	-	array with dictionaries

Table 51 - Convert Sinks output; stream

streams				
var name	**description**	**units**	**range**	**data type**
stream_id		-	-	integer
hourly_stream_capacity	array with hourly stream capacity	kWh	-	array
conversion_technologies	multiple dictionaries with technologies possible to implement	-	-	array with dictionaries

Table 52 - Convert Sinks output; conversion technologies

conversion_technologies – example of dictionary				
var name	**description**	**units**	**range**	**data type**
teo_equipment_name	TEO output; names for technologies packages given by TEO (e.g. 'single_heat_exchanger' , 'multiple_heat_exchanger',…)	-	-	string
output	TEO output; default:1	-	1	integer
input_fuel	TEO output; default: 'excess_heat'	-	'excess_heat'	string
output_fuel	TEO output; default: 'dhn_water_source'	-	'dhn_water_source'	string
equipment	array with technologies name (e.g. ['hx_plate', 'heat_pump','hx_plate'])	-	-	-
max_capacity	maximum power stream can provide	kW	[0,inf[	float

turnkey_a	turnkey a value (y=ax+b)	€/kW	[0,inf[	float
turnkey_b	turnkey b value (y=ax+b)	€	[0,inf[	float
conversion_efficiency	excess heat to usable heat conversion efficiency	-	[0,1]	float
om_fix	fix operation and maintenance costs	€/year.kW	[0,inf[	float
om_var	variable operation and maintenance costs	€/kWh	[0,inf[	float
emissions	CO_2 emissions per kWh	kg CO2/kWh	[0,inf[	float
technologies	array with technologies designed dictionaries; each dictionary contains detailed info about the technology	-	-	array with dictionaries

1. Grid Specific

As mentioned, firstly, the grid specific technologies are designed to meet the needs for the whole group of sinks. The grid specific technologies considered are: boiler, solar thermal and heat pump. Grid specific technologies are designed by evaluating the sinks group total capacity and the stream's temperatures. As an example, if there are two sinks' streams with the same schedule, one with a heating capacity need of 500kW at 70ºC, and the other with 250kW at 60ºC; the grid specific heating technology - boiler, solar thermal or heat pump - is designed for the nominal capacity of the aggregated streams schedule and to the largest temperature needed. In this case, 750kW provided at 70ºC.

2. Sinks' streams

The sinks can have three types of streams circuits: steam, hot water and chilled water. According to these streams' circuits, when performing the conversion for the sinks' streams, four main design options may occur.

For **steam** circuits:

Case 1) a heat pump is designed to provide steam to the sink, as well as a grid-sink HX and correspondent circulation pumping. The thermal energy supplied to the evaporator of the heat pump is given by the DHN;

For **hot water** circuits:

Case 2) if the DHN temperature meets the sink target temperature requirements, only a grid-sink HX and correspondent circulation pumping is designed;

Case 3) if the DHN temperature does not meet the sink target temperature requirements, grid-sink HX and correspondent circulation pumping are designed, as well as a heating technology to raise the grid temperature to meet the sink required temperature (heating technologies: solar thermal, heat pump, boiler).

For **chilled water** circuits, it is necessary to design an absorption/thermal chiller to convert the DHN heat into cooling:

Case 4) if the DHN temperature meets the absorption chiller generator temperature requirements and it can provide the chilled water temperature required, only the absorption chiller and correspondent circulation pumping are designed; if it cannot provide the chilled water temperature required, an absorption together with an electric chiller and correspondent circulation pumping are designed; if the DHN temperature does not meet the absorption chiller generator temperature requirements and it cannot provide the chilled water temperature required, an absorption together with an electric chiller and hot water boiler, and correspondent circulation pumping are designed;

In brief, the possible technologies designed for the conversion are: Heat Pump, HX, HX + heating/cooling technology. For all the design solutions, the circulation pumping required for the grid is also designed. All design packages data is collected and sent in a standardized form to the TEO module (script: *join_hx_and_technology*). The paragraphs below provide a few examples for the the design options.

Case 1)
When the sink desires steam, the only option dimensioned to utilize heat from the DHN and provide steam, is the heat pump (see Figure 39). As an example, considering that the DHN supplies hot water at 90ºC and returns at 50ºC, and a sink's stream with the following properties:

- circuit type: steam
- fluid: steam
- $\boldsymbol{T_{supply} = 40ºC}$
- $\boldsymbol{T_{target} = 150ºC}$
- $\boldsymbol{P_{stream} = 300\ kW}$

As an approximation, the supply capacity of the heat pump is considered to be the stream capacity since the phase change consumes a much larger amount of energy than heating the condensates until boiling point. Therefore, the heat pump capacity is,

$$P_{supply_{HP}} = P_{stream}\,[kW] = 300$$

It is assumed that the condenser supply temperature is equal to the stream temperature and evaporator temperature equal to the DHN return temperature. The power needed from the grid is the heat pump evaporator capacity which is computed according to the heat pump's EER - Energy efficiency ratio - and supply capacity

$$P_{DHN} = P_{evap_{HP}} = P_{supply_{HP}} * \left(1 - \frac{1}{EER}\right)$$

where,

$$EER = 0.55 * \frac{T_{cond_{HP}} + 273 + 5}{T_{cond_{HP}} - T_{evap_{HP}} + 10}$$

With the heat pump supply and evaporator capacity, and condenser and evaporator temperatures it is possible to design and cost the heat pump, as well as the grid circulation pumping.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

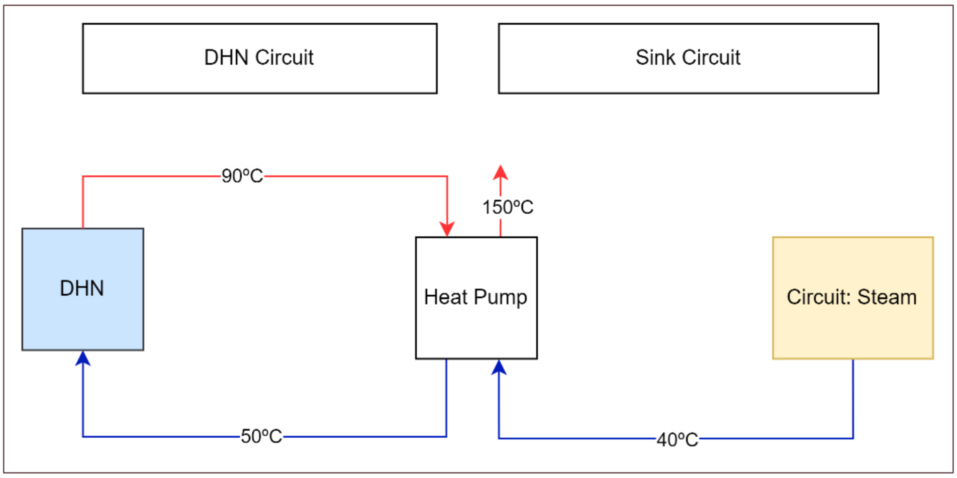


Figure 39 - Convert Sink; Case 1 – Heat Pump diagram

Case 2)

When the DHN temperature is larger, or equal, to the sink's hot water stream target temperature plus ΔT_{HX} (assumed that $\Delta T_{HX} = 5ºC$), only a heat exchanger network is designed. As an example, considering that the DHN supplies hot water at 90ºC and returns at 50ºC, and a sink's stream with the following properties:

- circuit type : hot water
- fluid: water
- $\boldsymbol{c_p = 4.2\ kJ/kg.K}$
- $\boldsymbol{T_{supply} = 20ºC}$
- $\boldsymbol{T_{target} = 80ºC}$
- $\boldsymbol{\dot{m}_{stream} = 200\ kg/h}$

Since the stream's T_{target} is inferior to the DHN supply temperature minus ΔT_{HX} (see Table 533), only a heat exchanger is designed (see Figure 40).

Table 53 - Convert Sink; Case 2 - HX design temperatures

	Sink Stream	DHN
Fluid	water	water
T_{cold}	20	50
T_{hot}	80	90

To compute the capacity of the HX, first, it is computed the capacity that the stream needs,

$$P_{stream}\,[kW] = \dot{m} * c_p * \frac{\Delta T}{3600} = 200 * 4.2 * \frac{(80-20)}{3600} = 7$$

The heat exchanger designed capacity is,

$$P_{HX_{DHN-stream}}[kW] = \frac{P_{stream}}{\eta_{HX}} = \frac{7}{0.95} = 7.36$$

With the working fluids' temperatures, presented in Table 533, and the computed HX capacity, it is possible to design the appropriate HX.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

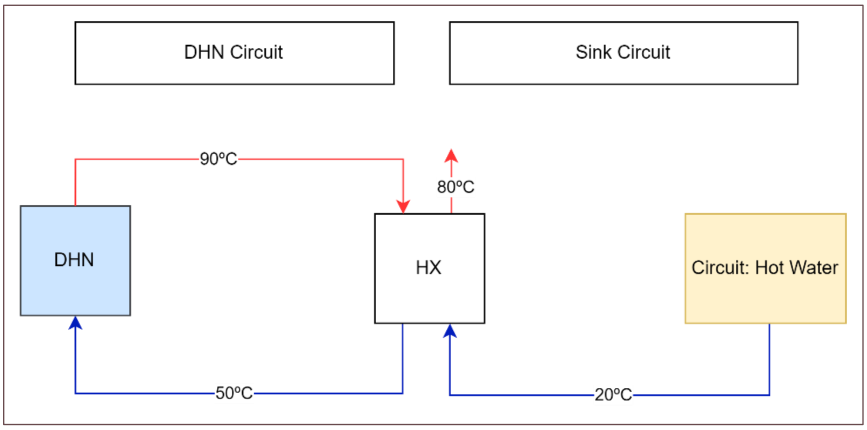


Figure 40 - Convert Sink; Case 2 – HX network diagram

Case 3)

When the DHN supply temperature is not enough to meet the required sink's hot water stream target temperature, it is necessary to design a heating technology to boost its temperature. The heating technologies that can be designed to increase the hot water circuit temperature are: boiler, heat pump, and solar thermal.

The procedure is similar to the one done in section 1.1.2.2.1.1. After designing the HX, the thermal capacity required to reach the stream's target temperature is estimated, and the heating technology designed.

As an example, utilizing the data described in Table 54, where a DHN supplies hot water at 70ºC which returns at 40ºC, and a sink's hot water circuit stream starting at 20ºC with a target temperature of 80ºC. The DHN supply temperature is inferior to the stream's target temperature, thus a heating technology must be designed (see Figure 41).

Table 54- Convert Sink; Case 2 – HX + Boiler design temperatures

	Sink Stream	**Grid Stream**
Fluid	water	water
T_cold	20	40
T_hot	80	70

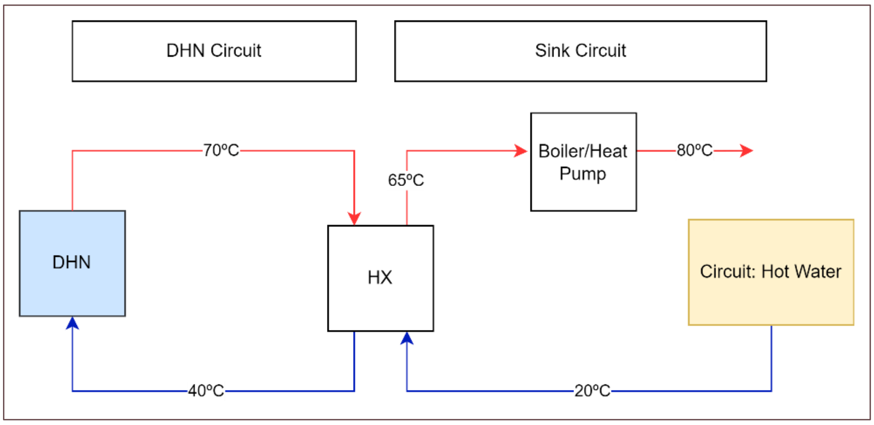


Figure 41 - Convert Sink; Case 2 – HX + Boiler/Heat Pump diagram

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

Similarly, when designing the solar thermal on the sources side, a solar thermal can also be designed to increase the sink's circuit stream temperature. A backup technology, such as a heat pump or a boiler, is also designed (see Figure 42) to guarantee that the circuit stream is always fulfilled.

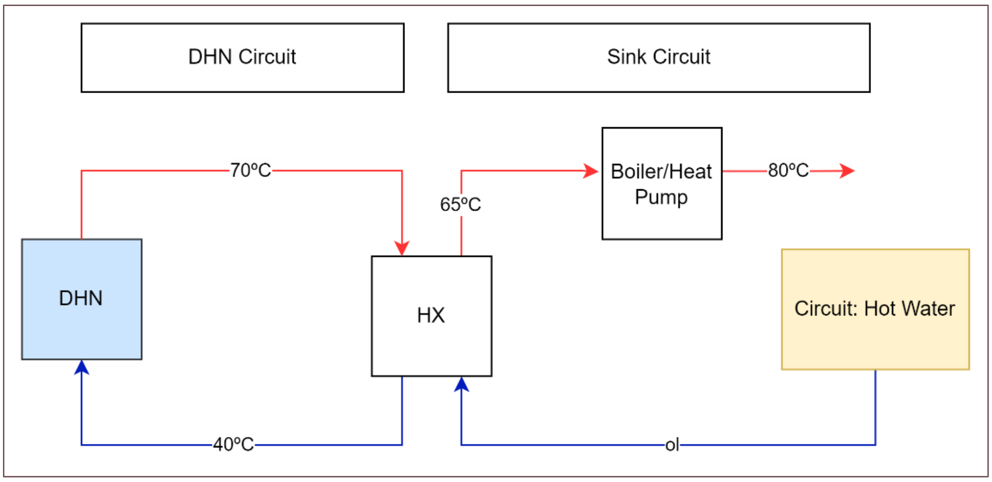


Figure 42- Convert Sink; Case 2 – HX + Solar thermal with backup diagram

Case 4)

When a sink requires cooling, cooling technologies have to be designed, in addition to the heat exchanger, to convert the heat from the grid. When cooling is necessary an absorption chiller is designed with the following assumptions:

- $T_{in_{generator}} = 85^{\circ}C$ (minimum temperature of the stream provided to the absorption chiller's generator);
- $T_{in_{generator}} = 70^{\circ}C$;
- $T_{out_{evaporator}} = 7^{\circ}C$ (minimum temperature of the stream leaving the absorption chiller's evaporator);

These assumptions lead to the folowing:

1) if the the DHN temperature is below the thermal chiller evaporator temperature a boiler will be designed to increase its temperature until the absorption chiller generator temperature is met;
2) if the sink stream requires a lower temperature than the minimum thermal chiller supply temperature, an electric chiller will be designed to fullfill the sink's stream desired temperature.

As an example, considering that the DHN supplies hot water at 80ºC and returns at 50ºC, and a sink's chilled water circuit with the following properties:

- circuit type : chilled water
- fluid: water
- $c_p = 4.2\ kJ/kg.K$
- $T_{supply} = 12^{\circ}C$
- $T_{target} = 7^{\circ}C$
- $\dot{m}_{stream} = 50000\ kg/h$

As an approximation, the supply capacity of the absorption chiller is considered to be the cooling capacity required by the stream.

$$P_{supply_{absorption\ chiller}}[kW] = P_{stream}\ [kW] = \dot{m} * c_p * \Delta T = 50000 * 4.2 * \frac{12-7}{3600} = 292$$

From the KB, the COP of the absorption chiller is obtained to compute the generator required thermal capacity,

$$P_{DHN} = P_{generator_{absorption\ chiller}}[kW] = \left(\frac{P_{supply}}{COP}\right)_{absorption\ chiller}$$

With power and temperatures characterized, it is possible to cost and dimension the absorption chiller and the circulation pumping.

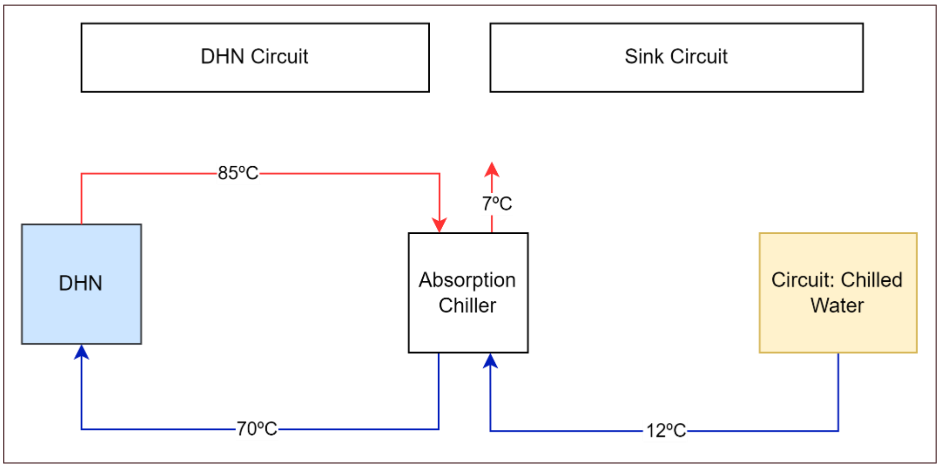


Figure 43 - Convert Sink; Case 3– Absorption chiller diagram

If the sink's circuit temperature is lower than minimum temperature provided by the absorption chiller, and additional electric chiller is designed to further decrease the temperature. Since the stream flowrate, supply and target temperatures is known, it is possible to design the electric chiller, as represented in Figure 44.

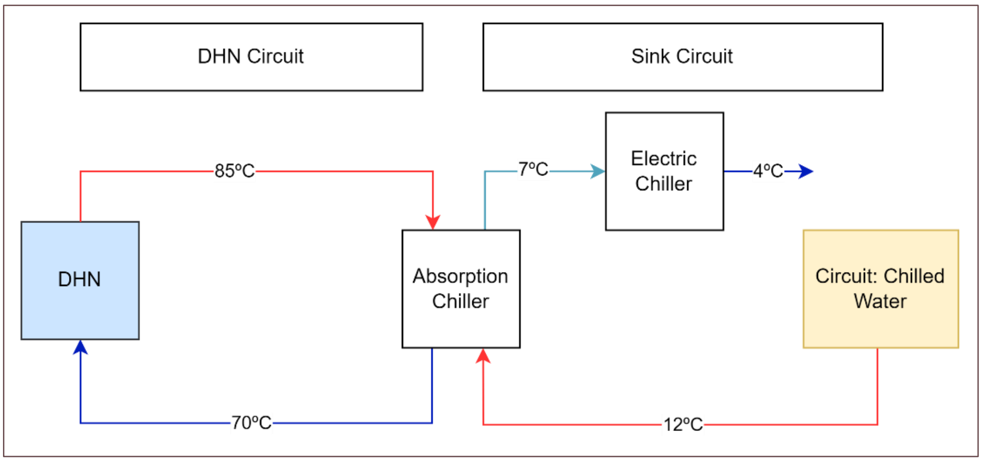


Figure 44 - Convert Sink; Case 3– Absorption chiller + Electric chiller diagram

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

Finally, if the DHN supplies hot water at an inferior temperature to the absorption chiller evaporator temperature, e.g. DHN supplying at 80ºC, a boiler is designed to increase the temperature of the working fluid to the absorption chiller evaporator temperature (see Figure 45). The minimum temperature of DHN to which this solution is designed is 80ºC, since below this the boiler provides all the necessary heat and no heat is provided by the DHN.

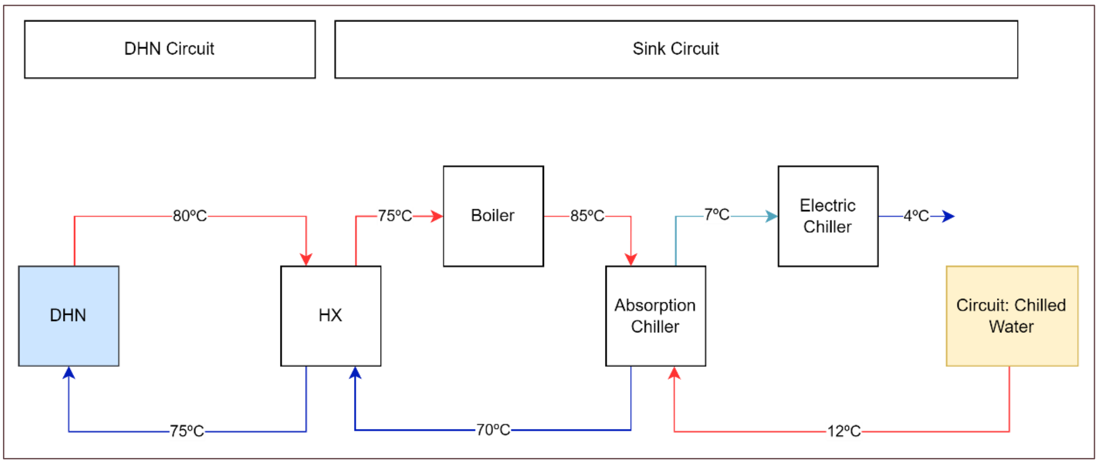


Figure 45 - Convert Sink; Case 3 – Boiler + Absorption chiller + Electric chiller diagram

2.1.6.5.3 Validation Test Cases

The validation test cases were done only on the routines which perform several iterations and computations or present a more complex thermal model: pinch analysis (script:*convert_pinch*), and the building (script:*building*) and greenhouse (script:*greenhouse*) characterization. Purely characterization routines do not have a validation test case presented in this document, since the computations are fairly simple and the user introduces most of the data. The simulations routines which make the conversion of heat from the the sources to the DHN (script: *convert_sources*) and from the DHN to the sinks (script: *convert_sinks*) are also not presented here due to the fact that the conversion computations are straightforward.

a. Simulation: Pinch analysis (script: convert_pinch)

Regarding the pinch analysis routine, two validation tests, a simple – Case Test 1– and more complex one – Case Test 2 -, are presented in this section. Nonetheless the routine was also validated in more examples ([4] [5] [2]).

Validation Test 1 – Simple Case
Simple cases, as the one presented in Figure 46, are solved quickly. Simple cases are those which do not possess a large number of streams and/or the routine can make straightforward HX designs. The larger number of streams the routine receives, the longer it takes to provide solutions. This because, the routine performs pinch analysis to all the combinations possible between streams, and due to the fact that the more streams are split and special cases exist (see section 320), the longer and iteration-intensive it becomes to achieve a HX network design.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

From the input data presented in Figure 46, the routine provided the results shown in Figure 47 and Figure 48.

Type	Supply T (°C)	Target T (°C)	F*Cp (MW °C^{-1})
Hot	750	350	0.045
Hot	550	250	0.04
Cold	300	900	0.043
Cold	200	550	0.02

ΔT_{min}=50 °C

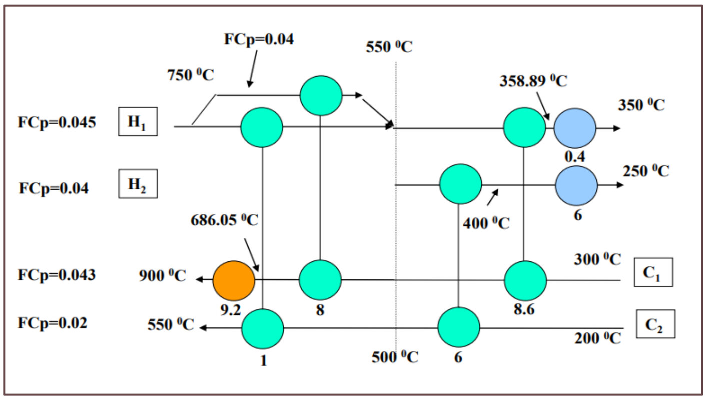


Figure 46 - Pinch Analysis validation test Case 1; Input data (left) and representation of the designed HX network [4]

When comparing the output data from the pinch analysis routine and the expected HX network, it can be seen that the results from Figure 476 are equal to the ones from Figure 465. The routine also provides an additional alternative and feasible HX network to the user (Figure 48). Even though, in this document only the output with all the streams are shown, the routine also analysed heat recovery in all streams combination to find more optimal solutions. The total running time for all streams' combinations was of 1.92s.

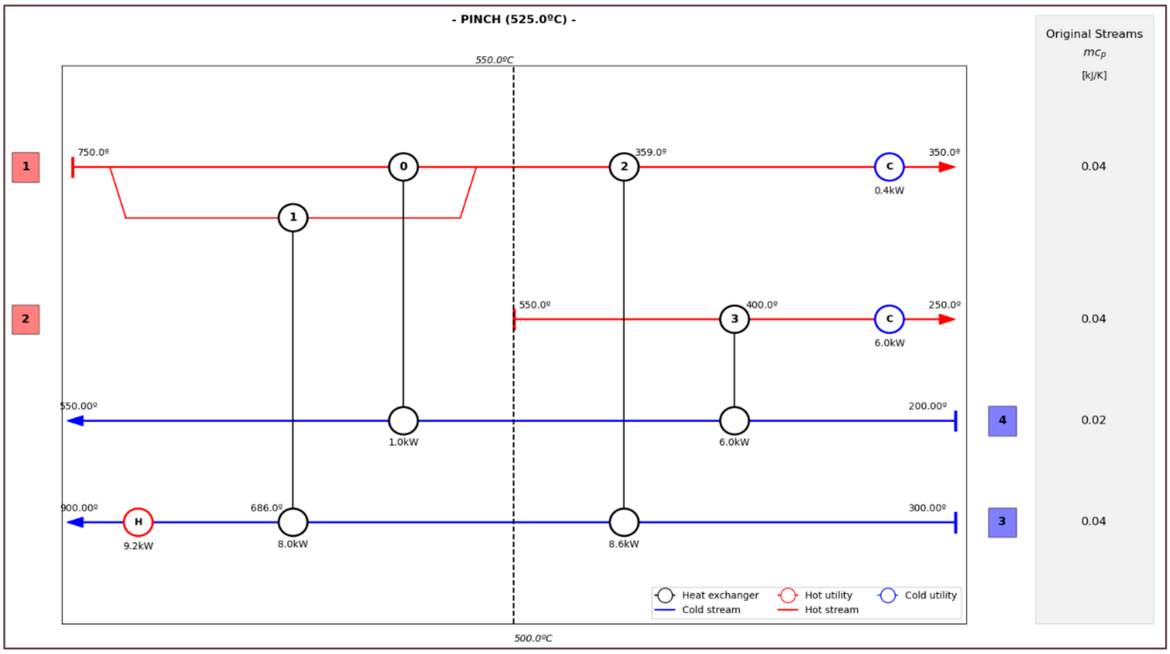


Figure 47 - Pinch Analysis validation test Case 1; Routine output data example 1

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

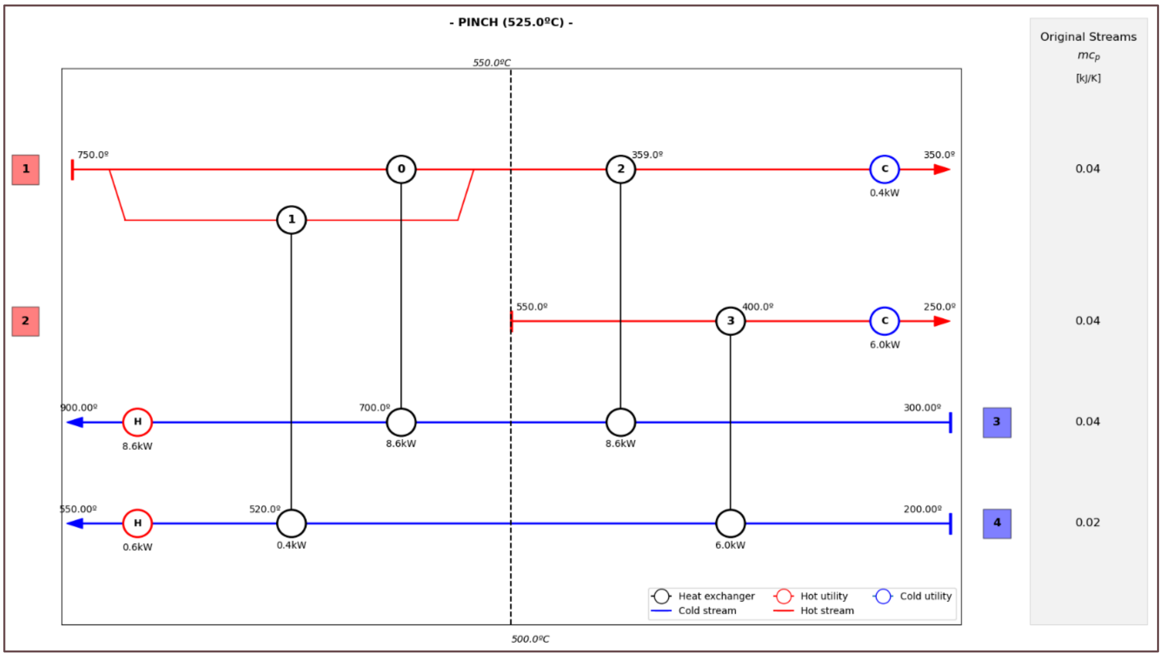


Figure 48 - Pinch Analysis validation test Case 1; Routine output data example 2

Validation Test 2 – Complex Case

The case presented in Figure 49 is a more complex case. It has a total of 7 streams and more importantly it does not have straightforward HX designs - the proposed network requires multiple splits - which will incur in a large number of iterations to achieve a solution.

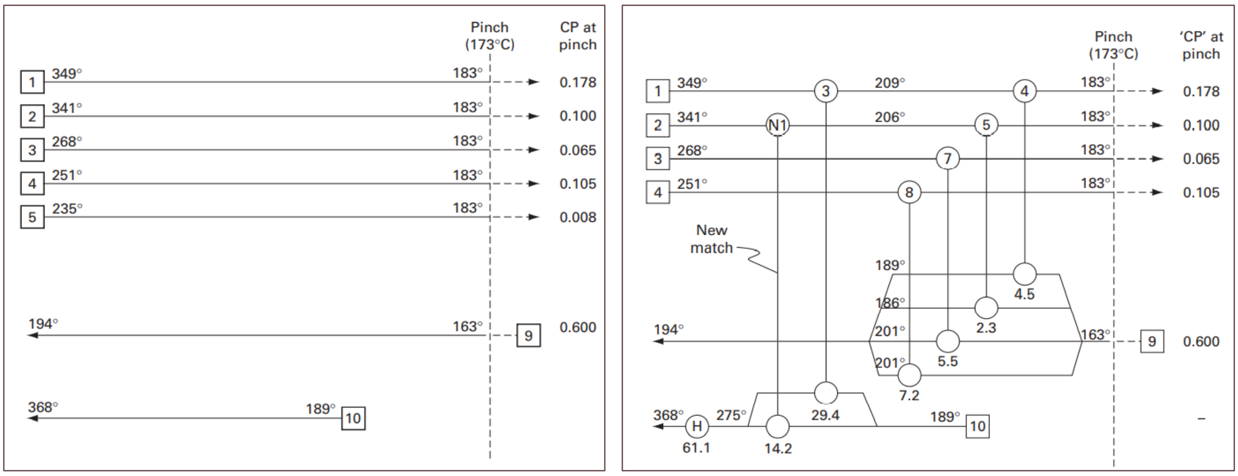


Figure 49 - Pinch Analysis validation test Case 2; Input data (left) and representation of the designed HX network [2]

When comparing the output data from the pinch analysis routine, presented in Figure 50, and the expected HX network, it can be seen that the routine presents a similar, however, not exactly equal result. This is due to the simplifications implemented in the algorithm. As an example, in Figure 49 some splits of stream 9 achieve a higher temperature, when matched, than the target temperature of stream 9. The target temperature of stream 9 is 194ºC, however some split streams reach 201ºC, and only when combined with the remaining splits respect the stream 9 target temperature. The

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

implemented algorithm does not consider this specific situation which leads to slightly different HX network design.

To obtain the solution of this complex case, without considering all streams combinations - which would consume a larger amount of time-, the total running time was of 891.6 s (14min 5s).

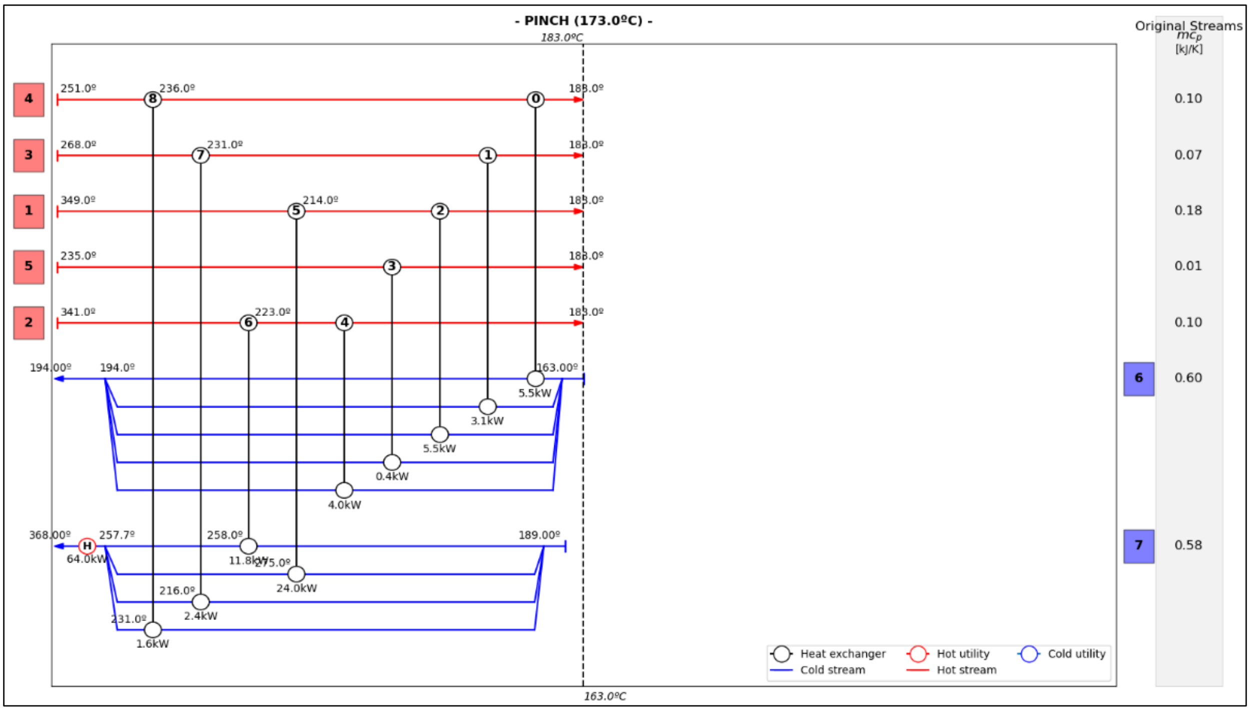


Figure 50 – Pinch Analysis validation test Case 2; Routine output data example

b. Characterization: Building (script: building)

Two test cases were utilized to validate the results obtained from the '*building*' routine. The first validation case is done on theoretical data of the literature [14], which simulates a building heating and cooling needs with Energy Plus, and the second validation was performed on the INEGI building. The aim of the '*building*' routine is to create a heating/cooling hourly profile. Perceptibly, due to the fact that this routine is a simplified model, the estimates will not be precise, however it can provide a good representation of a building heating/cooling needs. After the characterization routine is performed, the user, as explained in section 0, can a correct the estimated heating and cooling needs.

Validation Test 1 – Building (theoretical data)

The first validation data test was done on data from the literature [14]. The building analysed "(…) is a three-story high narrow plan office building with a 32 by 16 meters footprint and floor-to-ceiling height of 3.5 meters (…)" and "Each floor is divided into two zones. First zone is a large open office area while second zone represents common spaces such as corridors, toilets, tea kitchen, etc. (…)" The building is represented in Figure 51. The authors of the paper consider two distinct areas per floor - zone 1 and zone 2 - that have different set point temperatures. Zone 1 is considered as an open office, occupied during working hours, and zone 2 as an uncopied area. The '*building*' routine, however, due to its simplicity does not consider different zones. Therefore, it was assumed that the whole building was of type Zone 1.

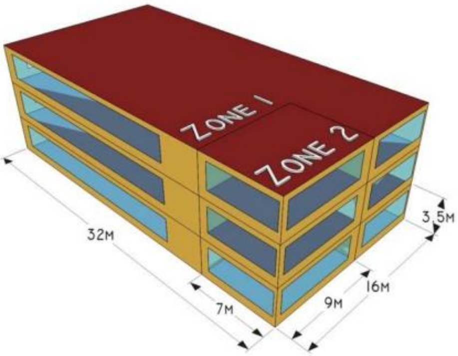


Figure 51 - Representation of the office building [14].

The authors analyse a less and more energy efficient building, whose efficiency differs due to the building material's thermal properties. The common data of both buildings can be observed in Table 55, and the thermal properties of the less and more efficient building can be observed in Table 56 and Table 57, respectively.

Table 55 - Building Validation Test 1; Input data

Input			
parameter		**value**	**unit**
width_floor		32	m
length_floor		16	m
height_floor		3.5	m
ratio_N_wall		0.5	-
ratio_S_wall		0.5	-
ratio_E_wall		0.5	-
ratio_W_wall		0.5	-
floor_number		3	-
building_orientation		South	-
daily periods		[7,19]	h
saturday_on		0	-
saturday_on		0	-
shutdown		August – [213, 243]	-
occupied set temperatures	T_cool_on	24	ºC
	T_heat_on	22	ºC
non-occupied set temperatures	T_off_min	12	ºC
	T_off_max	28	ºC
equipment + artificial lighting		27	W/m^2
renewal_air_per_person		10	L/s
air_change_hour		0.3	1/h

Table 56 - Building validation Test 1; Input data of the less efficient building

Input - Building 1		
parameter	**value**	**unit**
u_wall	0.53	W/m^2.K
u_roof	0.45	W/m^2.K
u_ground	0.84	W/m^2.K
u_glass	3.21	W/m^2.K

Table 57 - Building Validation Test 1; Input data of the more efficient building

Input - Building 2		
parameter	**value**	**unit**

u_wall	0.25	W/m^2.K
u_roof	0.15	W/m^2.K
u_ground	0.15	W/m^2.K
u_glass	1.98	W/m^2.K

The solar radiation and hourly temperature data, instead of being obtained from the *pvlib*, was obtained from the EnergyPlus website - London-Gatwick weather file - so that the values would represent with more accuracy the values used by the authors. However, notice that it is not known for sure if the actual temperatures and solar radiation data used by the author was the one used.

Tables Table 58 and Table 59 show the building heat and cooling loads obtained by the authors and the estimated loads by the '*building*' routine. Comparing the yearly building loads of the literature and the ones simulated for both buildings, it can be seen that the '*building'* routine can provide reasonable heating and cooling loads, and are approximate to the EnergyPlus results. The differences existing between models can be, in addition to the simplifications in the algorithm model, because it was considered the whole building as being of Zone 1, and due to the used climate data.

Table 58 - Building Validation Test 1; Comparison between the literature [14] and the estimated loads of the less efficient building

	Literature results	**Estimated results**	**Relative deviation**
Building 1	kWh/m^2	kWh/m^2	%
Heating Load	39,63	33,39	15,75
Cooling Load	32,39	29,23	9,76

Table 59 - Building Validation Test 1; Comparison between the literature [14] and the estimated loads of the more efficient building

	Literature results	**Estimated**	**Relative deviation**
Building 2	kWh/m^2	kWh/m^2	%
Heating Load	21,47	22,04	2,65
Cooling Load	51,22	44,17	13,76

Validation Test 2 – INEGI Building (real data)

The INEGI building was analysed to test if the building characterization routine could be applied to a real building, and represent it. The INEGI building (Figure 52) possesses two main buildings, the Torre – right building - and Corpo – left building. The INEGI Torre has a total of 10 floors (-1 to 8), where the -1 is composed by an auditory and miscellaneous rooms, 0 floor is the reception, 1st floor the restaurant, and the remaining floors 2nd to 8th a mix of open and private offices. The first floor of the building Corpo is a mix of open and private offices. At the time of the audits, INEGI was open Monday to Saturday, from 7h to 22h.

Only the heating needs can be compared with the routine output. The cooling needs are provided by electric chillers, whose consumption is aggregated with working equipment and lighting, being therefore infeasible to analyse. From the audit, the space heating needs are obtained by analysing the natural gas consumption, which is utilized by a hot water boiler, responsible for space heating, and the restaurant cuisine. The 1st floor of Corpo and all the Torre floors, except the -1 and 0 floor, heating needs are provided by the central space heating. Looking at the occupied hours, despite being open from 7h to 22h, the majority of the workers (with varying occupation percentage) work from 8h to 19h. In addition, it is considered that the building is not occupied on the weekends.

To validate the data, the routine was run twice, to characterize both Torre and Corpo, and then the heating needs of both buildings was aggregated. For the Torre, as a simplification, it is accounted that the 1st floor – the restaurant- is an open office like the above floors. The 1st floor, however, is only close to full occupation during lunch hours, being the remaining time rarely occupied. The input data for both Torre and Corpo can be seen in Table 60 and Table 61, respectively. The data common to both buildings is presented in Table 62 .

Figure 52 - Building Validation 2; INEGI building composed by Corpo (left building) and Torre (right building)

Table 60 - Building Validation 2; Torre building characteristics

Input - Torre building		
parameter	**value**	**unit**
width_floor	17	m
length_floor	17	m
height_floor	2.7	m
ratio_N_wall	0.9	-
ratio_S_wall	0.3	-
ratio_E_wall	0.3	-
ratio_W_wall	0.9	-
floor_number	8	-

Table 61 - Building Validation 2; Corpo building characteristics

Input – Corpo building		
parameter	**value**	**unit**
width	40	m
length	17	m
height	2.8	m
ratio_N_wall	0.9	-
ratio_S_wall	0.5	-
ratio_E_wall	1	-
ratio_W_wall	1	-
floor_number	1	-

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

Table 62 - Building Validation 2; Both Torre and Corpo building common characteristics

Input – Corpo building			
parameter		**value**	**unit**
latitude		41.10	º
longitude		-8.3539	º
building_orientation		South	-
daily periods		[7,19]	h
saturday_on		0	-
saturday_on		0	-
shutdown_periods		August month – [213, 243]	-
occupied set temperatures	T_cool_on	25	ºC
	T_heat_on	22	ºC
non-occupied set temperatures	T_off_min	-	ºC
	T_off_max	-	ºC

The climate data used, more specifically, the hourly solar radiation and ambient temperature is obtained for a typical meteorological year (TMY), from *pvlib*. The remaining data, such as thermal properties of the building materials and internal gains due to equipment/artificial lighting, and others, is the default data from the KB.

Before proceeding to the analysis of the validation, it should be highlighted that from the audit it is known the natural gas is consumed by the water boiler, which provides the space heating to the Torre (1st to 8th floor) and the 1st floor of Corpo, and by the canteen cuisine. To remove the canteen cuisine natural gas consumption, the June natural gas consumption was subtracted to all the months. June is typically a hot month, thus not needing space heating, with normal workers occupation. Table 63 presents the audit natural gas consumption with and without the canteen natural gas consumption in June, and the routine natural gas consumption, considering the boiler's efficiency (from the audit). Figure 53 presents the comparison between real and the model's natural gas consumption.

Table 63 - Comparison between Real and estimated natural gas consumption

	Real (2015)		**Theoretical Model**	
	Total Consumption	**Without canteen consumption**	**Considering boiler efficiency = 90%**	**Relative Deviation**
Month	kWh	kWh	kWh	[%]
Jan	16 514	14828,3	13672,22	7,8%
Feb	15 987	14301,3	6965,94	51,3%
Mar	4 689	3003,4	3554,54	18,4%
Apr	2 116	430,8	1544,56	158,6%
May	1 828	142,3	319,26	24,3%
Jun	1 686	0,0	63,57	-
Jul	1 086	0,0	0,00	0,0%
Aug	1 016	0,0	0,00	0,0%
Sep	1 628	0,0	6,49	-
Oct	1 816	130,1	374,01	87,5%
Nov	8 039	6353,7	7348,50	15,7%
Dec	9 606	7920,0	10011,56	26,4%
TOTAL	66011,9	47109,8	43860,66	6,9%

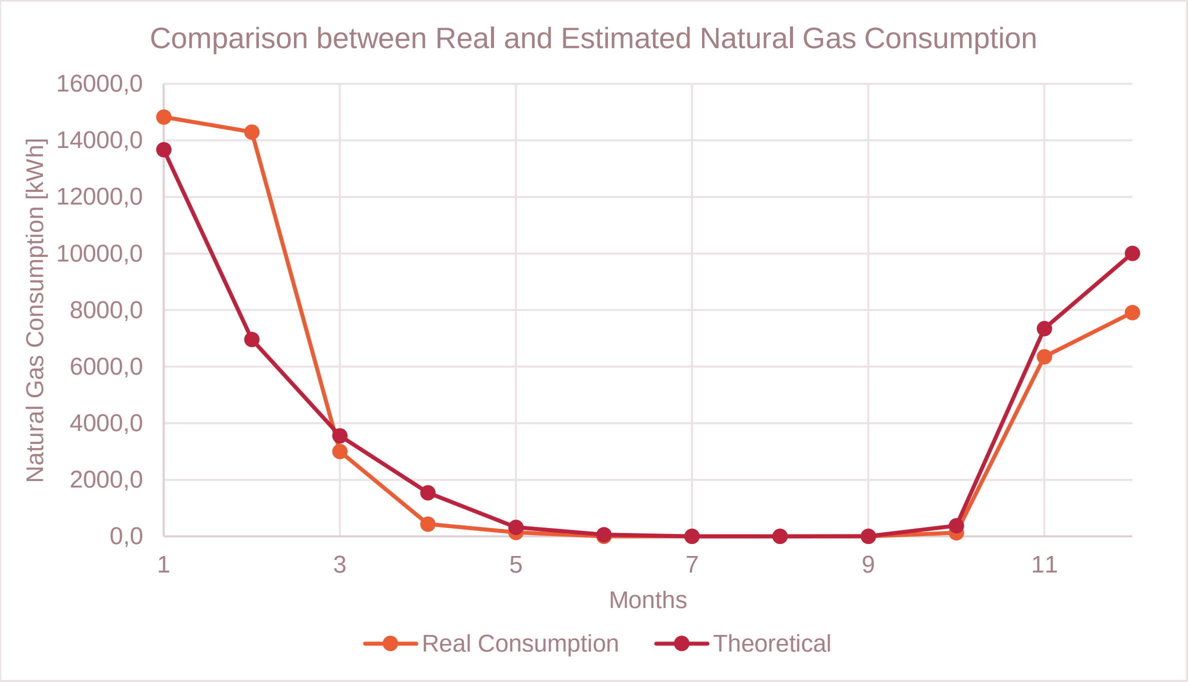


Figure 53 - Comparison between real and estimated natural gas consumption

Based on a small amount of inputs and several default values, it is possible to obtain a reasonable estimate on the natural gas consumption. The difference between real and estimated consumption can be due to different reasons in addition to the routine thermal model simplicity, as explained before:

- In the routine, the floors are all considered to be climatized and open offices, even though some floors have large number of private offices and even non climatized zones.
- The 1st floor where the restaurant is localized, is considered as an open office, even though its occupation and internal heat gains are different from an office.
- The restaurant cuisine natural gas consumption is not known for sure, which can also induce in a slight error.
- The routine considers that the windows' blinds are always open, meaning that the solar gains are always maximum.
- As an approximation, during the schedule occupied periods it is considered full occupation and space heating control.

The largest discrepancy in absolute values is found on February, explained not only by the reasons above but also by the fact that the climate data utilized is the TMY, and not the year 2015, which could have been a colder month than usual.

c. Characterization: Greenhouse (script: greenhouse)

The validation test is done based on experimental data from the literature [17] and [18] , which analyse 3 days of Chinese style greenhouse (see Figure 54) heating, and the second test to validate a rough estimate on the heating technology power required.

Validation Test – Greenhouse (experimental data)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

The first validation data test was done on data from the literature [17] and [18]. The authors analysed a greenhouse heating requirement for 3 days in March, near Winnipeg, Canada. From the literature the hourly ambient and indoor temperature variation inside the greenhouse are obtained - Figure 55 -, as well as the heating requirements -

Figure 56 - Experimental data; Utilized the hourly global horizontal solar radiation

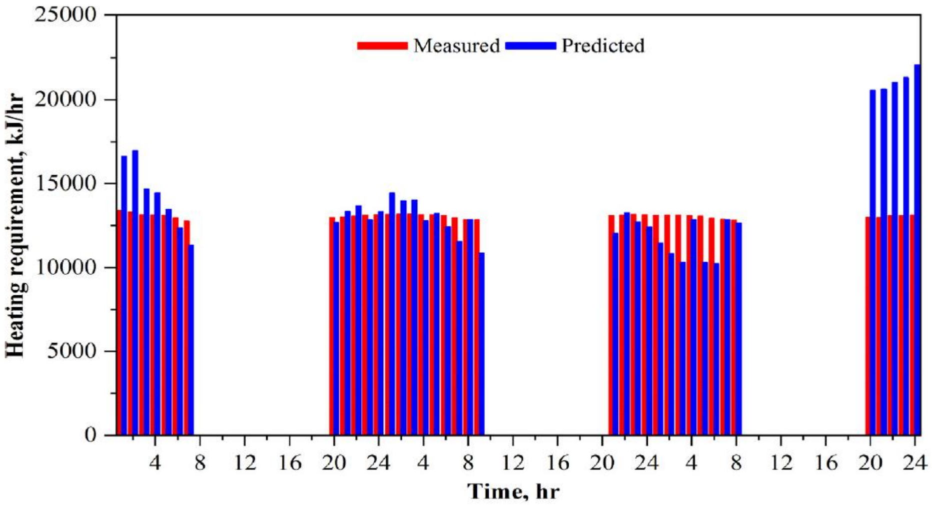


Figure 57. From Figure 56, it is possible to obtain the total incident solar radiation and wind speed. Additional input data for the greenhouse routine can be observed in Table 64 - It is considered that the greenhouse is tight sealed (parameter greenhouse_efficiency=1) since it is a recent built and placed in region with low temperatures.

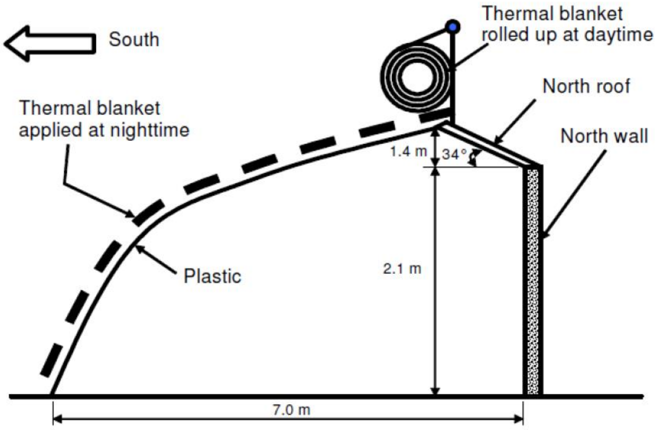


Figure 54 - Chinese greenhouse representation [18]

Moreover, from [18] some specific procedures are known to have occurred during the experiment, that were implemented specifically in the routine:

- when the greenhouse temperature reaches 12ºC, an electric heater, of 3.6kW is automatically turned on, until the indoor temperature reaches 18ºC, then it turns off;

- the thermal blanket is rolled down to cover the south roof before the sunset at around 5:30 p.m and uncover at 7:00 a.m;

- the North wall with thermal inertia was not modelled. It was considered the whole wall to be at the experimental data temperature (given in Figure 55).

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

Table 64 - Greenhouse Validation 1; Input data

Input - Greenhouse		
Parameter	Value	Unit
width	28	m
length	6.7	m
height	1.5	m
greenhouse_efficiency	1	-
thermal_blanket	1	-

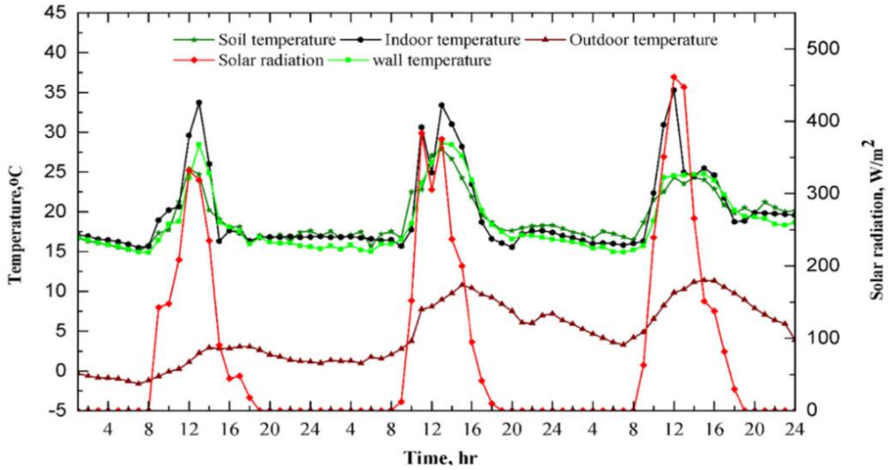


Figure 55 – Experimental data; Utilized the hourly temperatures [17]

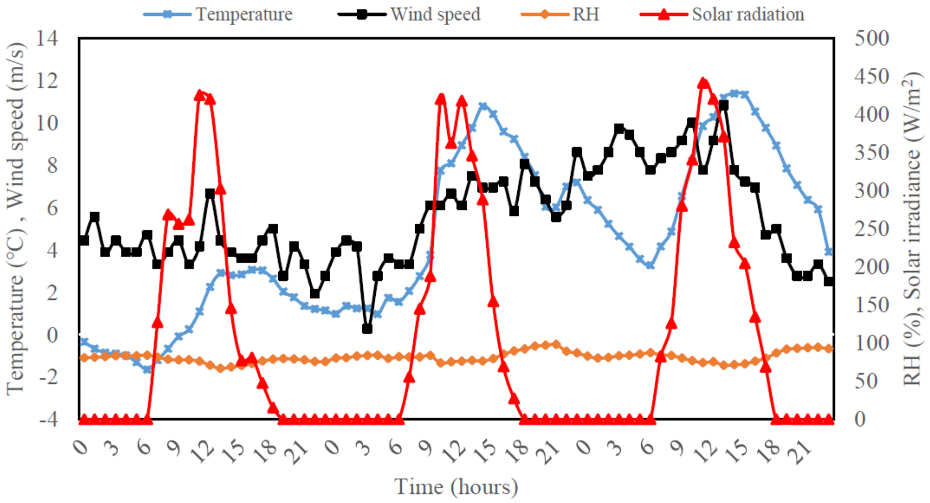


Figure 56 - Experimental data; Utilized the hourly global horizontal solar radiation [18]

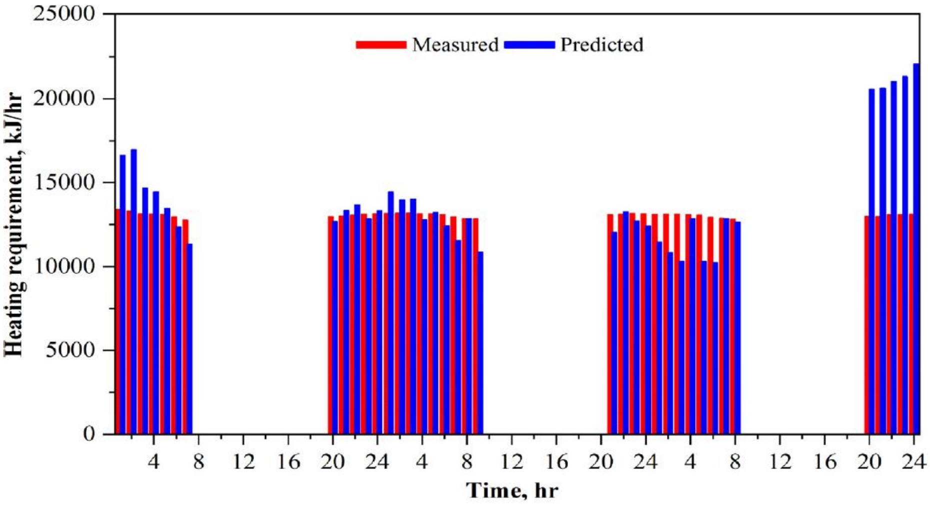


Figure 57 – Experimental data heating requirements [17]

The "*greenhouse*" routine was run twice in different forms to characterize the greenhouse. For the first characterization, the routine was slightly adapted to the experimental data, regarding the heaters' operation. Instead of computing the heating needs as described in section ii, it was introduced a heater that when the greenhouse temperature reaches 12ºC would provide of 3.6kW until the indoor temperature reaches 18ºC, then turning off. The results of the estimated heating needs can be seen in Figure 58, and the greenhouse indoor air temperature variation in Figure 59.

When looking at Figure 58, it can be seen that the model presents an earlier input of heating than the experimental, as the day ends. Adding to the simplicity of the model, it might be explained by the fact that the solar radiation gains are only considered on a horizontal surface – the greenhouse floor. However, the greenhouse's solar gains might actually be superior to the ones considered in the model since the greenhouse's north wall solar gains increase as the sun is lower.

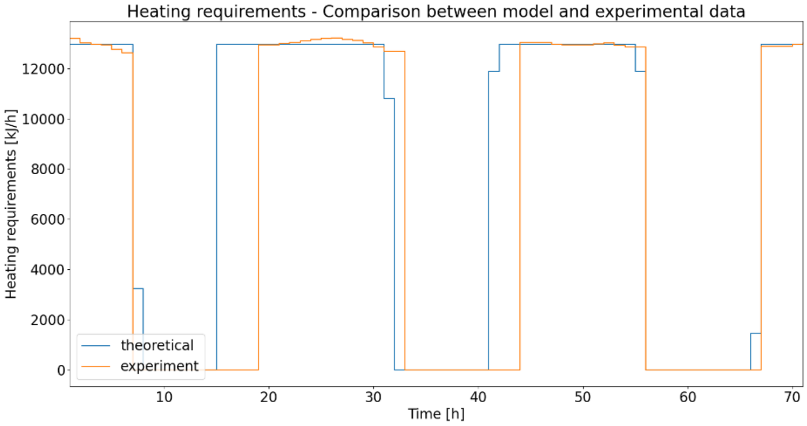


Figure 58 - Comparison between experimental [17] and estimated heating requirements (with heating mode similar to the experiment)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

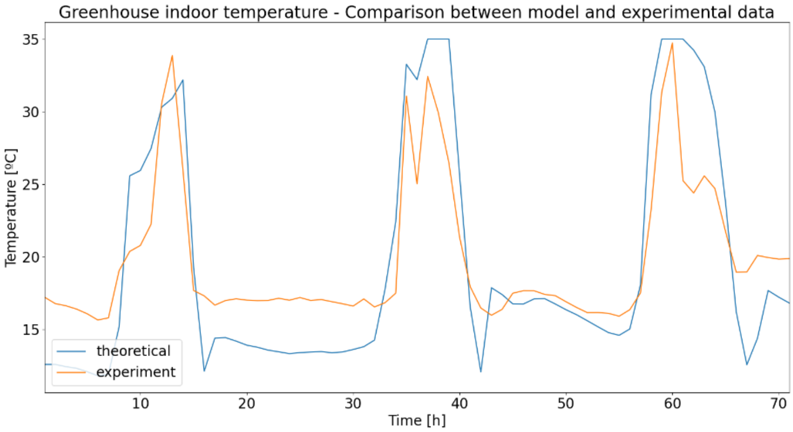


Figure 59 - Comparison between experimental (Md. Shamim Ahamed, 2018) and estimated greenhouse indoor temperature (with heating mode similar to the experiment)

For the second simulation, the routine space heating is as described in section ii. The heating provided is not static throughout time as it can be observed in Figure 60. The space heating does not provide 3.6kW, but the necessary heat to hold the temperature at a minimum of 12ºC (additional input added, as presented in Table 65). It can be confirmed that the routine can follow well the experimental data, and give a good representation of the heating requirements of the greenhouse.

Table 65 - Greenhouse Validation; Additional input data

Input - Greenhouse		
Parameter	Value	Unit
T_heat_on	12	ºC

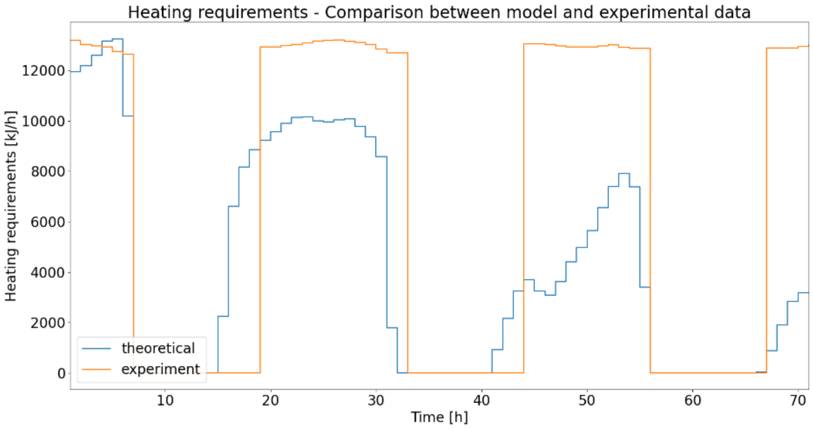


Figure 60 - Comparison between experimental (Md. Shamim Ahamed, 2018) and estimated heating requirements (with heating mode holding the temperature at a minimum of 12ºC)

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

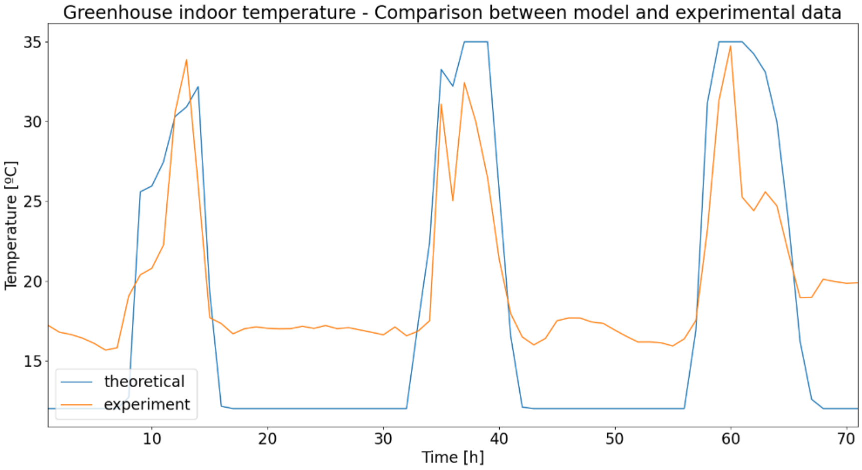


Figure 61 - Comparison between experimental (Md. Shamim Ahamed, 2018) and estimated greenhouse indoor temperature (with heating mode holding the temperature at a minimum of 12ºC)

2.1.6.5.4 Interaction with other modules

The characterization routines are purely based on interaction between the user and the CF module. The CF may access the KB for default values if the user does not provide enough data, as needed.

The routines that interact with the other modules are the source's simulation for the internal heat recovery ('convert_pinch' and 'convert_orc'), and the DHN streams conversion 'convert_sources' and the sink's DHN conversion convert_sinks'.

a. Internal heat recovery

The internal heat recovery routines receive the streams the user desires to analyse whether to design a possible ORC ('convert_orc') or to design a network of heat exchangers by performing a pinch analysis ('convert_pinch'). Both routines send techno-economic data of the best designed solutions to the Business Module (BM), so that a financial analysis can be executed in order to provide the user with a better understanding of the feasibility of implementing such technologies. The data exchanged between CF and BM, can be found in Figure 62.

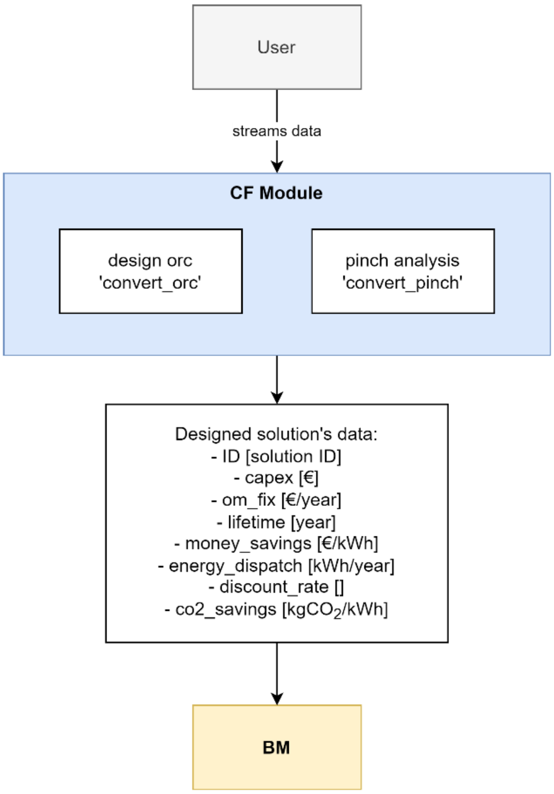


Figure 62 - Interaction between Core Functionalities and Business Module

b. DHN conversion

In the first step, the group of sinks and sources is provided to the CF so that the excess heat available from the sources and the heat needed by the sinks is identified. The CF sends to the GIS the maximum heat available and to be consumed, by respective sources and sinks, so that the GIS designs a feasible DHN and calculates the capital cost and the thermal losses of the DHN. TEO will then determine the least cost matching of sources and sinks considering the energy losses in the network. In the second step, the GIS and TEO will iterate until the exchanged capacities between sinks and sources, and grid losses converge. When the iteration ends, due to optimum design of grid and exchanged capacities between source and sink players, the CF send some technical data for the Market Module.

The data exchanged between CF and the remaining modules for the DHN conversion on both sinks and sources, can be found in Figure 63.

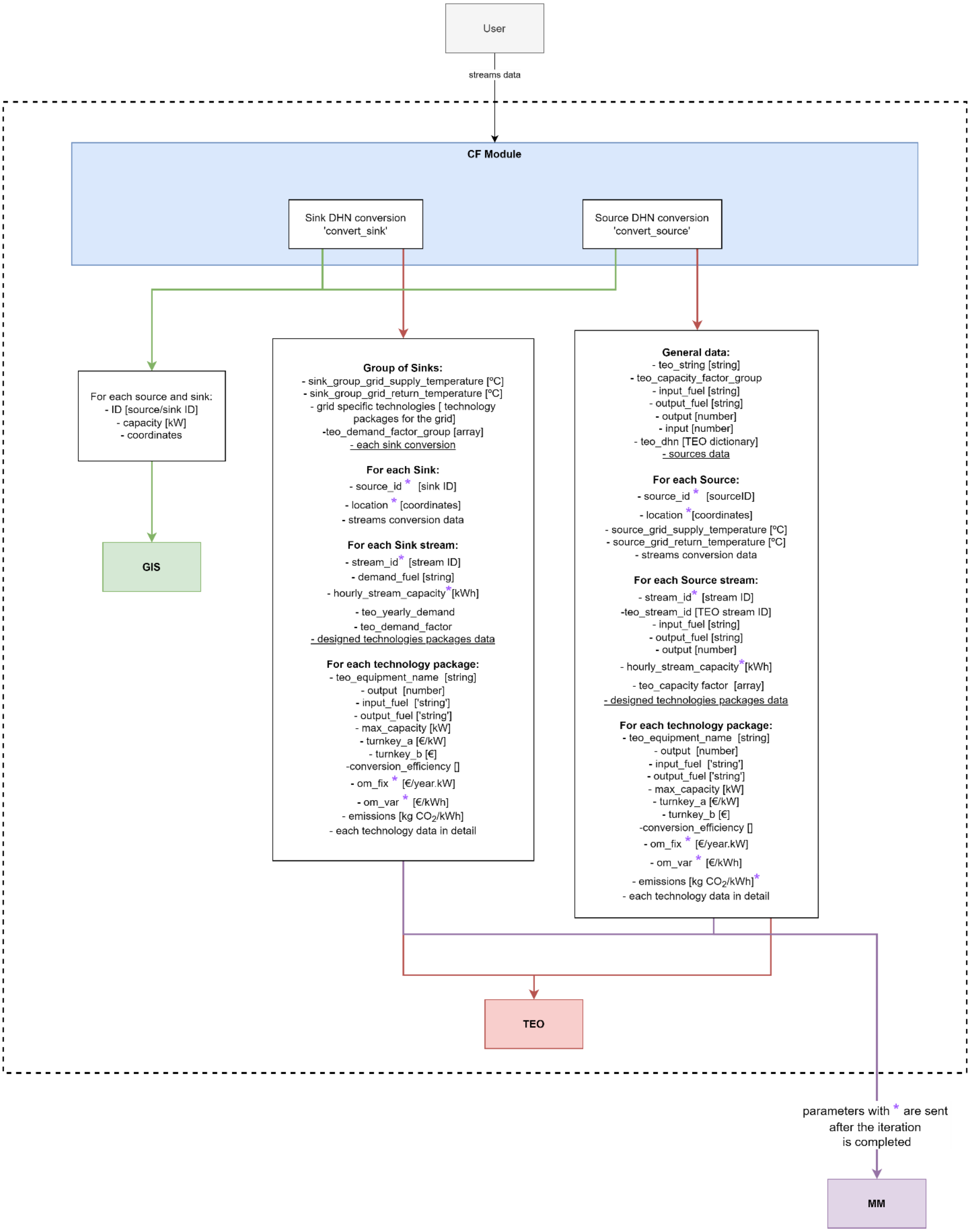


Figure 63 - Interaction between CF and GIS, TEO, and MM

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

2.1.7 Reports

2.1.7.1 Characterization

The CF characterization is deeply linked to the platform interface. After the user characterizes the streams of its sinks/sources, it is displayed all the inserted data, so that the user can confirm the streams information.

1) Users that perform the simple characterization of both sources and sinks, should visualize the full data of the streams, as presented in Table 66, prior to any simulation.

Table 66 - Report User; Simple Characterization

Stream Data		
var name	**description**	**units**
id	stream ID	-
fluid	fluid name	-
stream_type	if an ‘inflow’,’outflow’,’supply_heat’,’excess_heat’ stream	-
saturday_on	if working -1- or not -0- on Saturday	-
sunday_on	if working -1- or not -0- on Sunday	-
shutdown_periods	array with shutdown periods	days
daily_periods	array with daily periods	hours
capacity	stream capacity	kW
supply_temperature	stream supply/initial temperature	ºC
target_temperature	stream target/final temperature	ºC
flowrate	stream flowrate	kg/h

2) Users that perform the detailed characterization of sources, should visualize the full data of the streams, as presented in Table 66, associated to the respective equipment and process.

3) Users that perform the detailed characterization of sinks, should visualize the monthly heating and cooling demands generated by the “building” or” greenhouse” routines. These can be corrected by the user, which will update the correspondent hourly heating/cooling capacities on the stream’s characterization.

2.1.7.2 Simulation

The majority of CF simulation output data goes to the other modules for additional computations and final designs. The routines which present data to the user are the source simulations, in specific, ‘convert_orc’ and ‘convert_pinch’.

1) The routine ‘convert_orc’ will design the ORC according to the streams given (see ii) and provide the user with technical and economical parameters presented in

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

Table 67 - Report User; ORC design

ORC data		
var name	**description**	**units**
streams_id	array with the converted streams ID	-
electrical_generation_nominal	nominal electrical supply capacity	[kW]
electrical_generation_yearly	array with hourly electrical supply capacity	[kWh]
excess_heat_supply_capacity	excess heat available supply capacity	[kW]
conversion_efficiency	conversion efficiency of heat to electricity	-
turnkey	equipment turnkey	[€]
om_fix	equipment turnkey O&M fix	[€/year]
om_var	equipment turnkey O&M variable	[€]

2) The routine 'convert_ pinch' designs the heat exchanger network according to the streams given (see 0) and, not only provide the user with general data about the HX network, but also technical and economical parameters of the HX designed, and the emissions/energy/cost savings for the existing equipment

Table 68 - Report User; Pinch analysis, heat exchangers network general data

Heat Exchanger Network general data		
var name	**description**	**units**
ID	designed solution ID	-
streams	streams in pinch design	-
capex	design turnkey	€
om_fix	yearly O&M fixed costs	€/kW
hot_utility	power of the hot utility needed	kW
cold_utility	power of the cold utility needed	kW
discount_rate	discount rate to be applied on the business analysis	-
pinch_temperature	design pinch temperature	ºC
theo_minimum_hot_utility	theoretical power of the hot utility needed	kW
theo_minimum_cold_utility	theoretical power of the cold utility needed	kW
equipment_detailed_savings	array with dictionaries of each equipment savings when implementing the pinch design	-
pinch_hx_data	array with dictionaries of designed heat exchangers	-

Table 69 -- Report User; Pinch analysis, heat exchangers techno-economic data

Heat Exchanger techno-economic data		
var name	**description**	**units**
HX_Power	heat exchanger design power	kW
HX_Hot_Stream	hot stream ID	-
HX_Cold_Stream	cold stream ID	-
HX_Original_Hot_Stream	original hot stream ID (it can be different of HX_Hot_Stream if a stream split occurs - a new id is given to the split)	-

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

HX_Original_Cold_Stream	original cold stream ID (it can be different of HX_Cold_Stream if a stream split occurs - a new id is given to the split)	-
HX_Cold_Stream_flowrate	mass flowrate	kg/h
HX_Hot_Stream_flowrate	mass flowrate	kg/h
HX_Type	type of heat exchanger	-
HX_Turnkey_Cost	heat exchanger turnkey cost	€
HX_OM_Fix_Cost	heat exchanger O&M cost	€/kW
HX_Hot_Stream_T_Hot	hot stream hot temperature	ºC
HX_Hot_Stream_T_Cold	hot stream cold temperature	ºC
HX_Cold_Stream_T_Hot	cold stream hot temperature	ºC
HX_Cold_Stream_T_Cold	cold stream cold temperature	ºC
Storage	storage volume	m^3
Storage_Turnkey_Cost	storage turnkey cost	€
Total_Turnkey_Cost	whole package (heat exchanger + storage) turnkey cost	€
Recovered_Energy	amount of yearly energy recovered	kWh

Table 70 - Report User; Pinch analysis, equipment detailed savings

Heat exchanger network – Equipment detailed savings		
var name	**description**	**units**
Equipment_ID	equipmet ID	-
CO2_Savings_Year	yearly CO_2 emissions saved	kg CO_2
Recovered_Energy	yearly energy saved	kWh
Savings_Year	yearly monetary savings	€

Moreover, the routine also reproduces graphical representations of the HX networks designed, which can be visualized by the user (Figure 64).

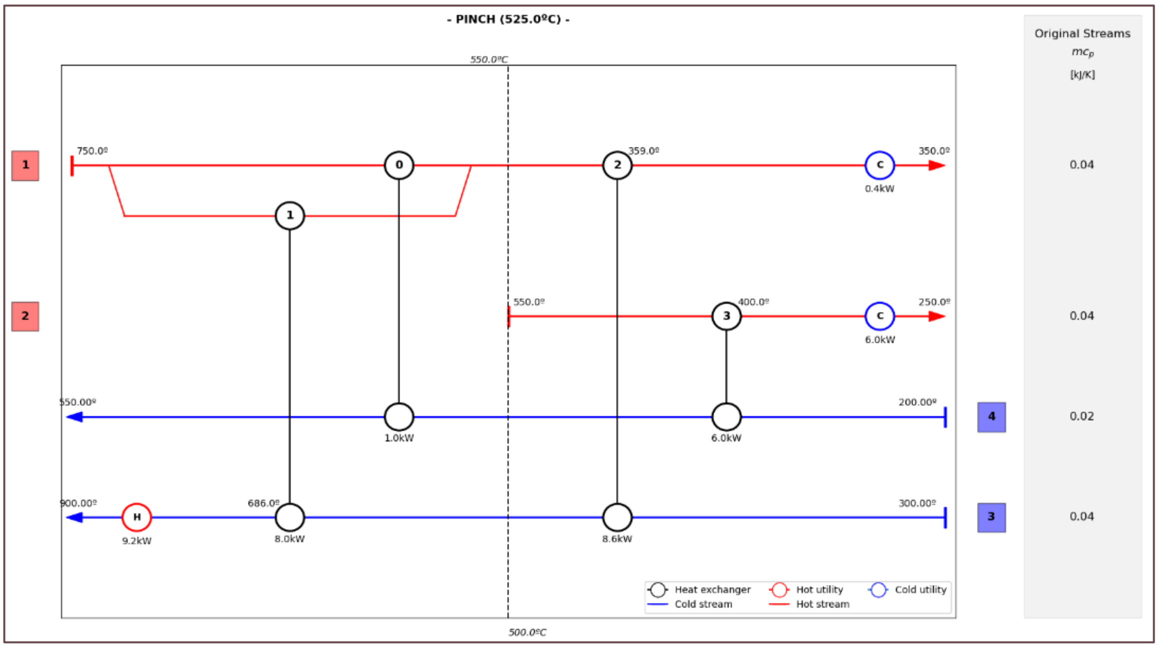


Figure 64 - Graphical representation of heat exchanger network

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

2.2 User Manual

2.2.1 Introduction to the CF

The purpose of the Core Functionalities (CF) module is to allow for the full characterization of the EMB3Rs platform objects (sinks and sources) and to provide information to all the remaining analysis modules, namely the geographical information systems (GIS) module, the techno-economic (TEO) module, the market module (MM), and the business module (BM); to run their simulations.

The CF module divides both sinks and sources methods into two main sections: characterization and simulation. The characterization routines focus on receiving the user inputs and performing the needed computations to characterize the created objects, e.g., when the user creates a sink object, namely a greenhouse, the CF will compute its hourly heating needs according to its location, dimensions, and other input parameters. The simulation routines focus on performing analysis based on the characterization information, e.g. for a source's excess heat streams (which were computed in the characterization), the conversion simulation will evaluate the available amount of energy that can be provided to a district heating network (DHN).

2.2.2 Main Features of the CF Module

The CF module main features according to the object type are:

- **Source:**
    - **Simple characterization: excess heat streams characterization (characterization)**
    - **Detailed characterization: Industry's equipment, processes, and streams characterization (characterization)**
    - **Internal heat recovery analysis (simulation)**
    - **Conversion of the source's excess heat streams to the DHN and evaluation of the technologies to be implemented (simulation)**
- **Sink:**
    - **Simple characterization: main circuits (hot water, steam and chilled water) characterization**
    - **Detailed characterization: Industry and buildings – greenhouse, hotel, residential, office - heating/cooling demand and streams characterization (characterization)**
    - **Conversion of the DHN to the sink needs and evaluation of the technologies to be implemented (simulation)**

Looking in more detail at the main platform objects.
When a user creates a **source** (see Figure 65), there are two methods to perform its characterization. A **simple** method if the user desires to characterize directly specific excess heat streams and a more **detailed** method for users who intend an industry complete characterization. The last method requires the users to introduce in detail their equipment and processes data. In terms of simulation, whether simplified or detailed characterization, the CF module will **convert** the source´s excess heat to the

DHN, estimating the available conversion heat and the technologies that could be implemented. Moreover, it is also evaluated the conversion of heat to electricity, by performing the Organic Rankine Cycle (ORC) conversion - **internal heat recovery**. Only for the users who performed the detailed characterization, it is performed the pinch analysis - internal heat recovery -, in which the CF suggests possible heat exchanger designs so that it can be recovered heat within processes.

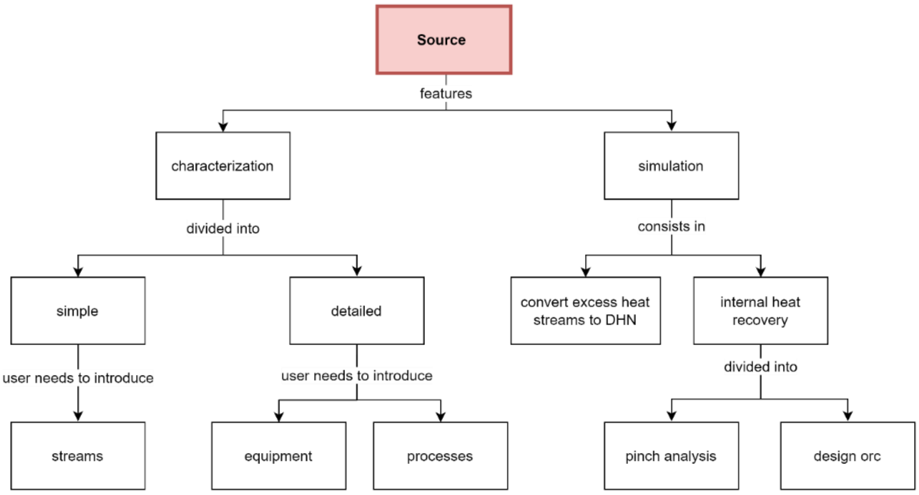


Figure 65 - Overview of the source architecture

When a user creates a **sink** (see Figure 66), it is prompted to the user to characterize its heating/cooling demand. Similar to the source, there is a **simplified** form for the user to input directly a specific heat/cold stream demand, and a more **detailed** form for the users who which to characterize buildings – residential, offices, hotels, and greenhouses. According to the user's buildings specification, the CF will characterize the building by generating the heating/cooling demand. Simulation-wise, the CF will evaluate the technologies that could be implemented on the DHN to meet the heat/cold sink´s needs.

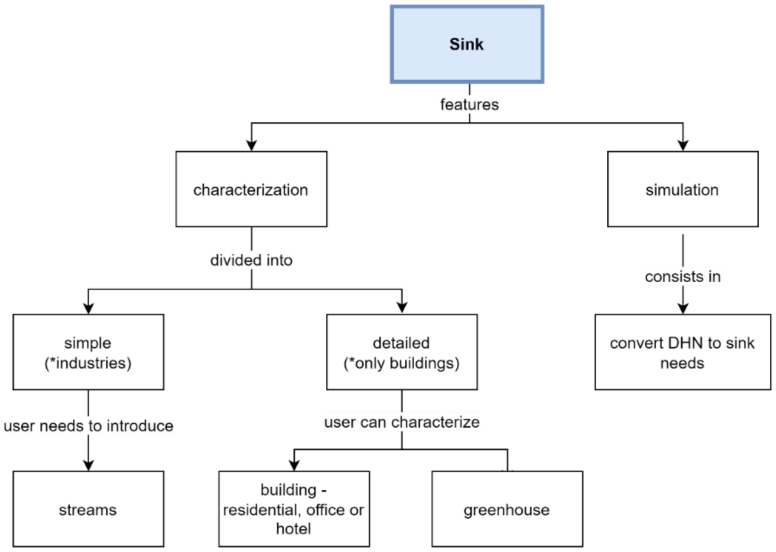


Figure 66 - Overview of the sink architecture

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

2.2.3 Objects Characterization

2.2.3.1 Description

The characterization routines are responsible for receiving the input data from the user, and, by performing several computations, assess the streams and its parameters, such as flowrate, temperatures and capacity. Figure 67 presents an overview on the characterization flow of both sinks and sources. The following sections assess in detail the characterization of each object type.

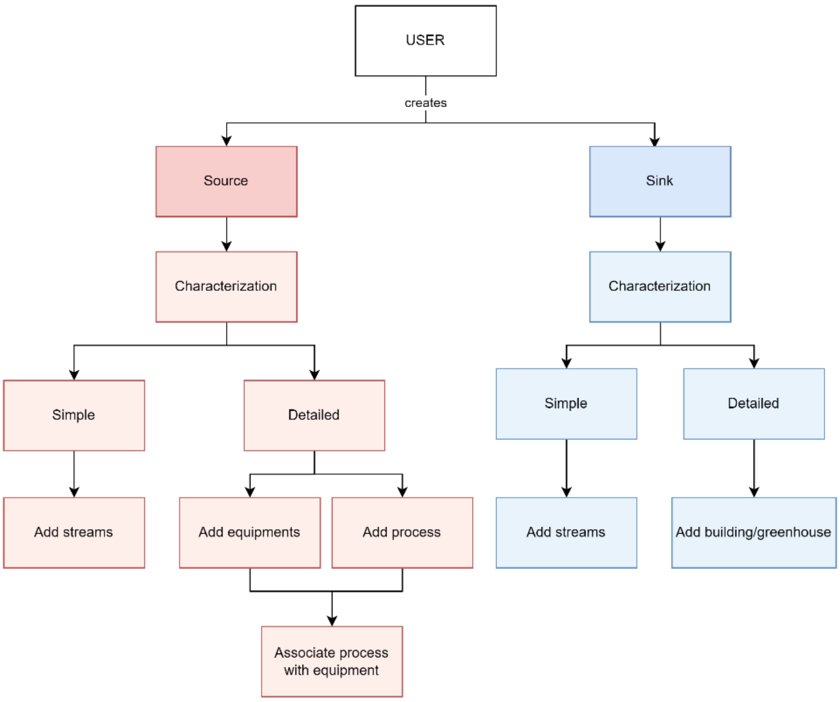


Figure 67 - Characterization graphical scheme

2.2.3.2 Source

The characterization is divided into **simple** - the user just needs to characterize its excess heat streams - and **detailed** - the user must describe all its equipment and processes in detail so that the algorithm can characterize the streams. Source objects characterized in both detailed and simplified methods can, later on, perform the excess heat streams conversion to the DHN or ORC design simulations. Only users who choose to carry out a detailed characterization can perform the pinch analysis simulation, as presented in Figure 10. Detailed information regarding the simulations is presented at the end of this document.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

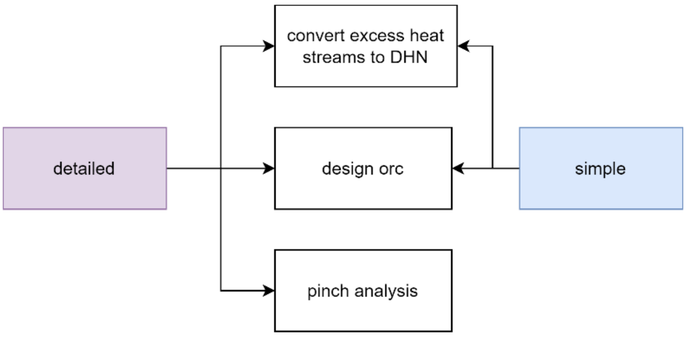


Figure 68 – Source characterization methods and respective simulation options

For the **characterization**:

1. The user enters the platform and chooses to create a source
2. The user sets the location and source name
    a. The user chooses to perform **simple characterization**
        i. User introduces streams and its properties.
        ii. Stream characteristics are estimated by the module
    b. The user chooses to perform **detailed characterization**
        i. User introduces processes and characterizes its streams
        ii. User introduces equipment and characterizes its streams.
            a. Equipment streams estimated by the module
3. User associates processes with respective equipment
4. Data is displayed on the platform so that the user is able to validate and/or modify objects data
5. Characterization data is saved

2.2.3.2.1 Simple characterization

Table 71 presents the variables, and respective description, the user needs to provide to perform the simple characterization. Table 72 presents an example on a couple of excess heat streams to be analyzed.

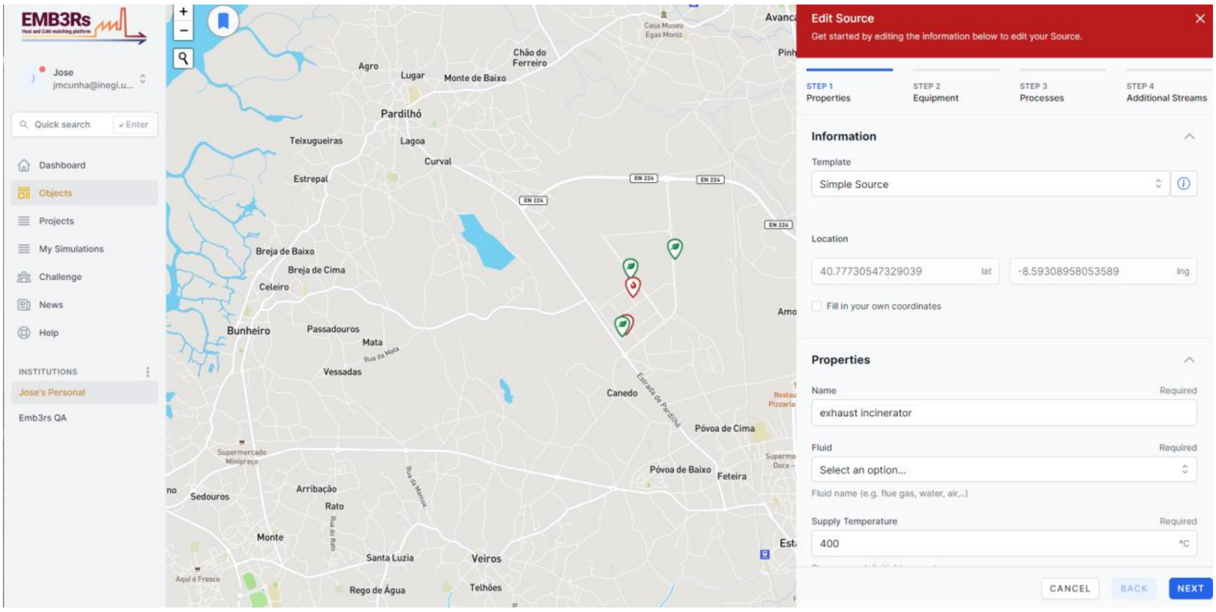
Table 71 - Simple characterization input data

Input data		
var name	**description**	**units**
supply_temperature	stream supply/initial temperature	ºC
target_temperature	stream target/final temperature	ºC
fluid	fluid name	-
fluid_cp	specific heat capacity	kJ/kg.K
flowrate	mass flowrate	kg/h
saturday_on	if working -1- or not -0- on Saturday	-
sunday_on	if working -1- or not -0- on Sunday	-
shutdown_periods	array with shutdown periods	days
daily_periods	array with daily periods	hours

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

*Table 72 - Example Simple Characterization User inputs. Parameters with * are provided by the KB, however, the user can also add new data*

Stream	Fluid*	Cp*	Supply T.	Target T.	Flowrate	Daily Periods	Saturday ON	Sunday ON	Shutdown Periods
units	-	kJ/kg.K	ºC	ºC	kg/h	h	-	-	day/month
1	flue gas	1.4	450	125	9000	5-15; 19-24	1	1	1/8 – 31/8
2	water	4.2	90	50	1000	0-24	1	1	1/7 – 31/8



Get started by editing the information below to edit your Source.

Information

Template: Simple Source

Location: 40.77730547329039 lat; -8.59308958053589 lng

☐ Fill in your own coordinates

Properties

Name (Required): exhaust incinerator

Fluid (Required): Select an option...

Fluid name (e.g. flue gas, water, air,..)

Supply Temperature (Required): 400 °C

Figure 69 - Printscreen of the platform input template for sources

2.2.3.2.2 Detailed characterization[2]

a. Process

The user can add multiple process objects for the same industry. The user must always provide the process schedule, the schedule type (e.g. batch or continuous), the operation temperature, and characterize the four types of streams possible: startup, maintenance, inflow, and outflow. There can be multiple streams of each stream type. For the startup, the user has to characterize the fluid, initial temperature, and mass. For both inflow and outflow the flow rate and the fluid, and also the supply and target temperature are needed, respectively. For the maintenance stream, the capacity needed to maintain a process at a certain temperature is requested.

Table 73 to Table 77 present the variables, and respective description, the user needs to provide to perform the processes' characterization.

---

[2] The source detailed characterization has been implemented in past iterations of the platform and kept in the manual for reference, although it is not used. All of the detailed characterization takes place in the background, and is not visible to the user. Along the development process, it was verified that users know their industrial processes very well, and it wouldn't be needed to add such complexity to the simulation. As of M38, the user can add simple sources with multiple streams for waste heat recovery, which addresses the needs identified so far.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

Table 73 – Processes input data

Process input data		
var name	**description**	**units**
id	process ID	-
equipment	heat/cooling equipment ID the process is associated to	-
operation_temperature	process operation temperature	ºC
saturday_on	if working -1- or not -0- on Saturday	-
sunday_on	if working -1- or not -0- on Sunday -	-
shutdown_periods	array with shutdown periods	days
daily_periods	array with daily periods	hours
schedule_type	0=continuous, 1=batch	-
cycle_time_percentage	batch production time ratio for the startup	-
startup_data	array with dictionaries with startup streams characteristics	-
maintenance_data	array with dictionaries with maintenance streams characteristics	-
inflow_data	array with dictionaries with inflow streams characteristics	-
outflow_data	array with dictionaries with outflow streams characteristics	-

Table 74 – Startup stream additional needed inputs

Startup stream input data		
var name	**description**	**units**
fluid	fluid name	-
initial_temperature	medium initial temperature	[ºC]
fluid_cp	fluid's cp	[kJ/kg.K]
mass	medium mass	[kg]

Table 75 - Maintenance stream additional needed inputs

Maintenance stream input data		
var name	**description**	**units**
capacity	capacity given to a process to compensate for the thermal losses	kW

Table 76 - Inflow stream additional needed inputs

Inflow stream input data		
var name	**description**	**units**
fluid	fluid name	-
supply_temperature	stream supply temperature	ºC
flowrate	stream flowrate	kg/h

Table 77 - Outflow stream additional needed inputs

Outflow stream input data		
var name	**description**	**units**
fluid	fluid name	-
target_temperature	stream target temperature	ºC
flowrate	stream flowrate	kg/h

An example of the data the user needs to provide to characterize the process and streams is given in Table 78 and Table 79, respectively.

Table 78 - Example Process Characterization

Process	**Process Name**	**Operation temperature**	**Daily Periods**	**Saturday ON**	**Sunday ON**	**Shutdown Periods**	**Schedule Type**	**Cycle Time Percentage**
units	-	-	h	-	-	day/month	-	%
1	Fermentation	85	5-15; 19-24	1	1	1/8 – 31/8	continuous	-
2	Cleaning	95	5-15; 19-24	1	1	1/8 – 31/8	batch	10

Table 79 - Example Process's Streams Characterization

Stream	**Process**	**Stream type**	**Fluid**	**Cp**	**Supply T.**	**Target T.**	**Flowrate**	**Mass**	**Capacity**
units	-	-	-	kJ/kg.K	ºC	ºC	kg/h	kg	kW
1	1	inflow	milk	3.93	20	85	9000		
2	1	outflow	hot whey	3.93	85	35	9000		
3	1	startup	water	4.2	20	85		15	
4	1	maintenance							90

b. Equipment

The user can add heating or cooling equipment. The equipment created is saved in the Knowledge Base (KB) with the attributes introduced by the user and estimated by the routines, e.g. equipment efficiency, supply capacity, among others. The heating equipment is characterized by having supply, excess heat, and inflow streams. The cooling equipment with supply and excess heat streams.

i. Heating Equipment

The following heating equipment can be added: boilers, burners and Combined Heat and Power – CHP – units. All the input data described below is required, so that the equipment streams can be characterized. The streams are namely: inflow stream, e.g. a water boiler requires inflow air for the combustion; excess heat stream, e.g. the flue gas due to combustion.

The mandatory inputs common to all equipment are described in Table 11.

Table 80 – General equipment inputs

Equipment General Data		
var name	**description**	**units**
id	equipment ID	-
equipment_sub_type	equipment name	-
supply_temperature	working fluid supply temperature	ºC
return_temperature	working fluid return temperature	ºC
saturday_on	if working -1- or not -0- on Saturday	-
sunday_on	if working -1- or not -0- on Sunday	-
shutdown_periods	array with shutdown periods	days
daily_periods	array with daily periods	hours
fuel_type	equipment fuel (‘natural_gas’;’biomass’;’fuel_oil’;’electricity’)	-

Table 12 was set apart from Table 11 to point out that the user can introduce directly the supply of the equipment capacity, if this is known, or get an indirect estimation of the supply capacity provided by the tool. For the indirect estimation, the user must associate to the equipment, the processes providing heat. When processes are associated to an equipment, yearly heat capacity needed for all process streams is summed and divided by the equipment working hours, to obtain an average equipment’s supply capacity.

Table 81 - General equipment inputs – provide just one

Equipment General – Additional Data		
var name	**description**	**units**
supply_capacity	equipment supply capacity	kW
processes	array with processes streams which are provided by the equipment	-

Table 13 presents the additional boiler inputs. The user should define if the boiler works on open or closed loop. Moreover, the user can designate the boiler’s efficiency, otherwise this value is obtained from the KB.

Table 82 – Boiler additional needed inputs

Boiler - Additional input data		
var name	**description**	**units**
open_closed_loop	if equipment works -1-, or not -0- on open loop/working fluid recirculation	-
global_conversion_efficiency	equipment efficiency	-

The CHP can be characterized either by its thermal supply capacity or by its electrical generation, and respective conversion efficiencies, as presented in Table 14.

Table 83 – CHP additional needed inputs

CHP - Additional input data		
var name	**description**	**units**
thermal_conversion_efficiency and supply_capacity	thermal_conversion_efficiency thermal supply capacity	- kW

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

electrical_conversion_efficiency and electrical_generation	electrical_conversion_efficiency supply electrical capacity	- kWe

The burners are commonly associated to ovens, furnaces and drying processes. It is not feasible to theorethically compute the excess heat to recover of such equipment. Therefore, for the burner, the user must provide the excess heat streams data, such as, supply and target temperatures, and flowrate to properly characterize the stream.

Table 84 – CHP additional needed inputs

Burner - Additional input data		
var name	**description**	**units**
burner_excess_heat_supply_temperature	Excess heat stream supply temperature	ºC
burner_excess_heat_target_temperature	Excess heat stream target temperature	ºC
burner_excess_heat_flowrate	Excess heat stream mass flowrate	kJ/kg.K

Looking at each equipment individually, it is presented in the following tables, an example of the data the user could provide. Only the calendar data varaiables are not provided in the tables (‘daily_periods’, ‘shutdown_periods’, ‘saturday_on’ and ‘sunday_on’)

Boiler

Table 85 – Example Boiler creation inputs

Equipment	**Equipment Sub Type**	**Fuel Type**	**Efficiency**	**Loop type**	**Supply T.**	**Return T.**
units	-				ºC	ºC
1	hot water boiler	Natural gas	0.9	1	90	45

CHP

Table 86 – Example CHP creation inputs

Equipment	**Equipment Sub Type**	**Fuel Type**	**Supply Capacity (Thermal)**	**Thermal Efficiency**	**Electrical Efficiency**	**Electrical Generation**	**Supply T.**	**Return T.**
units	-		kW		-	kWe	ºC	ºC
1	chp	Fuel oil			0.419	3325	450	125

Burner

Table 87 – Example Burner creation inputs

Equipment	**Equipment Sub Type**	**Fuel Type**	**Efficiency**	**E-H. Supply Temperature**	**E-H. Target Temperature**	**E-H. flowrate**

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

units	-			ºC	ºC	kg/h
1	burner	Natural gas	0.6	160	120	200

ii. Cooling Equipment

When adding a cooling equipment, the user must choose the equipment type (e.g. 'co2_chiller', 'cooling_tower', compression_chiller'), and characterize the equipment parameters, such as thermal supply capacity, global conversion efficiency (in this case, the COP - coefficient of performance), and working schedule. The inputs are similar to the ones presented in Table 11 and Table 12. An example of the data the user could provide is presented in Table 88.

Table 88 - Example Cooling equipment creation inputs

Equipment	Equipment Sub Type	Supply Capacity	Efficiency (COP)	Supply T.	Return T.
units	-	kW	-	ºC	ºC
1	compression chiller	400	2	3	12

2.2.3.3 Sink

Similarly to the Source, the characterization is divided into **simple** and **detailed**. For the **simple** characterization the industry user needs to characterize a hot water, steam and/or chilled water streams. A user that performs the **detailed** characterization can analyze a building heat and cooling needs by introducing the building characteristics. Sink objects characterized in both detail or simplified forms, can later on be handled to perform the simulation.

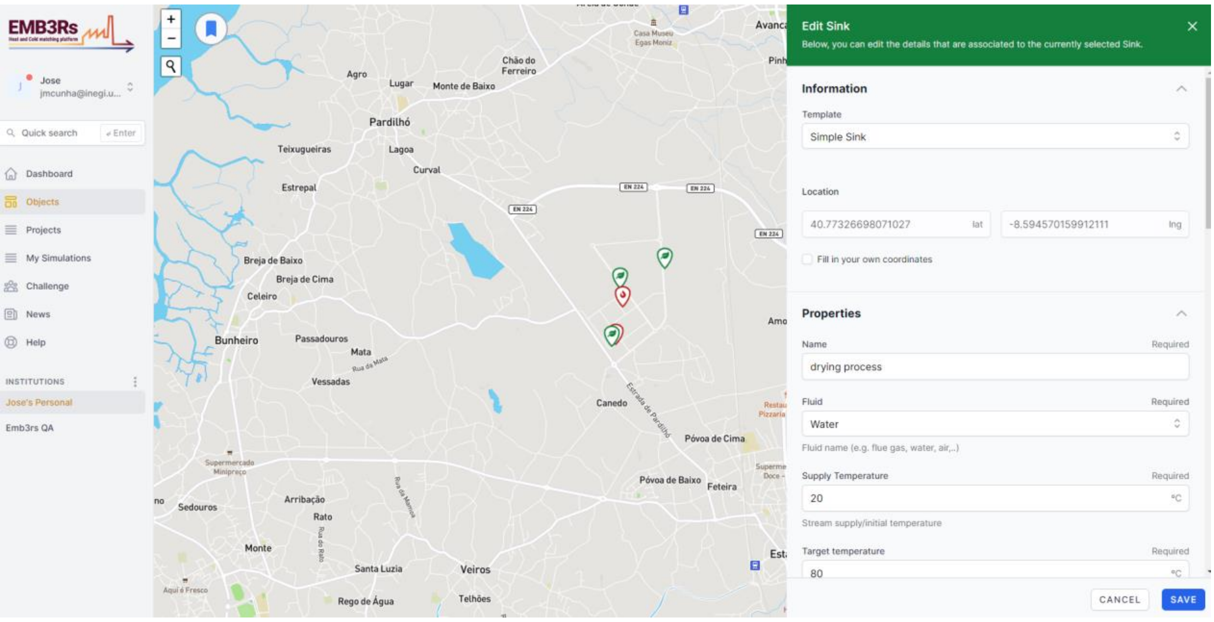
1. The user enters the platform and chooses to create a source or a sink
2. The user sets the location and sink name
    a. The user chooses to perform **simple characterization**
        i. User introduces streams and its properties. Predefined fluids properties are loaded from the KB.
        ii. Stream properties estimated by the module
    b. The user chooses to perform **detailled characterization**
        i. User introduces building/greenhouse properties. Predefined buildings properties are loaded from the KB.
        ii. Stream properties are estimated by the module
3. Data is displayed to the user and he/she is able to validate and/or modify objects data
4. Characterization data is saved

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

2.2.3.3.1 Simple characterization

An industry sink should provide the hot water, steam and chilled water parameters information and provide reference values of the technologies it possesses (if existent). Industries, usually, possess combinations – 1, 2 or all of them - of these 3 types working fluids that supply to the different processes. If the user posseses for example two hot water boilers, the stream's parameters provided should be according to the largest temperature hot water boiler. The same should be applied to the steam and cold water streams (smallest temperature in this case).

Similarly to the sources, the data the user needs to provide to characterize the streams to be analyzed can be found in Table 72. Sinks, which are buildings, must be characterized in the detailed characterization.



Below, you can edit the details that are associated to the currently selected Sink.

Information

Template: Simple Sink

Location: 40.77326698071027 lat, -8.594570159912111 lng

☐ Fill in your own coordinates

Properties

Name (Required): drying process

Fluid (Required): Water

Fluid name (e.g. flue gas, water, air,..)

Supply Temperature (Required): 20 °C

Stream supply/initial temperature

Target temperature (Required): 80 °C

Figure 70- Printscreen of the templates used to add simple sinks to the platform

2.2.3.3.2 Detailed characterization

The building and greenhouse routines are for users that intend to simulate a climate dependent heating/cooling demand. The functions will generate a quick estimate on the hourly heating/cooling demand profile for a full year based on climate data and buildings' indoor temperature requirements. The current implementation can generate demand for three types of buildings: residential, offices and hotels, as well as greenhouses. For both characterization routines, if the ouput does not the user can confirm if the estimated monthly values are close to the building's consumption and correct them if the values are not accurate, by introducing the monthly consumption values. It is applied a coefficient for the hourly profiles according to the estimated monthly consumptions and the values given by the user. Validation test cases and examples for buildings and greenhouse e characterization are described in the System Manual.

a. Building – Residential, Hotel and Office

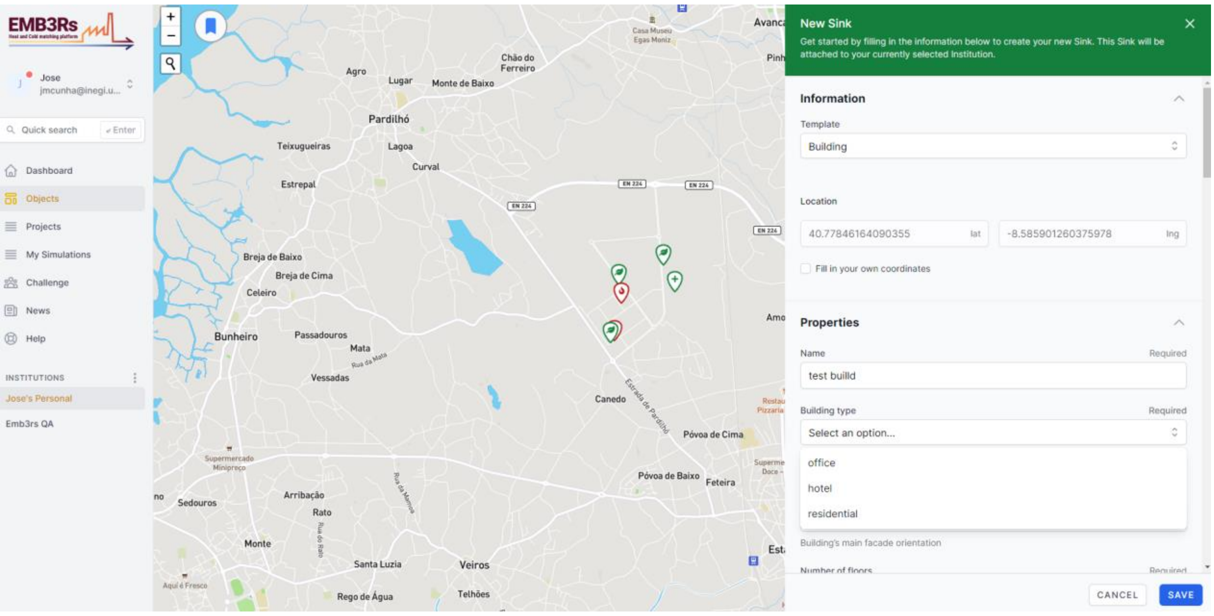
This routine's objective is to estimate a building's heating and cooling consumptions over the year with the input parameters the user provides. These inputs, which are presented in Table 89, are divided into mandatory and optional inputs. The user must introduce the mandatory inputs in order to run the simulation. These are simple parameters that any type of user (from basic to advanced) understands and can introduce to run the simulation. The optional values are advanced building parameters, which have default values so that the simulation runs, and should only be modified by advanced users. The user that desires a more accurate building characterization can modify these default optional valuesTable 43 - Building Mandatory/Optional inputs.

Table 89 - Building Mandatory/Optional inputs

Mandatory Input		
var name	**description**	**units**
latitude	building latitude	º
longitude	building longitude	º
number_floor	number of floors	-
width_floor	floor's width	m
length_floor	floor's length	m
height_floor	floor's height	m
ratio_wall_N	percentage of north wall area in total north facade area (wall and glass)	-
ratio_wall_S	percentage of south wall area in total south facade area (wall and glass)	-
ratio_wall_E	percentage of east wall area in total east facade area (wall and glass)	-
ratio_wall_W	percentage of west wall area in total west facade area (wall and glass)	-
saturday_on	if working -1- or not -0- on Saturday	-
sunday_on	if working -1- or not -0- on Sunday	-
shutdown_periods	periods of days the building is not occupied	days
daily_periods	daily hourly period sbuilding is occupied	hour
building_type	type of building	-
'residential' -> mandatory input -> number_person_per_floor	number of persons per floor; mandatory input for residential buildings	-
building_type = 'hotel' -> mandatory input -> number_rooms	number of rooms per floor; mandatory input for hotel buildings	-
space_heating_type	Space heating type: 1) 0 = Conventional; heaters working fluid supply temperature of 75ºC, heaters working fluid return temperature of 45ºC) 2) 1 = Low temperature; heaters working fluid supply temperature of 50 ºC, heaters working fluid return temperature of 30ºC)	-
building_orientation	building's main facade orientation	-
ref_system_fuel_type_heating	Fuel type associated; e.g. "natural_gas","electricity","biomass","fuel_oil","none"	-
ref_system_fuel_type_cooling	Fuel type associated; e.g. "natural_gas","electricity","biomass","fuel_oil","none"	-
Optional input		
number_person_per_floor	number of persons per floor	-
supply_temperature_heat	heaters working fluid supply temperature	ºC
target_temperature_heat	heaters working fluid target temperature	ºC
supply_temperature_cool	cooling working fluid supply temperature;	ºC
target_temperature_cool	cooling working fluid target temperature	ºC

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

T_cool_on	maximum temperature in a room during occupied hours; cooling is turned on.	ºC
T_heat_on	minimum temperature in a room during occupied hours; heating is turned on.	ºC
T_off_min	minimum temperature in a room during unoccupied hours; heating is turned on.	ºC
T_off_max	maximum temperature in a room during unoccupied hours; cooling is turned on.	ºC
tau_glass	glass windows transmissivity	-
alpha_wall	wall's radiation absorption coefficient	-
alpha_floor	floors' radiation absorption coefficient	-
alpha_glass	windows' radiation absorption coefficient	-
u_wall	wall U value	W/m^2.K
u_roof	roof U value	W/m^2.K
u_floor	floor U value	W/m^2.K
u_glass	glass U value	W/m^2.K
cp_roof	roof specific heat capacitance	J/m^2.K
cp_wall	wall specific heat capacitance	J/m^2.K
air_change_hour	air changes per hour due to infiltrations	1/h
renewal_air_per_person	fresh air changer per person	m^3/s.person
vol_dhw_set	volume of daily water consumption	m^3
Q_gain_per_floor	heat gains due to miscellaneous equipment (e.g. lamps, computers...)	W
emissivity_wall	walls radiation emissivity coefficient	-
emissivity_glass	windows' radiation emissivity coefficient	-
ref_system_eff_equipment_heating	Efficiency of the heating equipment	kWh
ref_system_eff_equipment_cooling	COP of the cooling equipment	kWh
real_heating_monthly_capacity	Real monthly data - for each month of the year	kWh
real_heating_yearly_capacity	Real yearly data - single value	kWh
real_cooling_monthly_capacity	Real monthly data - for each month of the year	kWh
real_cooling_yearly_capacity	Real yearly data - single value	kWh



New Sink

Get started by filling in the information below to create your new Sink. This Sink will be attached to your currently selected Institution.

Information

Template: Building

Location: 40.77846164090355 lat; -8.585901260375978 lng

☐ Fill in your own coordinates

Properties

Name (Required): test buiild

Building type (Required): Select an option...
- office
- hotel
- residential

Building's main facade orientation

Number of floors (Required)

Figure 71 - Printscreen of the template for adding a building to the platform

b. Greenhouse

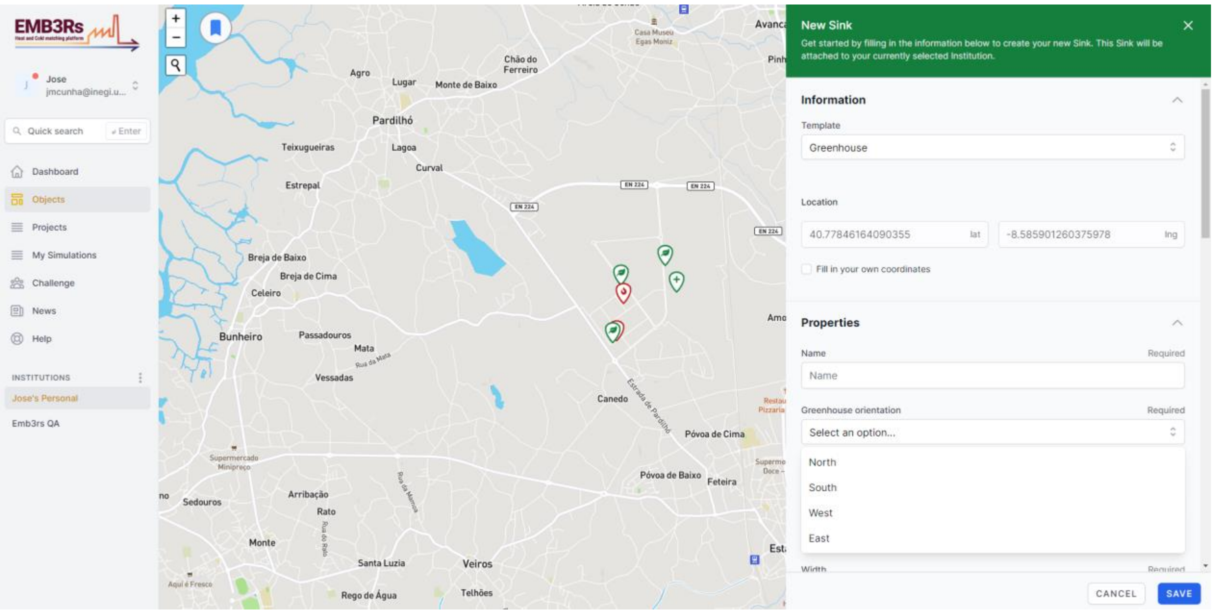
This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

This routine's objective is to apply a thermal model to simulate the hourly heat consumption over the year, based on the greenhouse parameters (see Table 43). The cooling needs are not computed since it is considered that when cooling is required the greenhouse cover is open to climatize with outside air temperature. Similarly to the building routine, for each location it is obtained through *pvlib* package the respective climate weather data.

Mandatory Input		
var name	**description**	**units**
latitude	latitude of the location	º
longitude	longitude of the location	º
width	width of greenhouse main facade	m
length	greenhouse length	m
height	greenhouse height	m
saturday_on	if working -1- or not -0- on saturday	-
sunday_on	if working -1- or not -0- on Sunday	-
shutdown_periods	periods of days during the year the greenhouse heating actuation is deactivated	days
daily_periods	daily hourly period the greenhouse heating actuation is activated	hour
greenhouse_orientation	greenhouse's main facade orientation	-
artificial_lights_system	if the greenhouse has artificial lighting system or not	-
hours_lights_needed	hours of light the plant needs (accounting with daily solar hours); only if lights_on =1	hours
greenhouse _efficiency	greenhouse air infiltration tightness; 1- tight cover with low infiltrations; 2 - medium sealing; 3 - leaky cover	-
ref_system_fuel_type	Fuel type associated; e.g. "natural_gas","electricity","biomass","fuel_oil","none"	-
Optional Input		
f_c	characterization of tightness of the cover to air infiltration [17]	-
T_heat_on	minimum allowed greenhouse interior air temperature for which the space heating starts actuating	ºC

supply_temperature_heat	heaters working fluid temperature; supply temperature < target temperature (supply temperature to the DHN heat exchanger)	ºC
target_temperature_heat	heaters working fluid temperature; target temperature < supply temperature (temperature obtained from the DHN heat exchanger)	ºC
leaf_area_index	average leaf area index of a plant	-
rh_air	relative humidity	-
u_cover	cover thermal conductivity	W/m. K
indoor_air_speed	indoor air velocity	m/s
leaf_length	characteristic length of a plant leaf	m
tau_cover_long_wave_radiation	Cover transmissivity coefficient to long-wave radiation	-
emissivity_cover_long_wave_radiation	Cover emissivity coefficient to long-wave radiation	-
tau_cover_solar_radiation	transmissivity coefficient to solar radiation	-
power_lights	light power per square meter	W/m^2
real_monthly_capacity	Real monthly data - for each month of the year	kWh
real_yearly_capacity	Real yearly data - single value	kWh

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121



New Sink

Get started by filling in the information below to create your new Sink. This Sink will be attached to your currently selected Institution.

Information

Template: Greenhouse

Location: 40.77846164090355 lat, -8.585901260375978 lng

☐ Fill in your own coordinates

Properties

Name (Required): Name

Greenhouse orientation (Required): Select an option...

North
South
West
East

Width (Required)

Figure 72 - Print screen of adding a greenhouse to the platform

2.2.4 Simulation

2.2.4.1 Description

The graphical scheme of how the user experiences the module in **simulation** mode is illustrated below.

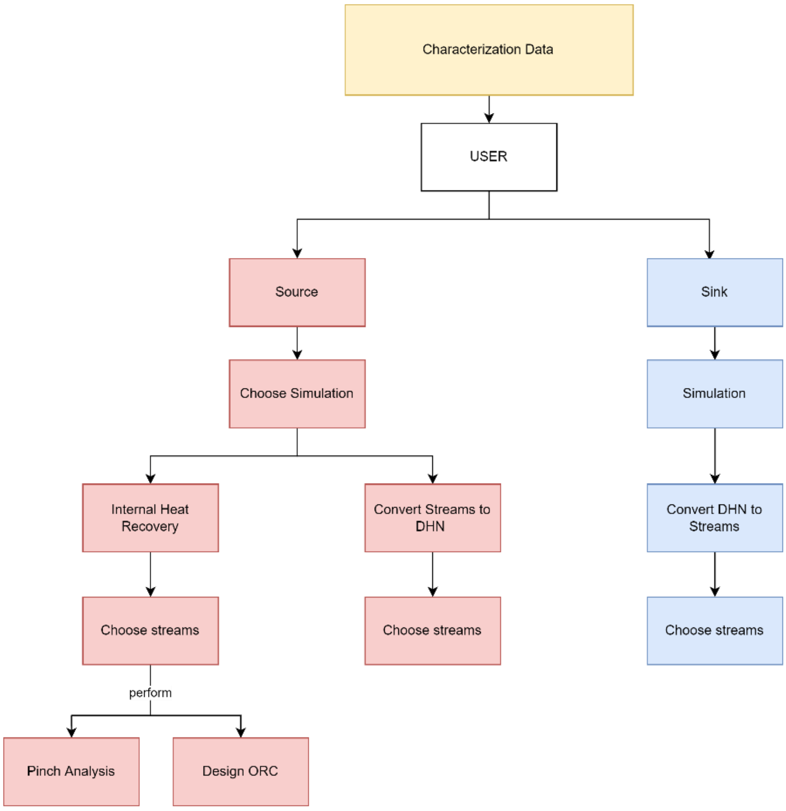


Figure 73- Simulation graphical scheme

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

2.2.4.2 Source Specific Simulations

The internal heat recovery consists in recovering the heat within the industry by designing a network of heat exchangers – applying the pinch analysis - that promotes the exchange of heat within the processes, thus, minimizing heat/cooling supply by external equipment; by utilizing the excess heat to produce electricity – implementing an Organic Rankine Cycle (ORC).

With the streams already characterized, the user with a source object can perform an **internal heat recovery simulation**:

1. **The user enters the platform and chooses a source**
2. **The user chooses to perform internal heat recovery, pinch analysis or ORC design, simulation**
3. **The user selects the streams to be analyzed**
4. **The pinch analysis and ORC module perform several analysis to the streams selected and provide as output the best design solutions and an HTML report**
5. **The output data is sent to the BM so that it performs a financial analysis to the designs**
6. **The data, both from the CF and BM, is displayed to the user**

2.2.4.2.1 ORC

This routine assesses as an internal heat recovery alternative, the design of a organic rankine cycle – ORC - for the production of electricity for self-consumption or to sell to the grid. The routine performs a series of organic rankine cycles, and returns as output the 3 (default number) best ORC designs in terms of lowest cost of electrical generation (€/kW).

Table 90 - ORC design inputs

Mandatory Input		
var name	**description**	**units**
location	[latitude,longitude]	º
get_best_number	number of best conversion cases Default:3	-
streams	See Table 10	-
fuels_data	Fuels price [€/kWh] and CO2 emission [kg CO2/kWh] for: natural gas, fuel oil, electricity, biomass.	-
Optional Input		
orc_T_evap	evaporator temperature	ºC
orc_T_cond	condenser temperature	ºC

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

To the user that performs the design of the ORC, the CF shows technical data, such as, the ORC capacity and conversion efficiency, and the excess heat streams utilized. Moreover, economic data is provided about the ORC, such as turnkey, O&M variable and fixed costs (see Table 91). Moreover, a HTML report is provided in a more user-friendly format.

Table 91 - Report User; ORC design

ORC – Report data		
var name	**description**	**units**
streams_id	array with the converted streams ID	-
electrical_generation_nominal	nominal electrical supply capacity	kW
electrical_generation_yearly	array with hourly electrical supply capacity	kWh
excess_heat_supply_capacity	excess heat available supply capacity	kW
conversion_efficiency	conversion efficiency of heat to electricity	-
turnkey	equipment turnkey	€
om_fix	equipment turnkey O&M fix	€/year
om_var	equipment turnkey O&M variable	€/kWh

As an example, for the following streams working all year presented in Table 92 were analyzed.

Table 92 - Example ORC design input

Example: Streams data					
Stream ID	**T_supply**	**T_target**	**Flowrate**	**Cp**	**Fluid**
units	ºC	ºC	kg/h	kJ/kg.h	-
1	400	250	32123	1.4	Flue gas
2	360	90	155897	1.4	Flue gas

The data of one of the designed solutions is presented in Table 93

Table 93 - Example ORC design output

Example: ORC designed		
var name	**value**	**units**
streams_id	[1]	-
electrical_generation_nominal	1352	kW
electrical_generation_yearly	11385430	kWh
excess_heat_supply_capacity	17399	kW
conversion_efficiency	0.086	-
turnkey	3409168	€

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

om_fix	73434	€/year
om_var	0.0014	€/kWh

2.2.4.2.2 Pinch analysis

In brief, the pinch analysis is a theoretical method that, based on fundamental thermodynamics, analyzes the heat flow through the industry's processes with the aim of recovering heat within those processes by creating a heat exchanger - HX - network, reducing energy needs and CO_2 emissions (for more information see [2] [3]). The implemented routine generates the maximum of different design combinations and provides the best solutions to the user. Moreover, an HTML report is provided in a more user-friendly format.

The user must provide in the platform the streams to be analyzed as well as the other parameters shown in

Table 94 - Pinch analysis inputs

Mandatory Input		
var name	**description**	**units**
pinch_delta_T_min	delta temperature for pinch analysis	ºC
all_input_objects	array with equipments/processes/isolated_stream dictionaries	-
fuels_data	Fuels price [€/kWh] and CO2 emission [kg CO2/kWh] for: natural gas, fuel oil, electricity, biomass.	-
Optional input		
lifetime	technologies lifetime for economic analysis. (default: lifetime=20 years)	year
number_output_options	Number of optimal output design options for each of the analysis (default: number_output_options=3)	-
interest_rate	Interest rate considered for BM (default:0.04)	-

To the user that performs the pinch analysis, the CF displays a graphical representation of the HX network designed, as presented in Figure 74 . Furthermore, general data for the pinch analysis is provided such as, the streams utilized, HX network turnkey and O&M costs, minimum and designed hot and cold utilities, and pinch temperature (Table 95). The user is also able to check specific data regarding each HX designed, as shown in Table 96. Finally, for the equipment which provide heat/cooling to the streams which will exchange it with the HX, there are energy, and thus costs and emissions savings (Table 97). As an example, considering a stream of milk which is heated from 20ºC to 60ºC by a stream which is cooling down, e.g. hot whey at 90ºC, the boiler which was providing the heat to the milk stream will have its capacity reduced, consuming less fuel and reducing its emissions.

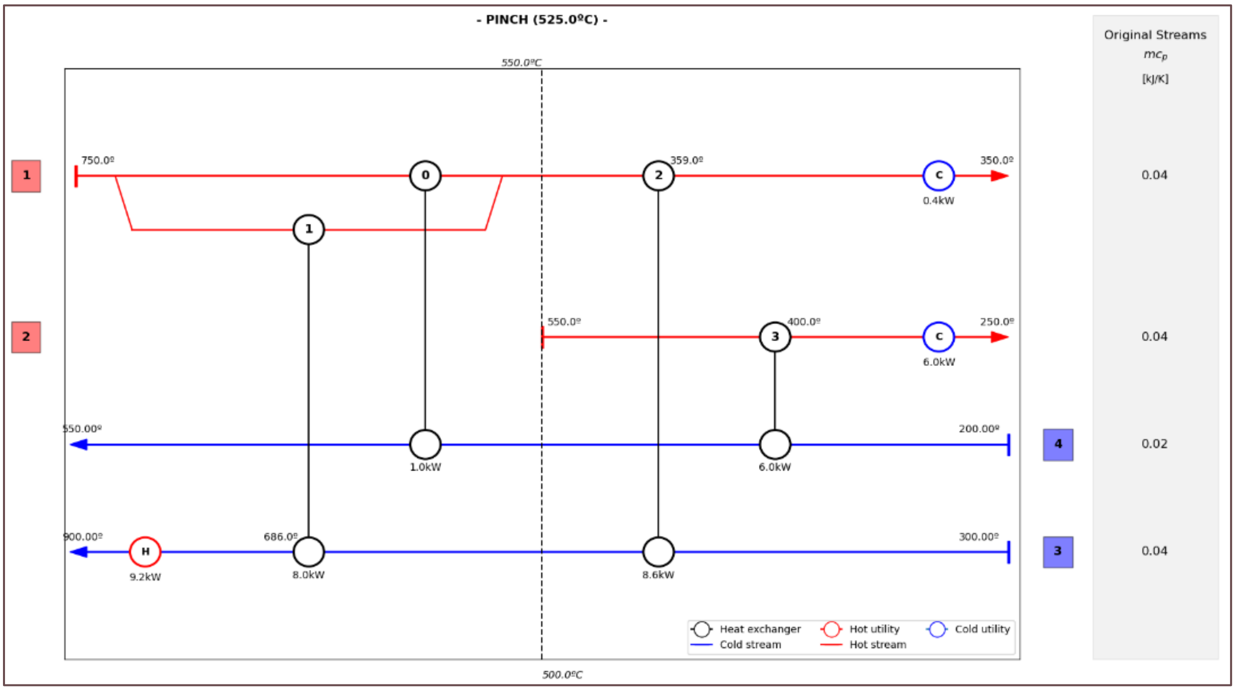


Figure 74 - Graphical representation of heat exchanger network

Table 95 - Report User; Pinch analysis, heat exchangers network general data

Heat Exchanger Network general data		
var name	**description**	**units**
ID	designed solution ID	-
streams	streams in pinch design	-
capex	design turnkey	€
om_fix	yearly O&M fixed costs	€/kW
hot_utility	power of the hot utility needed	kW
cold_utility	power of the cold utility needed	kW
discount_rate	discount rate to be applied on the business analysis	-
pinch_temperature	design pinch temperature	ºC
theo_minimum_hot_utility	theoretical power of the hot utility needed	kW
theo_minimum_cold_utility	theoretical power of the cold utility needed	kW

Table 96 -- Report User; Pinch analysis, heat exchangers techno-economic data

Heat Exchanger techno-economic data		
var name	**description**	**units**
HX_Power	heat exchanger design power	kW
HX_Hot_Stream	hot stream ID	-
HX_Cold_Stream	cold stream ID	-
HX_Original_Hot_Stream	original hot stream ID (it can be different of HX_Hot_Stream if a stream split occurs - a new id is given to the split)	-

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

HX_Original_Cold_Stream	original cold stream ID (it can be different of HX_Cold_Stream if a stream split occurs - a new id is given to the split)	-
HX_Cold_Stream_flowrate	mass flowrate	kg/h
HX_Hot_Stream_flowrate	mass flowrate	kg/h
HX_Type	type of heat exchanger	-
HX_Turnkey_Cost	heat exchanger turnkey cost	€
HX_OM_Fix_Cost	heat exchanger O&M cost	€/kW
HX_Hot_Stream_T_Hot	hot stream hot temperature	ºC
HX_Hot_Stream_T_Cold	hot stream cold temperature	ºC
HX_Cold_Stream_T_Hot	cold stream hot temperature	ºC
HX_Cold_Stream_T_Cold	cold stream cold temperature	ºC
Storage	storage volume	m^3
Storage_Turnkey_Cost	storage turnkey cost	€
Total_Turnkey_Cost	whole package (heat exchanger + storage) turnkey cost	€
Recovered_Energy	amount of yearly energy recovered	kWh

Table 97 - Report User; Pinch analysis, equipment detailed savings

Heat exchanger network – Equipment detailed savings		
var name	**description**	**units**
Equipment_ID	equipmet ID	-
CO2_Savings_Year	yearly CO_2 emissions saved	kg CO_2
Recovered_Energy	yearly energy saved	kWh
Savings_Year	yearly monetary savings	€

As an example, for the following streams working all year without equipment associated,

Table 98 - Example streams data

Example: Streams data				
Stream ID	**T_supply**	**T_target**	**Flowrate**	**Cp**
units	ºC	ºC	kg/h	kJ/kg.h
1	750	350	81	2
2	550	250	72	2
3	300	900	77.4	2
4	200	550	36	2

and pinch_delta_T_min = 50ºC.

One of the soltuions is the one represented in Figure 74. The general HX network data is the following,

Table 99 - Example Pinch analysis design

Example: General Network data		
var name	**value**	**units**
ID	0	-
streams	[1,2,3,4]	-
capex	5972	€
om_fix	597	€/kW
hot_utility	9.2	kW
cold_utility	6.4	kW

pinch_temperature	525	ºC
theo_minimum_hot_utility	9.2	kW
theo_minimum_cold_utility	6.4	kW

The data for HX 1 (see Figure 74) is,

Table 100 - Example Pinch analysis HX design

Example: Heat Exchanger 1 data		
var name	**value**	**units**
HX_Power	8	kW
HX_Hot_Stream	1	-
HX_Cold_Stream	3	-
HX_Original_Hot_Stream	1	-
HX_Original_Cold_Stream	3	-
HX_Cold_Stream_flowrate	0.021	kg/h
HX_Hot_Stream_flowrate	0.02	kg/h
HX_Type	hx_plate	-
HX_Turnkey_Cost	1570	€
HX_OM_Fix_Cost	157	€/kW
HX_Hot_Stream_T_Hot	750	ºC
HX_Hot_Stream_T_Cold	550	ºC
HX_Cold_Stream_T_Hot	686	ºC
HX_Cold_Stream_T_Cold	500	ºC
Storage	0	m^3
Storage_Turnkey_Cost	0	€
Total_Turnkey_Cost	1570	€
Recovered_Energy	70080	kWh

2.2.4.3 Source/Sink DHN Conversion

This is not a specific CF simulation, but rather a whole platform simulation since different modules are involved. The steps presented in this document ilustrate the role of the CF.

For the conversion of both sinks and sources, the user flow will be:

1. The user defines the area where the Open Street Map (OSM) network should be loaded. Existing grids from the CF module are shown on the map if added.
2. All sources and sinks on the location are displayed automatically to the user. The user can remove sinks and sources manually from the automatic selection made.
3. The user sets the flow, return, ambient and ground temperature for the DHN. Predefined values are loaded from the KB.
4. The CF module computes a first conversion of both sinks/sources and sends them to the GIS so that it defines the grid network
5. The GIS and TEO modules will iterate to find the optimum exchange capacities according to the grid data and send the information to the CF
6. The CF will update the conversions according to the grid losses and perform a new conversion of the source and sink streams. The data is then sent to the TEO module to iterate again with the GIS, returning to step 5. When the iteration

converges, the data is sent to the Business Module so that it performs a financial analysis.

7. After all the modules run, the optimal DHN solution, according to the user requirements, is displayed to the user (optimal grid design, sources and sinks technologies and capacities, as well as the techno-economical , business and market analysis).

2.2.4.3.1 Convert Sources

This routine designs and estimates the costs of converting the available heat of its excess heat streams to the District Heating Network - DHN. The data the user must provide is in Table 101 and Table 102. For each excess heat stream available the conversion technologies are designed, e.g. a heat exchanger to recover the heat from a hot stream and supply it to the DHN. This routine runs together with the TEO and GIS modules in order to design the correct links and provide a more realistic estimate on the DHN.

Table 101 - Convert Sources input

Convert Sources input		
var name	**description**	**units**
group_of_sources	array with sources dictionaries	-
sink_group_grid_supply_temperature	DHN supply temperature	ºC
sink_group_grid_return_temperature	DHN return temperature	ºC
existing_grid_data	Existent grid connection point data	-

Table 102 - Convert Sources input; group of sources dictionary

group of sources - dictionary		
var name	**description**	**units**
id	source ID	-
location	[latitude,longitude]	º
fuels_data	Fuels price [€/kWh] and CO2 emission [kg CO2/kWh] for: natural gas, fuel oil, electricity, biomass.	-

2.2.4.3.2 Convert Sinks

This routine designs the conversion technologies necessary so that the DHN can meet the sinks' heating/cooling needs. For a given group of sinks, the DHN supply and return temperatures are set by the sink which demands the largest temperature. The data the user must provide is in Table 103 and Table 104Table 102. Initially, grid specific technologies are designed to meet the heating/cooling requirements of the group of sinks. The DHN specific technologies are, as the name suggests, technologies designed to cover the heating and cooling needs of whole group of sinks. Then, for each sink stream the conversion technologies are designed.

Table 103 - Convert Sinks input

Convert Sinks input		
var name	**description**	**units**

group_of_sinks	array with the sinks to be converted	-
grid_supply_temperature	Grid supply temperature	ºC
grid_return_temperature	Grid return temperature	ºC

Table 104 - Convert Sinks input; group of sinks

group of sinks - dictionary		
var name	**description**	**units**
id	sink ID	-
location	sink location as [latitude,longitude]	º
fuels_data	Fuels price [€/kWh] and CO2 emission [kg CO2/kWh] for: natural gas, fuel oil, electricity, biomass.	-

Regarding the Source/Sink Conversion DHN, there are no CF reports to show to the user.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

2.3 References

[1] “SCM Frigo,” [Online]. Available: https://www.scmfrigo.com/.

[2] I. C. Kemp, Pinch Analysis and Process Integration, Elsevier, 2007.

[3] L. March, *Introduction to Pinch Technology,* 1998.

[4] “University of Oklahoma - Pinch,” [Online]. Available: https://www.ou.edu/class/che-design/che5480-13/PINCH%20ANALYSIS%20Part%204-%20Pinch%20Design%20Method-Maximum%20Energy%20Recovery%20Networks.pdf.

[5] “Example - Special Case,” [Online]. Available: https://processdesign.mccormick.northwestern.edu/index.php/Pinch_analysis.

[6] M. U. M. I. A. P. Byung-Sik Park, “Review of Organic Rankine Cycle experimental data trends,” *Energy Conversion and Management,* 2018.

[7] ElectraTherm.

[8] *Infinity Turbine.*

[9] B. G.P.Hammonda, “Heat recovery opportunities in UK industry,” *Applied Energy,* 2013.

[10] [Online]. Available: https://bigladdersoftware.com/epx/docs/8-2/engineering-reference/index.html.

[11] J. R. Garratt, The Atmospheric Boundary Layer.

[12] J.A.Palyvos, “A survey of wind convection coefficient correlations for building envelope energy systems’ modeling,” *Applied Thermal Engineering,* 2007.

[13] G. Brown, Sun, Wind, and Light: Architectural Design Strategies, 2013.

[14] Y. Z. L. M.-H. V. I. H. Ivan Korolija, “Selecting HVAC systems for typical UK office buildings,” 2009.

[15] “ HVAC Applications -"Service Water Heating",” in *ASHRAE Handbook*, 1999.

[16] H. G. K. T. Md Shamim Ahamed, “A quasi-steady state model for predicting the heating requirements of conventional greenhouses in cold regions,” *Information Processing in Agriculture,* 2017.

[17] H. G. T. Md. Shamim Ahamed, “Development of a thermal model for simulation of supplemental heating requirements in Chinese-style solar greenhouses,” *Computers and Electronics in Agriculture,* 2018.

[18] ,. M. S. A. ,. C. M. G. Shuyao Dong, “A Time-Dependent Model for Predicting Thermal Environment,” *Energy Systems and Applications in Agriculture,* 2021.

[19] “EngineeringToolbox,” 2021. [Online].

[20] “Eurostat API,” 2021. [Online]. Available: https://pypi.org/project/eurostatapiclient/.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

3 GIS MODULE

3.1 System Manual

3.1.1 Module Overview

The Geographical Information System (GIS) modules' purpose within the EMB3RS platform is to analyze the network dimension and bring in the spatial dimension between sources and sinks. The application of the GIS is tailored for looking into the option of reusing the excess heat/cold at a certain distance within a District Heating and Cooling (DHC) system. It assumes a potential network solution between a particular set of sources and sinks among the Open Street Map (OSM) road network. The related investment costs into the grid and the corresponding heat/cold losses are calculated based on that network solution.

The GIS module receives information from the core functionalities (CF) module, the knowledge base (KB) as well as the platform/user and sends information to the other calculation modules, namely the CF module, Techno-Economic Optimization (TEO) module, Market Module (MM), and Business Module (BM).

The main features of the GIS module calculations are:

- **DHC network calculation based on different heat/cold sources and sinks (routing),**
- **calculation of the heat/cold losses and investment costs of the resulting DHC network solution (heat loss and cost calculation).**

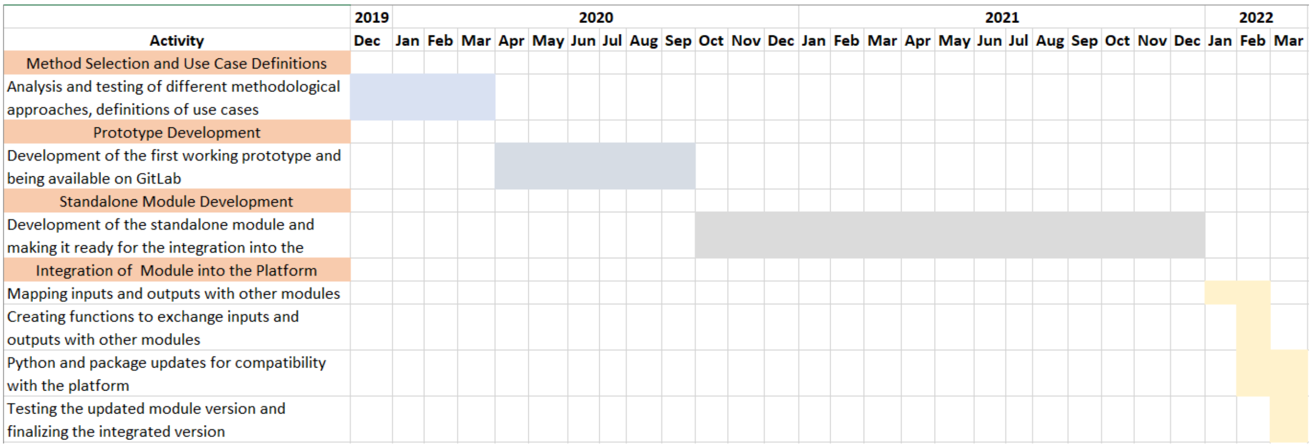
The main calculation steps of the GIS module are given below:

- **The GIS gets inputs -information on sources and sinks- from the CF module**
- **The GIS gets inputs -exchange capacities of sources and sinks- from the TEO module**
    - **The GIS module does not get any information from the TEO module in the first iteration. If advance calculation is chosen, input from the TEO module is used in the subsequent iterations. Otherwise, the GIS module will have a single iteration without the TEO module.**
- **The GIS gets inputs -the project area and information on the existing grid network if any- from the platform/user.**
- **The GIS loads the OSM road network data for the project area**
- **The GIS connects all sources and sinks to the closest road junction (node) on the OSM street network via a straight line to create a closed network**
- **The GIS integrates all source/sinks via the shortest connection into a single network solution**
- **The GIS calculates the related network heat losses/cold "gains"**
- **The GIS calculates the related network investment costs**
- **The GIS outputs the solution to the platform and all the other modules**

3.1.2 Module Development Timeline

The activities done during the module development and respective times spent for them (until M30) are given in Figure 75: Development timeline of the GIS module. After this, the bulk of the work developed focused on supporting the integration on the platform.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121



	2019	2020												2021												2022		
Activity	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar
Method Selection and Use Case Definitions																												
Analysis and testing of different methodological approaches, definitions of use cases	■	■	■	■																								
Prototype Development																												
Development of the first working prototype and being available on GitLab					■	■	■	■	■	■																		
Standalone Module Development																												
Development of the standalone module and making it ready for the integration into the											■	■	■	■	■	■	■	■	■	■	■	■	■	■	■			
Integration of Module into the Platform																												
Mapping inputs and outputs with other modules																										■	■	
Creating functions to exchange inputs and outputs with other modules																											■	
Python and package updates for compatibility with the platform																											■	■
Testing the updated module version and finalizing the integrated version																												■

Figure 75: Development timeline of the GIS module.

3.1.3 Module Requirements and Specifications

3.1.3.1 General Module Architecture

The GIS module consists of two main functions: "create_network" and "optimize_network".

The "create_network" function serves as the first step in the GIS module. It receives inputs from the user/platform, the CF module, and the TEO module -starting from the second iteration if the advanced calculation is chosen-. Then, it returns an Open Street Map graph to the platform.

The "optimize_network" function is the second step of the GIS module. It calculates a thermal network solution and related thermal losses and investment costs. The general model architecture is given in Figure 76.

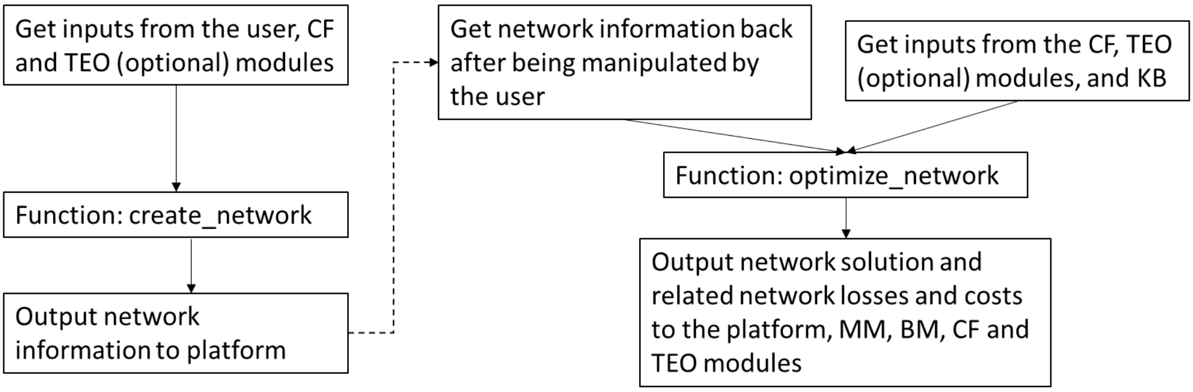


Figure 76: General model architecture of the GIS module.

3.1.3.2 Module Requirements

The integrated version of the GIS module has following dependencies:

- **python = 3.9**
- **osmnx = 1.1.2**
- **scikit-learn = 1.0.2**
- **numpy = 1.22.3**
- **pyomo =5.7**
- **haversine = 2.5.1**
- **pandas = 1.4.1**
- **folium = 0.12.1**

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

- **geopandas = 0.10.2**
- **shapely = 1.8.0**
- **networkx = 2.7.1**
- **colorama = 0.4.4**
- **jsonpickle = 2.1.0**
- **gurobipy = 9.5.1.**

Please note that for the package "gurobipy", the channel "pip" is used. For all the other remaining dependencies "conda-forge" channel is used. Also, the use of an up-to-date package version is recommended. However, the user should be aware that any function used in the model might be deprecated. The user should use the same versions given above, and is encouraged to report any issues found.

The optimization models are modeled with PYOMO and solved with the GUROBI solver. Therefore, a valid GUROBI license is required.

3.1.3.3 Module Features

3.1.3.3.1 Inputs and Outputs

The GIS module gets inputs from the CF and the TEO (optional) modules, the KB, and the platform/user. The inputs and outputs vary for the two main functions of the GIS module. Detailed visualization of the general input-output structure is given in Figure 77. In the following subsections, only inputs/outputs that are provided/received by the user are described.

3.1.3.3.2 Create Network Function

The "create_network" function’s first input is the “Project Area” which defines the area that should be analysed, as defined by the user. This input determines the road network that will be obtained from Open Street Map. This input is essential to have an optimal solution because if the defined area is too small, there is a possibility that the obtained road network is not sufficient to connect all sources and sinks, which leads to an infeasible solution.

“Network Resolution” determines the detail of the road network obtained from OSM. This parameter should be set to “low” if the project area is large. This will lead to a lower detailed road network hence a decreased computational time.

“Existing Grid Network” is an optional input. Users can provide information about the existing grid network via this variable. Existing pipes have a priority in the model to decrease the costs. For each pipe the IDs of sources/sinks connected by the pipe, latitudes and longitudes of those sources/sinks, diameter and length of the pipe in m, total cost of the pipe in EUR, and if the respective pipe is a surface pipe should be defined.

The information about sources and sinks, e.g., their locations and capacities, are obtained via the CF module.

The output of the “create_network” function is an Open Street Map graph. The GIS module gives this output to the platform, and the platform displays it to the user. In this step, the user can manipulate some data. The user could change the bold variables in the outputs of function “create_network” given in Figure 77. Here the user can change the surface type, put a restriction on a road element, define a pipe as a surface pipe, set a pipe as a part of the existing grid network, and define its cost and inner diameter. The Open Street Map graph is then returned to the “optimize network” function as an input.

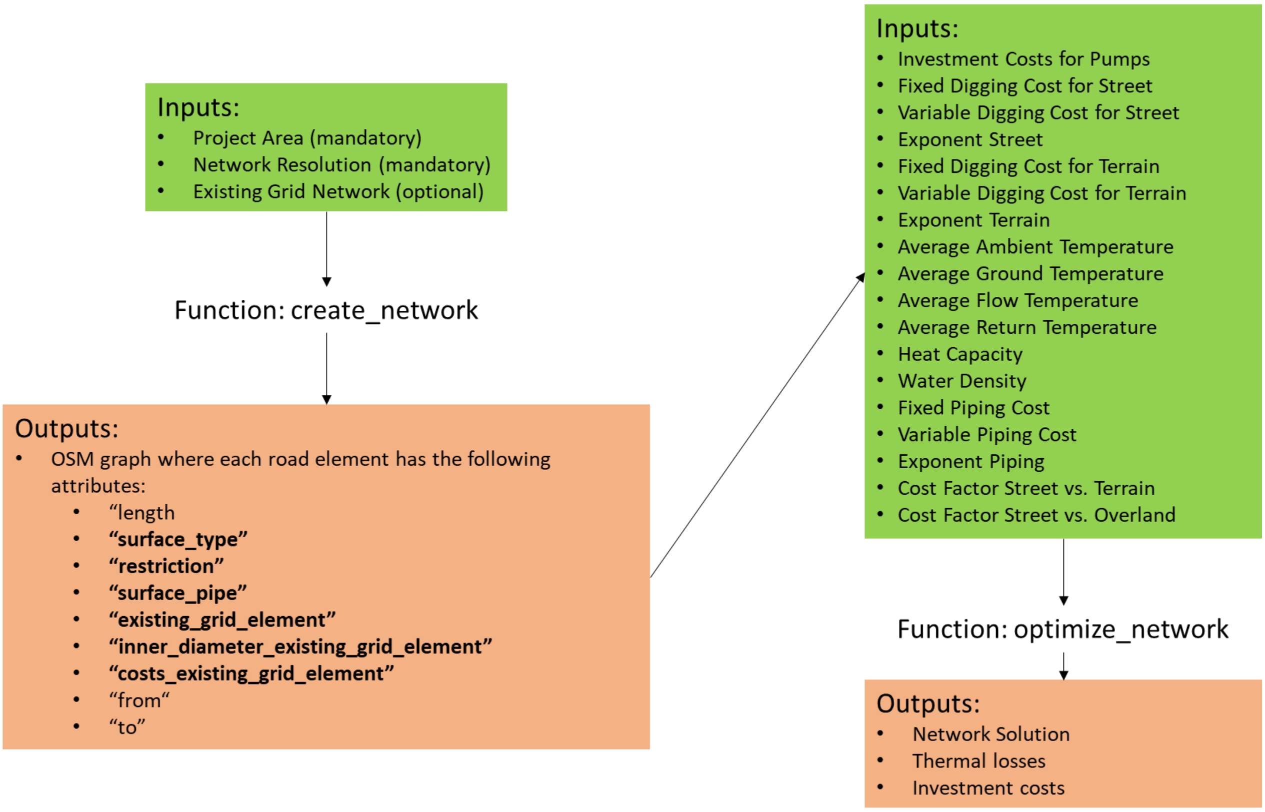


Figure 77: General input-output structure of the GIS module’s two main functions.

3.1.3.3.3 Optimize Network Function

The “optimize_network” function gets the Open Street Map graph back from the platform after it is manipulated by the user. This is the first input of the function.

The remaining inputs are the cost parameters used for network cost calculation and the parameters used for the thermal loss calculation. Labels and descriptions of these parameters are given in Table 105. Although most of the parameters are indicated as mandatory, all of these parameters have a default value stored in the Knowledge Base. If the user does not specify these parameters, default values will be used.

Table 105: Table of inputs required from the user.

Function	**Mandatory**	**Label**	**Description**	**Unit**
create_network	TRUE	Network Resolution	Defines if network resolution is high or low, i.e., how detailed the streets are loaded. If a large network is used, network resolution should be set to low to decrease computational time. Set to high by default.	-
create_network	FALSE	Existing Grid Network	The information on the existing grid network. For each pipe, IDs of sources/sinks connected by the pipe, latitudes, and longitudes of those sources/sinks, diameter and length of the pipe, total cost of the pipe, and if the respective pipe is a surface pipe should be defined.	Diameter in m. Length in m. Total cost of the pipe in EUR.
create_network	TRUE	Project Area	The area that will be considered for the grid. The user could specify the area by drawing a	-

			rectangular shape on the map via the GUI of the platform.	
optimize_network	FALSE	Investment Costs for Pumps	Investment costs for pumps. Set to 0 by default.	EUR
optimize_network	TRUE	Fixed Digging Cost for Street	Fixed digging cost for streets. Set to 350 by default.	EUR/m
optimize_network	TRUE	Variable Digging Cost for Street	Variable digging cost for streets. Set to 700 by default.	EUR/m²
optimize_network	TRUE	Exponent Street	The exponent of the digging cost for the street. Set to 1.1 by default.	-
optimize_network	TRUE	Fixed Digging Cost for Terrain	Fixed digging cost for terrains. Set to 200 by default.	EUR/m
optimize_network	TRUE	Variable Digging Cost for Terrain	Variable digging cost for terrains. Set to 500 by default.	EUR/m²
optimize_network	TRUE	Exponent Terrain	The exponent of the digging cost for the terrain. Set to 1.1 by default.	-
optimize_network	TRUE	Average Ambient Temperature	Yearly average ambient temperature. Set to 25 by default.	°C
optimize_network	TRUE	Average Ground Temperature	Yearly average ground temperature. Set to 8 by default.	°C
optimize_network	TRUE	Average Flow Temperature	Yearly average flow temperature. Set to 100 by default.	°C
optimize_network	TRUE	Average Return Temperature	Yearly average return temperature. Set to 70 by default.	°C
optimize_network	TRUE	Heat Capacity	Heat capacity at a specific temperature (average of flow and return temperatures). Set to 4.18 by default.	J/kgK
optimize_network	TRUE	Water Density	Water density at a specific temperature (average of flow and return temperatures). Set to 1000 by default.	kg/m³
optimize_network	TRUE	Fixed Piping Cost	The fixed component of the piping cost. Set to 50 by default.	EUR/m
optimize_network	TRUE	Variable Piping Cost	The fixed component of the piping cost. Set to 700 by default.	EUR/m²
optimize_network	TRUE	Exponent Piping	The exponent of the piping cost. Set to 1.3 by default.	-
optimize_network	TRUE	Cost Factor Street vs. Terrain	Determines how much cheaper it is to lay 1 m of pipe into a terrain than a street. Expressed in decimals: 0.1 means it is 10% cheaper.	Decimals
optimize_network	TRUE	Cost Factor Street vs. Overland	Determines how much cheaper it is to place 1 m of the pipe over the ground than putting it into the street. Expressed in decimals: 0.4 means it is 40% cheaper.	Decimals

The outputs of the “optimize_network” function are:

- **the network solution visualized on Open Street Map (OSM),**
- **the network losses,**
- **the investment costs.**

The GIS module also outputs the potential grid area independent of the network solution. This potential grid area shows all the possible routes for the pipes in OSM.

3.1.3.4 Contributions and Requirements for the Knowledge Base

The GIS module interacts with the KB for getting the default values of the parameters, as described below. The default values stored in the KB are given in Table 106. For all of these parameters, the model first checks if there is a user-defined value. If not, then the model goes to the KB and gets the default value from there. Descriptions of these parameters can be found in Table 105.

Table 106: Default values of the parameters in the Knowledge Base.

Label	**Parameter**	**Default Value**	**Unit**
Investment Costs for Pumps	invest_pumps	0	EUR
Variable Digging Cost for Street	vc_dig_st	700	EUR/m^2
Fixed Digging Cost for Street	fc_dig_st	350	EUR/m
Exponent Street	vc_dig_st_ex	1.1	-
Fixed Digging Cost for Terrain	fc_dig_tr	200	EUR/m
Variable Digging Cost for Terrain	vc_dig_tr	500	EUR/m^2
Exponent Terrain	vc_dig_tr_ex	1.1	-
Cost Factor Street vs. Terrain	factor_street_terrain	0.1	-
Cost Factor Street vs. Overland	factor_street_overland	0.4	-
Fixed Piping Cost	fc_pip	50	EUR/m
Variable Piping Cost	vc_pip	700	EUR/m^2
Exponent Piping	vc_pip_ex	1.3	-
Average Ambient Temperature	ambient_temp	25	°C
Average Ground Temperature	ground_temp	8	°C
Average Flow Temperature	flow_temp	100	°C
Average Return Temperature	return_temp	70	°C
Heat Capacity	heat_capacity	4.18	J/kgK
Water Density	water_den	1000	kg/m3

3.1.3.5 Functioning of the GIS Module

The network solution is found based on a road network graph obtained from OSM. Every road element of the OSM network is a possible pathway where a pipe could be laid. The user can set costs to every road element, add new road elements, or restrict edges. Previous calculations of test examples have shown that finding the shortest path to connect all selected sources and sinks is most beneficial in almost any case

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

for finding the cheapest network solution. To further improve the network solution based on the shortest path, the ground conditions are also considered for finding the optimal grid solution, as digging costs may vary with changing surface conditions while the costs for the pipes stay the same regardless of the ground composition. These digging costs are determined by two surface classes, one for streets and one for terrain surface. No surface class is assigned if a pipe goes on the surface. The network calculation always aims to find one single network between all sources and sinks.

3.1.3.5.1 Highlights of the Network Calculation

User-defined sources and sinks are integrated into the network.

The OSM network is loaded, and all sources and sinks are connected to the closest road junction. A pipe element – also called a pipe section – is always defined by a road section defined by two junctions.

There are bilateral dependencies between the GIS and the TEO module: the TEO module requires the loss and investment cost calculation from the GIS module, and the GIS module requires the actual energy flows from the TEO module to decide on the correct energy transmission and distribution. Thus, an iteration is implemented between the TEO and the GIS in order to calculate appropriate losses and costs.

The algorithm aims to find the cheapest pipe network between all chosen sources and sinks within the road network – constraints in terms of digging costs can be set for all road elements within two different classes and then be assigned to every individual road element. These classes are divided into streets and terrain. The user can set these classes' variable and fixed cost components. Surface pipes will have digging costs of 0.

The user may change the cost difference factor of costs between the class street and terrain. This factor determines how much cheaper it is to lay a pipe into terrain than on the street and is used to optimize the network. The length of each road network element is then weighted with this factor and put into the optimization. This factor is proposed by the KB and can be changed by the user. If this factor is defined as 0.15, the model assumes one meter of pipe laid into a terrain surface is 15% cheaper than one meter of pipe laid into a street surface. This is then used for weighting each road element which is a potential pathway for a pipe. The user can further define such a cost difference factor between a street and no surface (surface pipe). This applies if routes for surface pipes are investigated.

Existing grid networks will be loaded for the project area. The user can restrict roads manually and add new road elements. The user can set the road element to a restricted element (must not be part of the network) or to a must-build element (has to be part of the network). This feature is useful if the user specifically wants to include or exclude certain pathways.

The user can manually declare a road element as an existing grid element within the GIS module. The module prioritizes existing grid elements. The transmission capacity of existing elements has to be added by the user.

The GIS optimization can consider an existing grid if the user provides the information. Existing grid elements are treated as priority edges. The GIS module always tries to connect the closest points between the loaded OSM network and the existing network via the shortest Euclidian distance.

Based on the network solution calculated within the network calculation task of the GIS module, the network length and the needed transmission capacities are determined. Combining the information of the found network length as well as pipe capacities, the temperature difference between flow and return temperature, the temperature difference between the pipes and the ambient/ground temperature, the water density at a specific temperature, and the heat capacity of water, a nominal loss calculation based on hot water as heat carrier is performed. The determination of pressure losses is not included in the analysis. The seasonal variations of the grid temperatures are also neglected in the module – yearly averages are used instead -. The cost calculation for the network consists of the investment costs of the grid and investments in pumps that can be added optionally by the user. O&M costs of the grid are considered within the TEO module.

3.1.3.5.2 The Highlights of the Network Costs and Losses

- **The necessary capacities of all pipes are found by the GIS module and then translated into a corresponding pipe diameter for every pipe element.**
- **Nominal losses are determined for every pipe element in Watts and between each source and sink.**
- **Nominal “gains” within cooling grids are determined for every pipe element in Watts and between each source and sink.**
- **Supply and return temperature of the grid are obtained from the KB if the user does not define them.**
- **The costs include the grid investment and investments into pumps which the user can add. All investments related to the technologies between the grid and the objects (source/sinks), e.g., heat exchangers connecting the grid to the source/sink and the O&M costs, are included in the TEO module.**
- **The user can change multiple inputs for the loss (e.g., flow/return temperature and ambient/ground temperatures) and network cost (e.g., digging/pipes) calculation.**
- **Road elements defined as existing grid elements within the GIS require a capacity value input from the user. The GIS module then considers this capacity constraint as a fixed value. The module calculates losses for that element; the user may add optional costs for that element.**
- **The user can set road elements to potential surface or ground pipes – this will decide if the ambient or the ground temperature will be used for the loss calculation if the optimization chooses this road element. Furthermore, different calculation formulae are used to calculate surface and ground pipes loss. If a road element is declared as a possible pathway for a surface pipe, the digging costs of that element will be considered 0.**

The loss for ground pipes is calculated based on the formulae taken from the THERMOS tool [1]. The required pipe capacity is converted into a certain pipe diameter using the equation below:

$$P(d)[kW] = (-0.4834 + 4.7617(d^{0.3701}))\delta t\ \rho\ c$$

P represents the pipe capacity, d the pipe diameter, δt the temperature difference of supply and return temperature, ρ the water density at the mean of supply and return temperature, and c the specific heat capacity of water. After translating the pipe capacity into a necessary pipe diameter, the following formula can be used to calculate the heat losses depending on the pipe diameter:

$$P_{loss}\left[\frac{W}{m}\right] = \delta t'(0.16805\ ln(d) + 0.85684)$$

$\delta t'$ is given by the difference of the mean supply and return temperatures and the ground temperature. The diameter of the pipes is calculated by the formula above. The losses for surface pipes in W/m are, however, calculated differently than the ones for ground pipes. The formula used for calculation is shown below and was taken from [2]:

$$P_{loss}\left[\frac{W}{m}\right] = \frac{2\pi(\delta t^*)}{\frac{1}{\lambda_D}\ln\left(\frac{r_D}{r_R}\right) + \frac{1}{r_D\alpha_a}}$$

In order to be able to conduct the surface loss calculations, literature values were obtained from [2] for the calculation. Approximate average values for α_a and λ_D are set to 23.2 and 0.026 respectively. Here δt^* stands for the difference between the ground temperature and the mean of the supply and return temperatures. To simplify the loss calculation process, a pre-calculation is done for

$$preloss\left[\frac{W}{mK}\right] = \frac{2\pi}{\frac{1}{\lambda_D}\ln\left(\frac{r_D}{r_R}\right) + \frac{1}{r_D\alpha_a}}$$

considering different inner diameter classes, the typical pipe-specific inputs r_D and r_R of each inner diameter class, and a medium level of pipe insulation. Results of this pre-calculation for different inner diameter classes are given in Table 107. In the model, the inner diameter value of a surface pipe is looked up in this table, and the preloss value of closest inner diameter value is chosen. This value is then multiplied by δt^* to calculate the final loss value.

Table 107: Preloss values for different inner diameter classes.

inner diameter [m]	preloss [W/mK]
0.02	0.115994719
0.025	0.138092835
0.032	0.151097572
0.04	0.171799705
0.05	0.193944277
0.065	0.219829985
0.08	0.23157219
0.1	0.241204678
0.125	0.280707496
0.15	0.320919872
0.2	0.338510753
0.25	0.326870585
0.3	0.37625986
0.35	0.359725183
0.4	0.372648019
0.45	0.42747404
0.5	0.359725659
0.6	0.420023799
0.7	0.478951908
0.8	0.540336445
0.9	0.600053257
1	0.662751592

Cooling grids rather have unwanted “heat gains” than heat losses. The calculation of cooling losses is conducted in the same way as mentioned above, apart from using the absolute temperature difference. E.g., the temperature difference may be negative for a cooling grid, as the grid temperature is lower than the ground temperature. In this case, the negative temperature difference is negated.

All network costs are calculated by the formulae below:
The unit digging costs are given by:

$$\text{Unit Digging Cost}\left[\frac{EUR}{m}\right] = A + [(d)(B)]^{x},$$

where A and B represent cost parameters dependent on a certain location and surface type. While A represents the share of fixed cost per meter of pipe, B represents the variable share depending on the pipe diameter d and pipe length l. x is the exponent of the digging cost formula. The total digging cost is then calculated by multiplying the unit digging cost by the pipe length l.
The unit pipe costs are given by:

$$\text{Unit Piping Cost}\left[\frac{EUR}{m}\right] = C + [(d)(D)]^{x},$$

where C is the fixed share cost per meter of pipe, D represents the variable share depending on the pipe diameter d and length l. x is the exponent of the piping cost formula. The total piping cost is calculated by multiplying the unit piping cost by the pipe length l.
Calculating the costs of surface pipes works in principle the same way as for ground pipes. The only difference is that the digging cost shares are neglected.

3.1.3.6 Simulation Process

3.1.3.6.1 Actors

- Platform User
- GIS module
- Knowledge Base
- CF Module
- TEO Module

3.1.3.6.2 Pre-conditions

- The user must be logged into the platform
- The user must have determined a location
- The user must have defined a project area
- The user must have added sources/sinks via the CF module
- The user must have inserted all necessary data for sources/sinks via the CF module
- The CF module is run
- The user must have chosen at least one source and one sink
- The user has chosen a simulation that requires the GIS module

3.1.3.6.3 Basic Flow For The User

A graphical scheme of how the user experiences the GIS module is illustrated below:

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

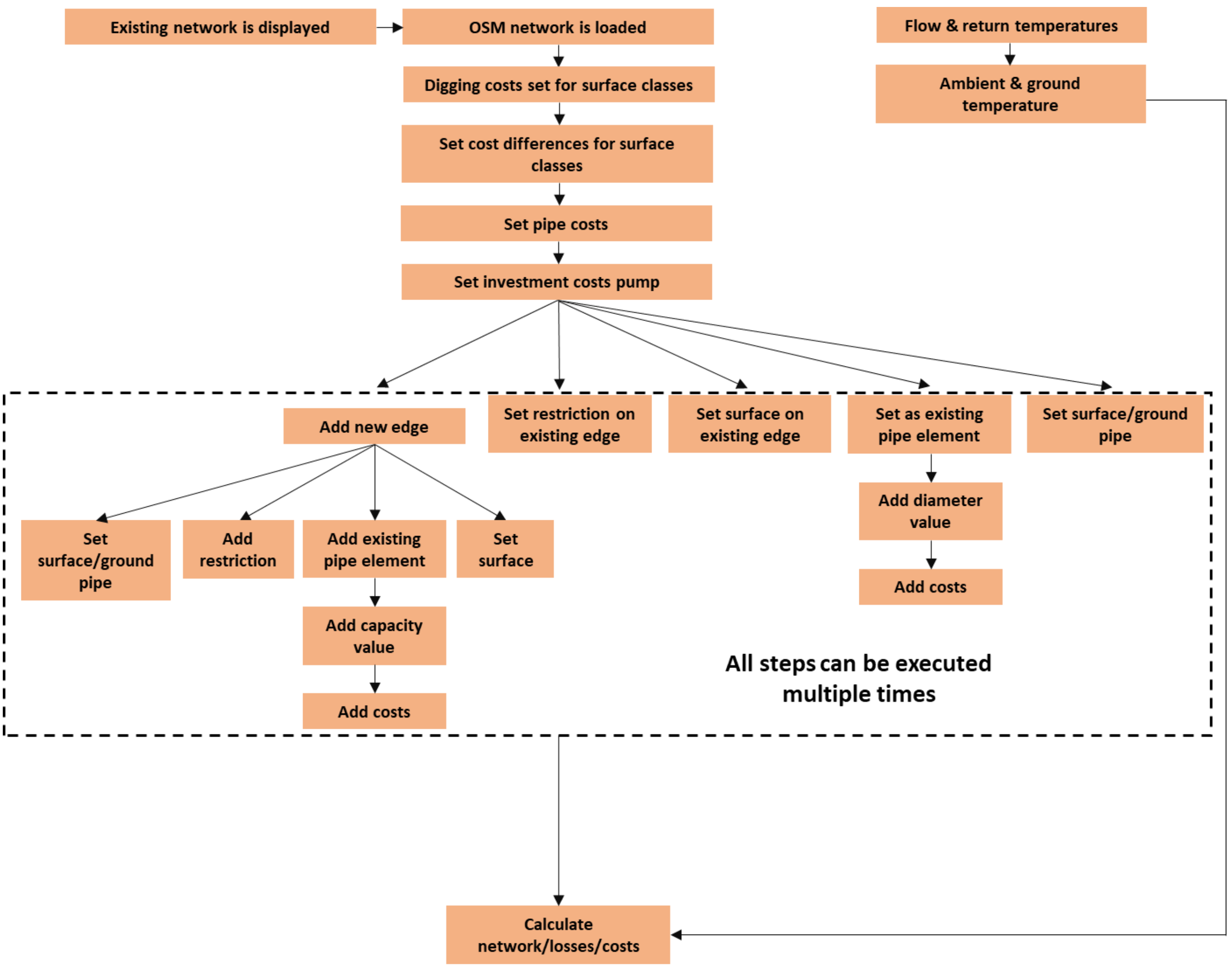


Figure 78: Basic flow for the user.

The more detailed sequence of user activities can be found below; graphical illustrations of the steps described can be found in the Visual Model section.

6. The user enters the Graphical User Interface (GUI) and defines the area where the OSM network should be loaded.
7. The OSM street network is loaded automatically for the user-defined area (Figure 79). The OSM network is automatically connected to the existing grid if added. All sources and sinks are connected to the closest road junction (Figure 80).
8. The inputs for the general settings illustrated in Figure 81 are displayed to the user.
9. The user sets the flow, return, ambient, and ground temperatures. Predefined values are loaded from the KB.
10. The user sets the pipe cost factor difference between street and terrain surface as well as between street and no surface. Predefined values are loaded from the KB.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

11. The user defines the digging costs as fixed and variable for street and terrain surfaces including the formula exponent. Predefined values are loaded from the KB.
12. The user defines the fixed and variable pipe costs including the formula exponent. Predefined values are loaded from the KB.
13. The user may input investment costs for pumps.
14. The user may add a new network element by clicking on the “add network element” button and connecting a new edge to the road network graph (Figure 82).
15. The user may change the surface class of each edge; by default, the surface will be defined as a street (Figure 83).
16. The user may set a restriction on each edge; by default, this value will be set to optional (Figure 84).
17. The user may set a road element to an existing pipe element; by default, this value will be set to No (Figure 85).
18. The user may set a road network as a potential surface pipe pathway (Figure 86).
19. The user exits the GUI.

3.1.3.6.4 Alternate/Exception Flows

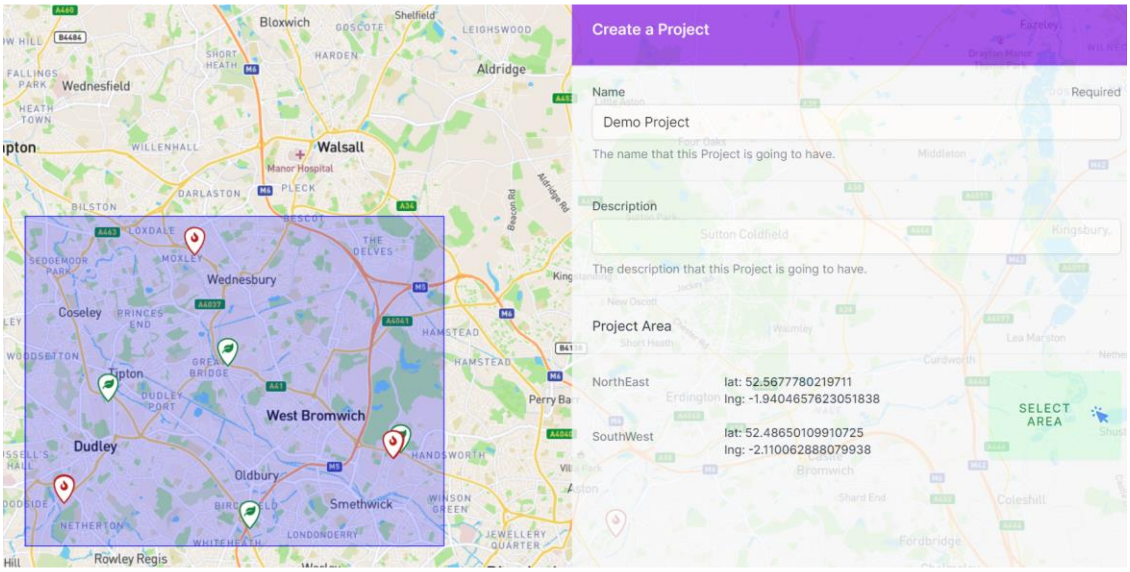
1a – user inputs information about the existing grid
12a – if the user has set a road element to an existing pipe element, the user needs to add a diameter value. Further, the user can add costs for that element if wanted (Figure 85)
13a – if the user has set a road network element to a surface pipe, the digging costs attribute of that edge will be set automatically to None.

3.1.3.6.5 Post Conditions

- The graphical network is stored
- An array with the losses/costs/network length between each source/sink pair is forwarded to the other modules
- The user can check the costs and losses of each element by clicking on the graphical representation of the grid.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

3.1.3.6.6 Visual Model



Create a Project

Name: Demo Project (Required)

The name that this Project is going to have.

Description

The description that this Project is going to have.

Project Area

NorthEast: lat: 52.5677780219711, lng: -1.9404657623051838

SouthWest: lat: 52.48650109910725, lng: -2.110062888079938

Figure 79: User defines the area for loading OSM graph.

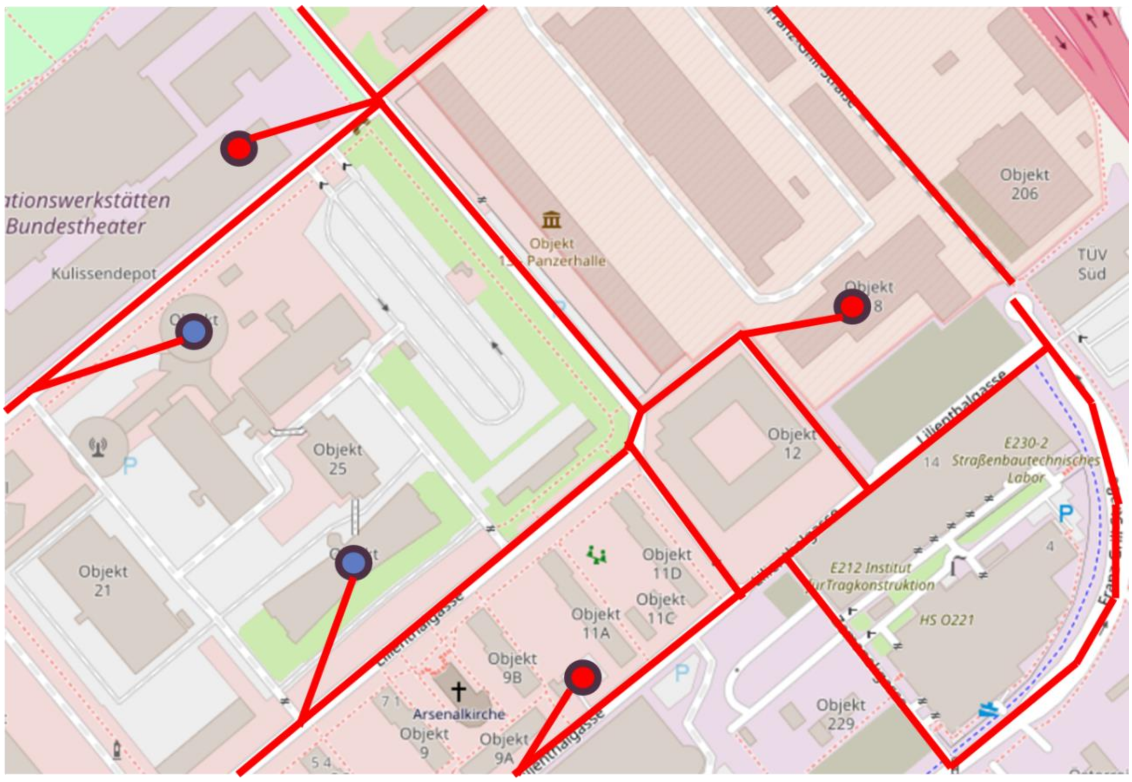
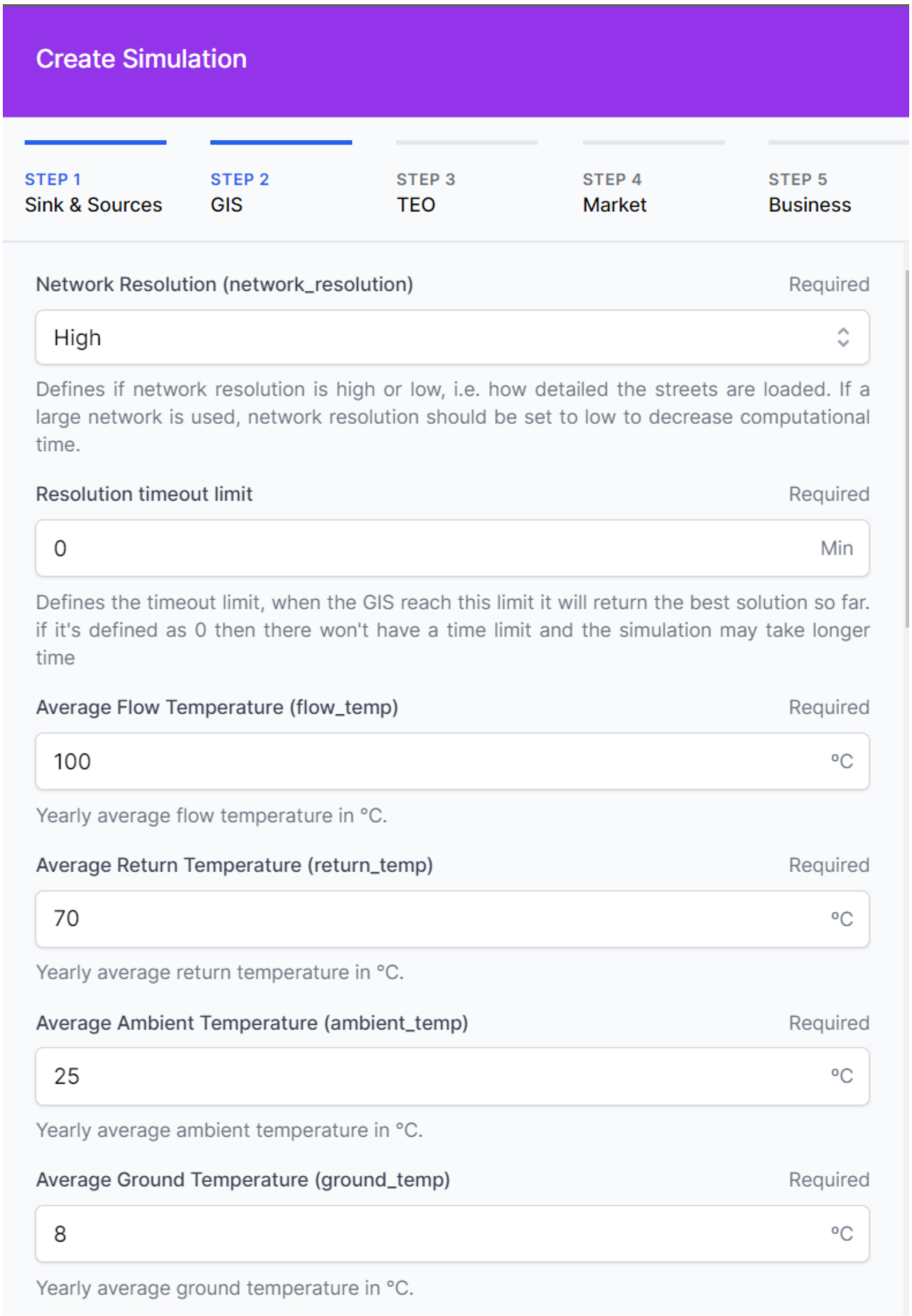


Figure 80: OSM graph is loaded for the project area (mock-up).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121



Create Simulation

Network Resolution (network_resolution) Required

High

Defines if network resolution is high or low, i.e. how detailed the streets are loaded. If a large network is used, network resolution should be set to low to decrease computational time.

Resolution timeout limit Required

0 Min

Defines the timeout limit, when the GIS reach this limit it will return the best solution so far. if it's defined as 0 then there won't have a time limit and the simulation may take longer time

Average Flow Temperature (flow_temp) Required

100 ºC

Yearly average flow temperature in °C.

Average Return Temperature (return_temp) Required

70 ºC

Yearly average return temperature in °C.

Average Ambient Temperature (ambient_temp) Required

25 ºC

Yearly average ambient temperature in °C.

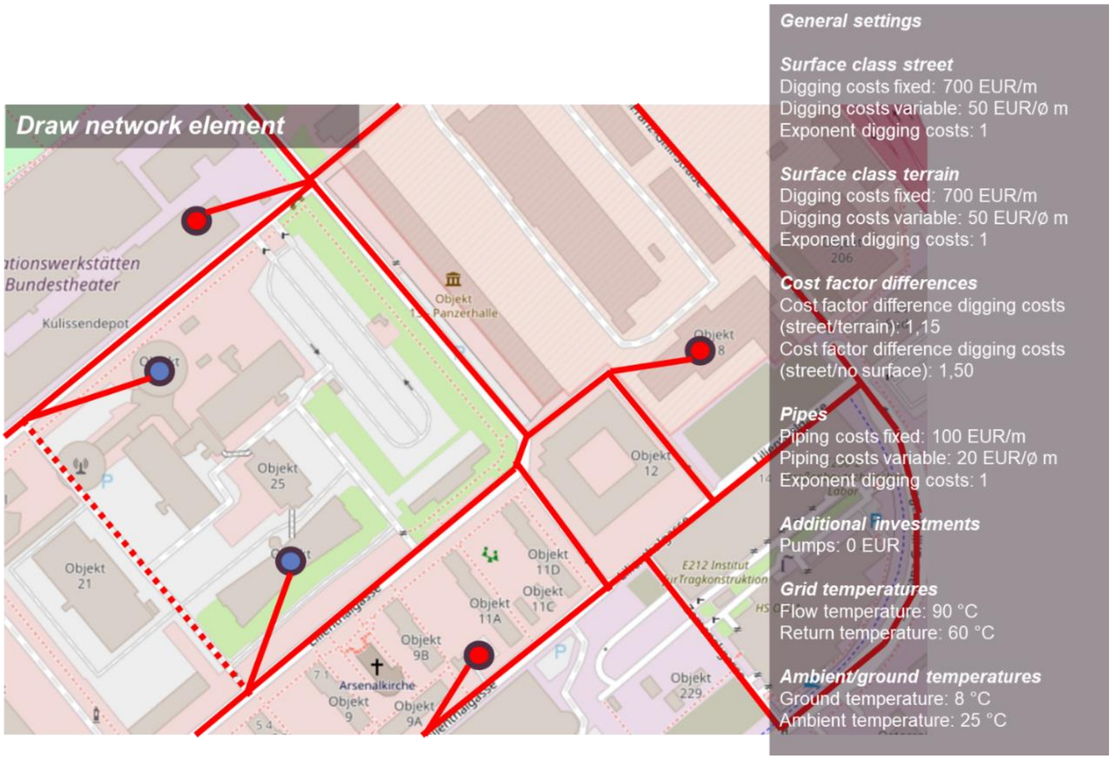
Average Ground Temperature (ground_temp) Required

8 ºC

Yearly average ground temperature in °C.

Figure 81: Input window for GIS module.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121



General settings

Surface class street
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Surface class terrain
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Cost factor differences
Cost factor difference digging costs (street/terrain): 1,15
Cost factor difference digging costs (street/no surface): 1,50

Pipes
Piping costs fixed: 100 EUR/m
Piping costs variable: 20 EUR/ø m
Exponent digging costs: 1

Additional investments
Pumps: 0 EUR

Grid temperatures
Flow temperature: 90 °C
Return temperature: 60 °C

Ambient/ground temperatures
Ground temperature: 8 °C
Ambient temperature: 25 °C

Figure 82: Adding network element to the graph (mock-up).

Element xy

Surface type: street
Existing grid element: No
Overland pipe: No
Restriction: must build

General settings

Surface class street
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Surface class terrain
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Cost factor differences
Cost factor difference digging costs (street/terrain): 1,15
Cost factor difference digging costs (street/no surface): 1,50

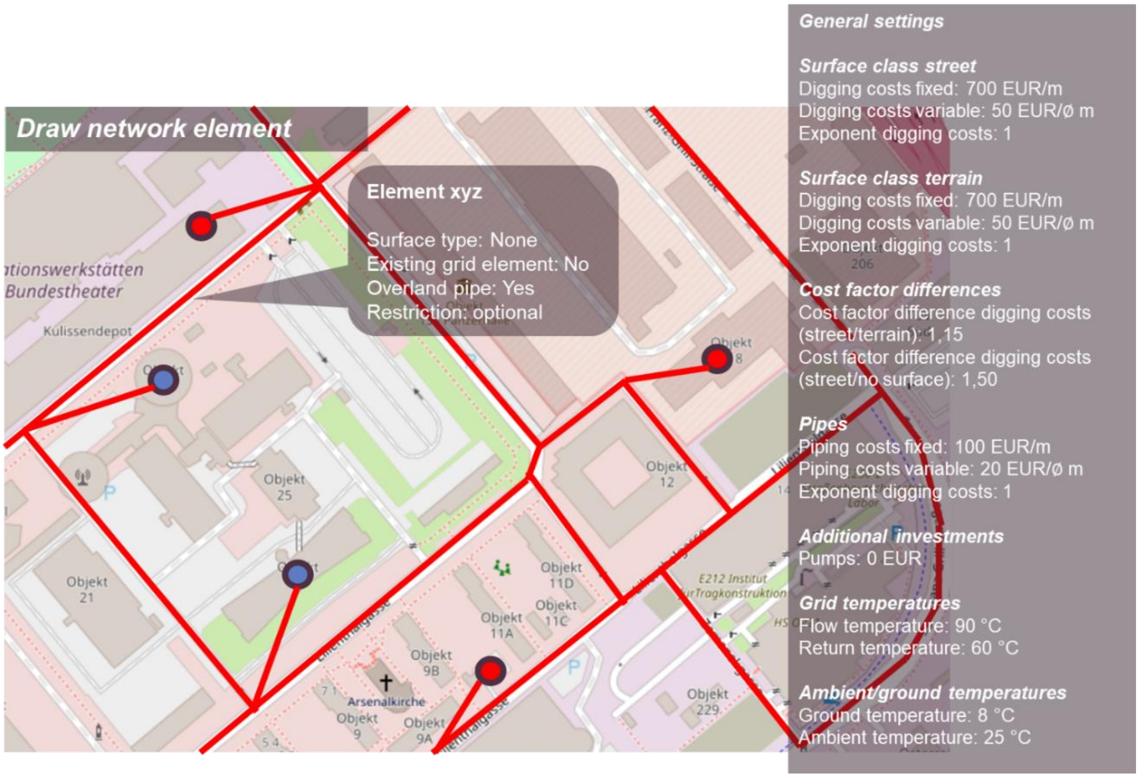
Pipes
Piping costs fixed: 100 EUR/m
Piping costs variable: 20 EUR/ø m
Exponent digging costs: 1

Additional investments
Pumps: 0 EUR

Grid temperatures
Flow temperature: 90 °C
Return temperature: 60 °C

Ambient/ground temperatures
Ground temperature: 8 °C
Ambient temperature: 25 °C

Figure 83: Changing surface type of road element (mock-up).



Element xyz

Surface type: None
Existing grid element: No
Overland pipe: Yes
Restriction: optional

General settings

Surface class street
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Surface class terrain
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Cost factor differences
Cost factor difference digging costs (street/terrain): 1,15
Cost factor difference digging costs (street/no surface): 1,50

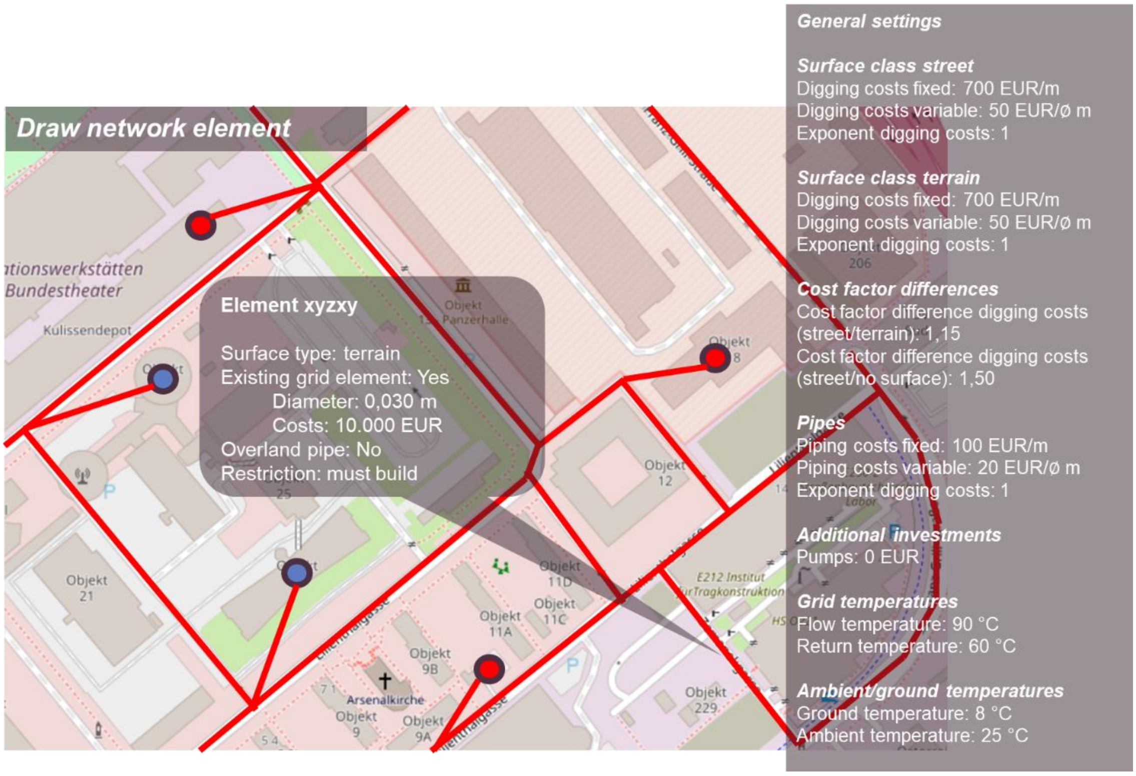
Pipes
Piping costs fixed: 100 EUR/m
Piping costs variable: 20 EUR/ø m
Exponent digging costs: 1

Additional investments
Pumps: 0 EUR

Grid temperatures
Flow temperature: 90 °C
Return temperature: 60 °C

Ambient/ground temperatures
Ground temperature: 8 °C
Ambient temperature: 25 °C

Figure 84: Setting restrictions on the element (mock-up).



Element xyzxy

Surface type: terrain
Existing grid element: Yes
Diameter: 0,030 m
Costs: 10.000 EUR
Overland pipe: No
Restriction: must build

General settings

Surface class street
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Surface class terrain
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Cost factor differences
Cost factor difference digging costs (street/terrain): 1,15
Cost factor difference digging costs (street/no surface): 1,50

Pipes
Piping costs fixed: 100 EUR/m
Piping costs variable: 20 EUR/ø m
Exponent digging costs: 1

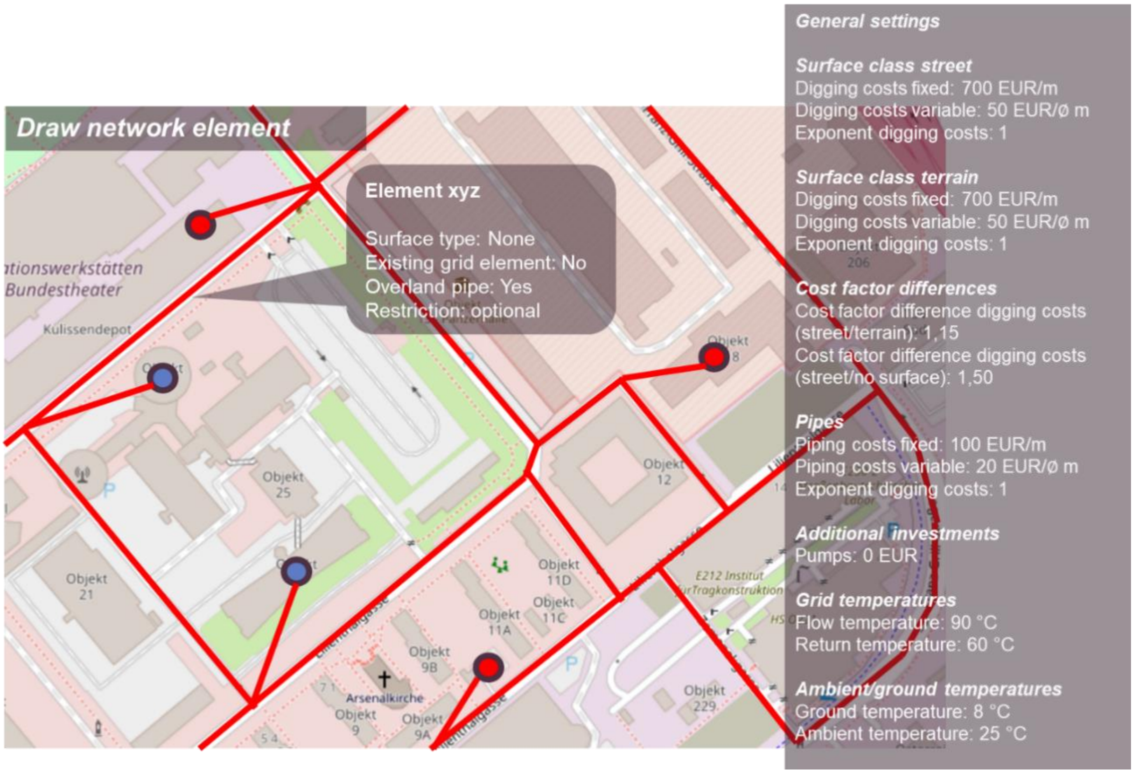
Additional investments
Pumps: 0 EUR

Grid temperatures
Flow temperature: 90 °C
Return temperature: 60 °C

Ambient/ground temperatures
Ground temperature: 8 °C
Ambient temperature: 25 °C

Figure 85: Adding existing network element (mock-up).

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121



Element xyz

Surface type: None
Existing grid element: No
Overland pipe: Yes
Restriction: optional

General settings

Surface class street
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Surface class terrain
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Cost factor differences
Cost factor difference digging costs (street/terrain): 1,15
Cost factor difference digging costs (street/no surface): 1,50

Pipes
Piping costs fixed: 100 EUR/m
Piping costs variable: 20 EUR/ø m
Exponent digging costs: 1

Additional investments
Pumps: 0 EUR

Grid temperatures
Flow temperature: 90 °C
Return temperature: 60 °C

Ambient/ground temperatures
Ground temperature: 8 °C
Ambient temperature: 25 °C

Figure 86: Setting overland pathway for pipes (mock-up).

3.1.3.7 Integration and Interactions with other Modules

The calculations of the GIS module are inputs for the other calculation modules. It also gets inputs from other modules.

The GIS module's main dependency is on the CF module for the information on sources and sinks. The CF module provides IDs, locations (latitude, longitude), and maximum capacities of sources and sinks. The GIS module provides the grid losses between each source and sink pairs as well as total network losses to the CF module.

The connection between the TEO and the GIS modules is of particular interest. Interdependencies between these modules require an iteration among them. The TEO bases its calculation on losses/costs from the GIS. The GIS calculates its losses and costs based on the maximum exchange capacities due to energy flows and sources/sinks to be integrated. Thus, iteration loops between the modules are implemented. These loops shall ensure an adequate accuracy of the results.

The user uses the GIS module to estimate a potential network solution for a given set of sources and sinks. Heat/cold losses and investment costs are calculated for this specific network solution. The TEO then tries to optimize the actual thermal flows between all sources and sinks and may decide to exclude sources or sinks from the network due to, e.g., a mismatch of supply or demand profiles. This potential exclusion of sources or sinks may lead to the necessity of recalculating the network. In addition, the maximum exchange capacities between all sources and sinks are determined within the TEO and processed within the GIS in order to increase the network costs and loss calculation accuracy.

Iterations between the GIS and the TEO modules will gradually increase the result accuracy but may be very resource-intensive, as multiple iterations may be required. Especially the exact exchange capacities between all sources and sinks may vastly affect the pipe infrastructure investments, as the network capacities will be shaped according to the matching energy flows.

The module can be used in two different ways, firstly as a standalone module leading to more simple results and secondly in cooperation with the TEO module yielding more precise results due to the consideration of economically interesting heat flows between sources and sinks. Flows of these two calculation methods are illustrated in Figure 87. The way the GIS module is used depends on the simulation type the user has chosen.

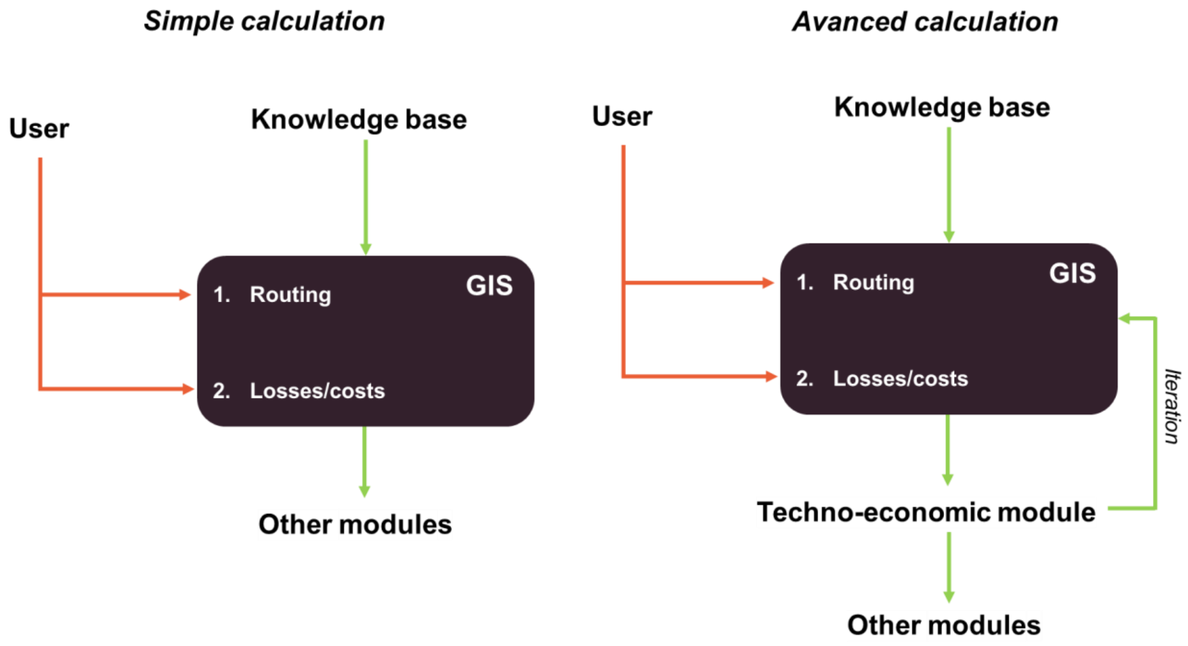


Figure 87: Visualization of simple and advanced calculations.

Within the simple calculation**,** the module gets inputs (e.g., location and capacities) of all sink and source points to integrate into the network. Along with different calculation steps, the user can change other default inputs (e.g., costs or technical parameters) and may set certain costs and restrictions for network elements. Based on the characteristics of all sources, sinks, and potential routes to connect them to a network, its' related pipe dimensions, heat/cold losses, and the related costs are calculated.
The results of this calculation are then available for other modules. However, the techno-economic optimization module calculates the technically and economically optimal matching between sources and sinks. Therefore, this simple calculation does not account for optimized energy exchange between sources and sinks.
The advanced calculation includes a feedback loop with the TEO module. The user chooses the source and sink points to be integrated (obtained from the CF module) and may set certain costs and restrictions for existing network elements or adds new ones. The GIS module then identifies a network route and calculates related losses and costs. Based on that preliminary losses and costs (no matching has been conducted yet), the TEO module performs a matching process optimizing the exchange of heat/cold between the different sources and sinks towards predefined economic targets. This may exclude mismatching sources and sinks from the network, and a new network solution must be calculated. The TEO module thus feeds back the information about which sources and sinks to include and their maximum exchange capacities. Based on this information, a recalculation of the precise network, losses, and costs is initiated. The advanced calculation is elaborated in more detail below.

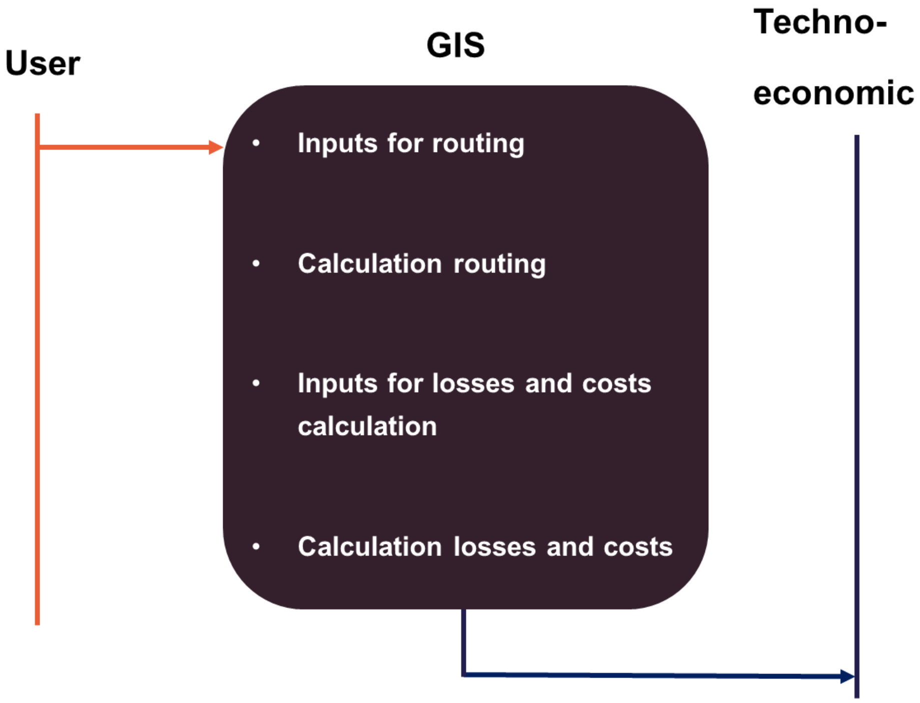


Figure 88: First step of advanced calculation with the GIS module

As shown in Figure 88, within the first part of the advanced calculation, the user sets all inputs for the routing. Based on these, the routing and the loss/cost calculation are conducted. Thereby, simplified costs and losses are calculated. The TEO module needs these to perform the techno-economic optimization of matching sources and sinks.

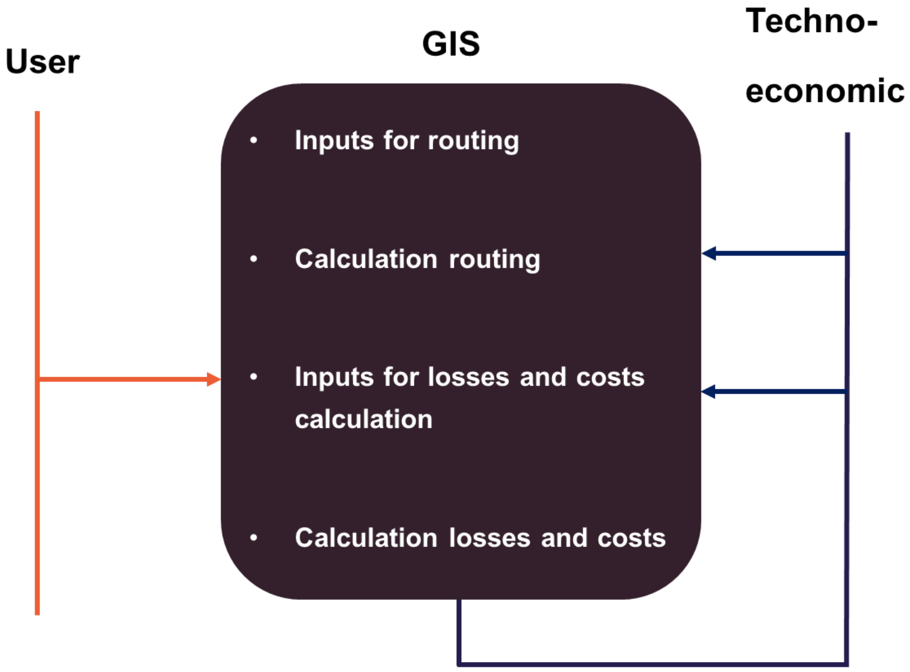


Figure 89: Second step of advanced calculation with the GIS module

The optimization may conclude that the integration of specific sources and sinks into the network may not be technically or economically feasible. Information about sources and sinks to be integrated and exchange capacities are returned to the GIS module. If needed, the GIS module recalculates a new network solution due to specific sinks not

being integrated into the results of the first run of the TEO module. Costs and losses are calculated based on matching capacities. These more accurate results are then forwarded again to the TEO module and in succession to the other modules.
GIS modules' interaction with other modules is providing outputs. BM receives total network cost from the GIS. MM receives the list of agents (sources and sinks) existing the network solution from the GIS.

As of month 38, GIS module is fully integrated with all the modules and has all features required by the use cases. The advanced features that are not a priority for the EMB3Rs case studies are developed within the standalone GIS module and will be integrated into the platform in later stages. These features are:

- considering a detailed existing grid network,
- using surface and pipe classes,
- forcing or restricting use of roads.

3.1.4 Reports

3.1.4.1 Contribution to the Main Simulation Report

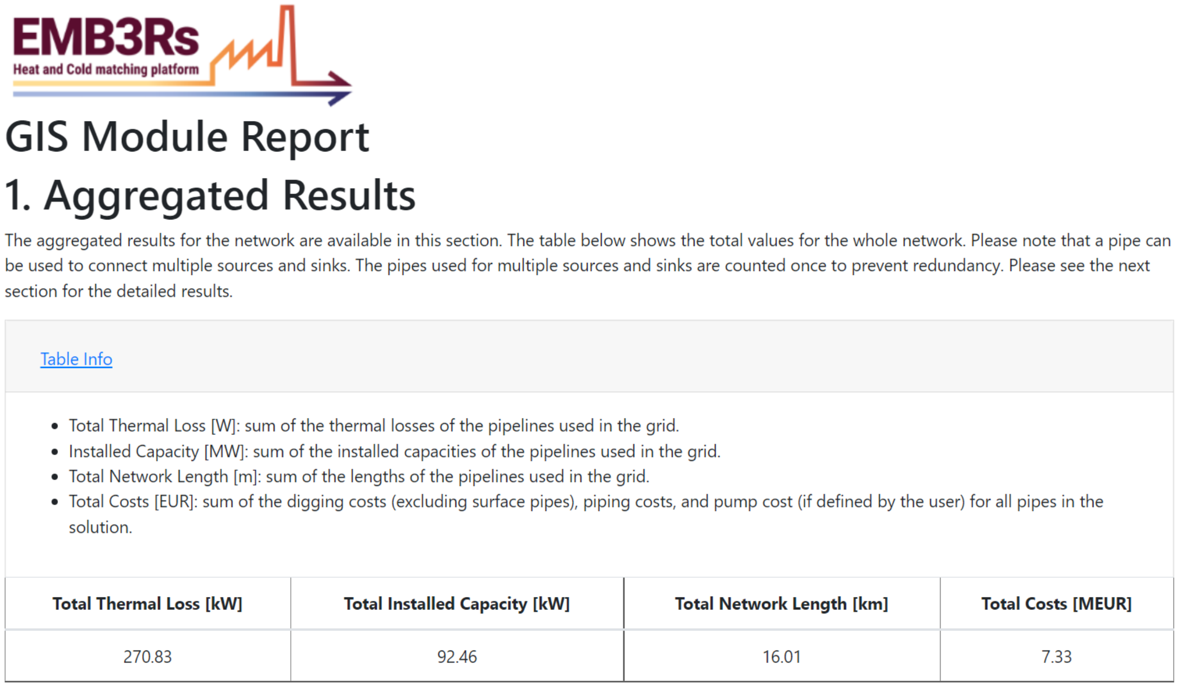
The GIS module's contribution to the main simulation report is the total network length, total network cost and the total network losses. The individual values for each pipeline is provided in the detailed report. While calculating the total values, the pipes that are used for multiple sources and sinks are counted once to prevent redundancy.

Table 108: GIS Module's contribution to the main simulation report.

Variable Name	Description	Unit
Total Network Length	Sum of the lengths of the pipelines used in the grid.	m
Total Network Cost	Sum of the digging costs (excluding surface pipes), piping costs, and pump cost (if defined by the user) for all pipes in the solution.	EUR
Total Network Losses	Sum of the thermal losses of the pipelines used in the grid.	W

A sample screenshot from GIS module's aggregated simulation report is given below:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121



GIS Module Report

1. Aggregated Results

The aggregated results for the network are available in this section. The table below shows the total values for the whole network. Please note that a pipe can be used to connect multiple sources and sinks. The pipes used for multiple sources and sinks are counted once to prevent redundancy. Please see the next section for the detailed results.

Table Info

- Total Thermal Loss [W]: sum of the thermal losses of the pipelines used in the grid.
- Installed Capacity [MW]: sum of the installed capacities of the pipelines used in the grid.
- Total Network Length [m]: sum of the lengths of the pipelines used in the grid.
- Total Costs [EUR]: sum of the digging costs (excluding surface pipes), piping costs, and pump cost (if defined by the user) for all pipes in the solution.

Total Thermal Loss [kW]	Total Installed Capacity [kW]	Total Network Length [km]	Total Costs [MEUR]
270.83	92.46	16.01	7.33

Figure 90: Screenshot of GIS module's main simulation report.

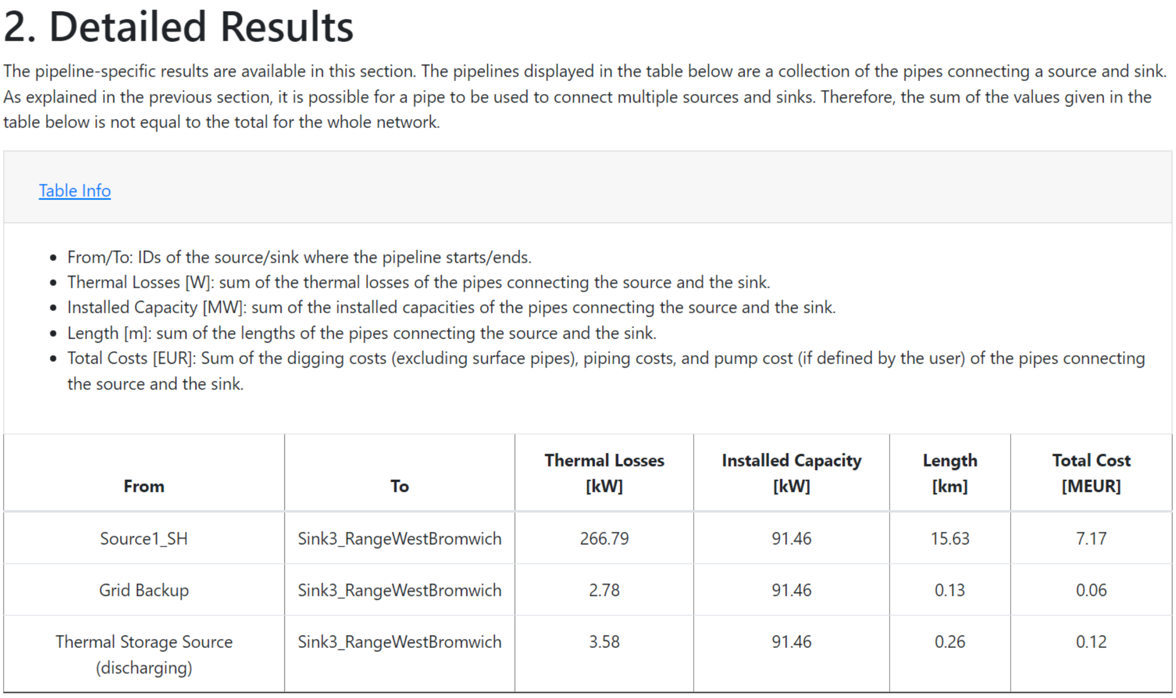
3.1.4.2 Contribution to the Detailed Simulation Report

The GIS module's contribution to the detailed simulation report is the pipeline-specific length, pipe-specific cost, pipe-specific installed capacity, and the pipe-specific thermal losses. The pipelines displayed in the report are a collection of the pipes connecting a source and sink.

Table 109: GIS Module's contribution to the detailed simulation report.

Variable Name	Description	Unit
Pipe Length	Sum of the lengths of the pipelines connecting the source and the sink.	m
Total Cost of the Pipeline	Sum of the digging costs (excluding surface pipes), piping costs, and pump cost (if defined by the user) of the pipes connecting the source and the sink.	EUR
Total Thermal Loss of the Pipeline	Sum of the thermal losses of the pipes connecting the source and the sink.	W
Total Installed Capacity of the Pipeline	Sum of the installed capacities of the pipes connecting the source and the sink.	MW
From	Name of the source/sink where the pipeline starts	-
To	Name of the source/sink where the pipeline ends	-

A sample screenshot from GIS module's detailed simulation report is given below:



2. Detailed Results

The pipeline-specific results are available in this section. The pipelines displayed in the table below are a collection of the pipes connecting a source and sink. As explained in the previous section, it is possible for a pipe to be used to connect multiple sources and sinks. Therefore, the sum of the values given in the table below is not equal to the total for the whole network.

Table Info

- From/To: IDs of the source/sink where the pipeline starts/ends.
- Thermal Losses [W]: sum of the thermal losses of the pipes connecting the source and the sink.
- Installed Capacity [MW]: sum of the installed capacities of the pipes connecting the source and the sink.
- Length [m]: sum of the lengths of the pipes connecting the source and the sink.
- Total Costs [EUR]: Sum of the digging costs (excluding surface pipes), piping costs, and pump cost (if defined by the user) of the pipes connecting the source and the sink.

From	To	Thermal Losses [kW]	Installed Capacity [kW]	Length [km]	Total Cost [MEUR]
Source1_SH	Sink3_RangeWestBromwich	266.79	91.46	15.63	7.17
Grid Backup	Sink3_RangeWestBromwich	2.78	91.46	0.13	0.06
Thermal Storage Source (discharging)	Sink3_RangeWestBromwich	3.58	91.46	0.26	0.12

Figure 91: Screenshot of GIS module's detailed simulation report.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

3.2 User Manual

3.2.1 Introduction

The purpose of the Geographical Information System (GIS) module is to analyze the spatial dimension of the EMB3Rs platform. The GIS module conducts several calculations represented by the following main features

- district heating and cooling (DHC) network calculation based on different heat/cold sources and sinks to be included
- calculation of the thermal losses and investment costs of the resulting DHC network (grid).

The main outputs of the GIS module are:

- DHC network where all the possible sources and sinks are connected
- investment cost of the calculated DHC network
- thermal losses of the calculated DHC grid.

3.2.2 Inputs and Outputs

The general input and output structure of the GIS module is given in Figure 92. The inputs that are expected from the user, their labels, and descriptions of the inputs are given in Table 110. Under the “Function” column, it is indicated which function is using the input. It is also indicated if the input is mandatory or not. Please note that all mandatory inputs except for “Project Area” have a default value stored in the Knowledge Base. In other words, if the user does not have enough information to set a value for those variables or basically wants to use default variables, he/she has the option not to give input. However, the user must provide the “Project Area” input by choosing it via the platform.

Also, note that the unit digging and piping costs are calculated in the following format:

$$\text{Unit Digging/Piping Costs}\left[\frac{EUR}{m}\right] = \text{fixed cost} + [(\text{diameter})(\text{variable cost})]^{\text{exponent}}.$$

Therefore, all the inputs named as a fixed cost in Table 110 correspond to the fixed cost in the formula above. Similarly, inputs named as a variable cost correspond to the variable cost in the formula above. Finally, the inputs named as the exponent correspond to the exponent in the formula above. The model calculates the diameter, so it is not user input. If a pipe is an overland pipe, the model automatically assigns a digging cost of zero to it.

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121

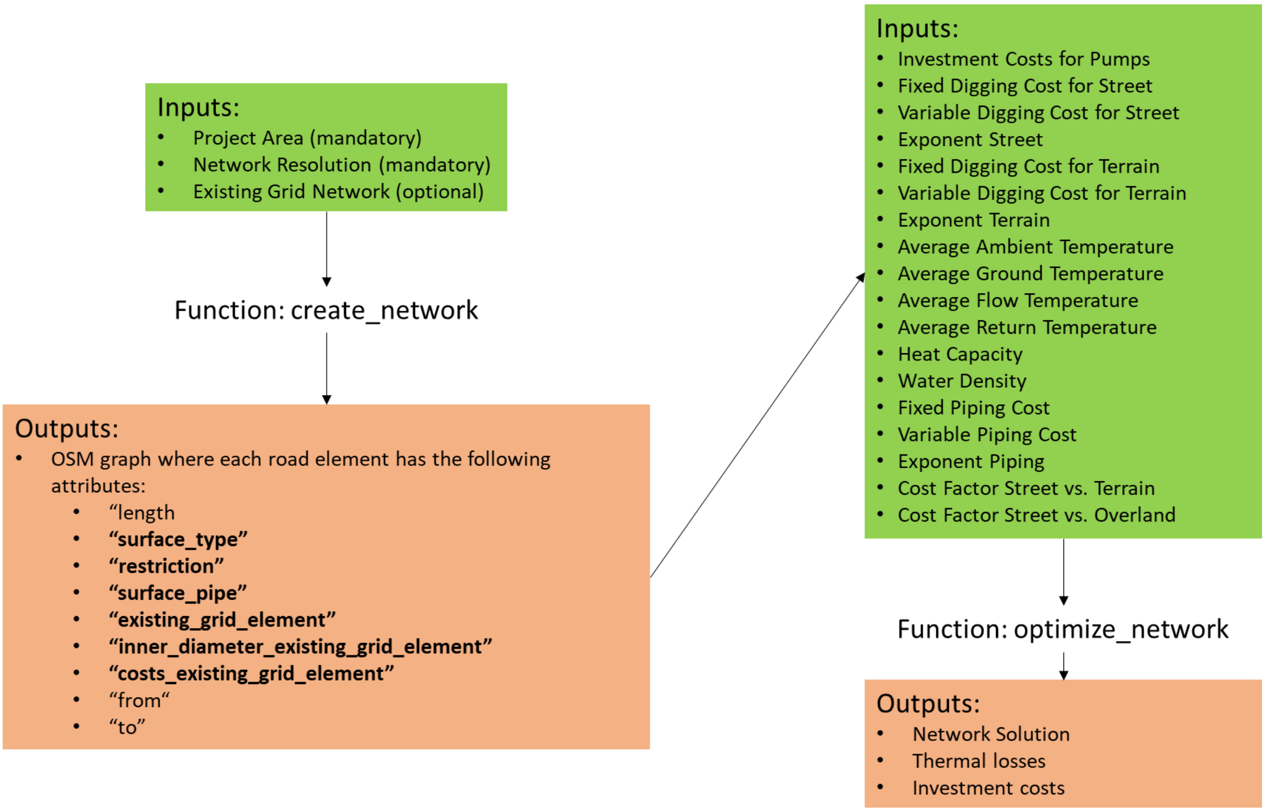


Figure 92: General input-output structure of the GIS module.

Table 110: Inputs of the GIS Module.

Function	**Mandatory**	**Label**	**Description**	**Unit**
create_network	TRUE	Network Resolution	Defines if network resolution is high or low, i.e., how detailed the streets are loaded. If a large network is used, network resolution should be set to low to decrease computational time. Set to high by default.	-
create_network	FALSE	Existing Grid Network	The information on the existing grid network. For each pipe, IDs of sources/sinks connected by the pipe, latitudes, and longitudes of those sources/sinks, diameter and length of the pipe, total cost of the pipe, and if the respective pipe is a surface pipe should be defined.	Diameter in m. Length in m. Total cost of the pipe in EUR.
create_network	TRUE	Project Area	The area that will be considered for the grid. User could specify the area by drawing a rectangular shape on the map via platform.	-
optimize_network	FALSE	Investment Costs for Pumps	Investment costs for pumps. Set to 0 by default.	EUR
optimize_network	TRUE	Fixed Digging Cost for Street	Fixed digging cost for streets. Set to 350 by default.	EUR/m
optimize_network	TRUE	Variable Digging Cost for Street	Variable digging cost for streets. Set to 700 by default.	EUR/m²

optimize_network	TRUE	Exponent Street	The exponent of the digging cost for the street. Set to 1.1 by default.	-
optimize_network	TRUE	Fixed Digging Cost for Terrain	Fixed digging cost for terrains. Set to 200 by default.	EUR/m
optimize_network	TRUE	Variable Digging Cost for Terrain	Variable digging cost for terrains. Set to 500 by default.	EUR/m^2
optimize_network	TRUE	Exponent Terrain	The exponent of the digging cost for the terrain. Set to 1.1 by default.	-
optimize_network	TRUE	Average Ambient Temperature	Yearly average ambient temperature. Set to 25 by default.	°C
optimize_network	TRUE	Average Ground Temperature	Yearly average ground temperature. Set to 8 by default.	°C
optimize_network	TRUE	Average Flow Temperature	Yearly average flow temperature. Set to 100 by default.	°C
optimize_network	TRUE	Average Return Temperature	Yearly average return temperature. Set to 70 by default.	°C
optimize_network	TRUE	Heat Capacity	Heat capacity at a specific temperature (average of flow and return temperatures). Set to 4.18 by default.	J/kgK
optimize_network	TRUE	Water Density	Water density at a specific temperature (average of flow and return temperatures). Set to 1000 by default.	kg/m3
optimize_network	TRUE	Fixed Piping Cost	The fixed component of the piping cost. Set to 50 by default.	EUR/m
optimize_network	TRUE	Variable Piping Cost	The fixed component of the piping cost. Set to 700 by default.	EUR/m^2
optimize_network	TRUE	Exponent Piping	The exponent of the piping cost. Set to 1.3 by default.	-
optimize_network	TRUE	Cost Factor Street vs. Terrain	Determines how much cheaper it is to lay 1 m of pipe into a terrain than a street. Expressed in decimals: 0.1 means it is 10% cheaper.	Decimals
optimize_network	TRUE	Cost Factor Street vs. Overland	Determines how much cheaper it is to place 1 m of the pipe over the ground than putting it into the street. Expressed in decimals: 0.4 means it is 40% cheaper.	Decimals

Function	**Mandatory**	**Label**	**Description**	**Unit**
create_network	TRUE	Network Resolution	Defines if network resolution is high or low, i.e., how detailed the streets are loaded. If a large network is used, network resolution should be set to low to decrease computational time. Set to high by default.	-
create_network	FALSE	Existing Grid Network	The information on the existing grid network. For each pipe, IDs of sources/sinks connected by the pipe, latitudes, and longitudes of those	Diameter in m. Length in m. Total cost of the pipe in EUR.

			sources/sinks, diameter and length of the pipe, total cost of the pipe, and if the respective pipe is a surface pipe should be defined.	
create_network	TRUE	Project Area	The area that will be considered for the grid. User could specify the area by drawing a rectangular shape on the map via platform.	-
optimize_network	FALSE	Investment Costs for Pumps	Investment costs for pumps. Set to 0 by default.	EUR
optimize_network	TRUE	Fixed Digging Cost for Street	Fixed digging cost for streets. Set to 350 by default.	EUR/m
optimize_network	TRUE	Variable Digging Cost for Street	Variable digging cost for streets. Set to 700 by default.	EUR/m
optimize_network	TRUE	Exponent Street	The exponent of the digging cost for the street. Set to 1.1 by default.	-
optimize_network	TRUE	Fixed Digging Cost for Terrain	Fixed digging cost for terrains. Set to 200 by default.	EUR/m
optimize_network	TRUE	Variable Digging Cost for Terrain	Variable digging cost for terrains. Set to 500 by default.	EUR/m
optimize_network	TRUE	Exponent Terrain	The exponent of the digging cost for the terrain. Set to 1.1 by default.	-
optimize_network	TRUE	Average Ambient Temperature	Yearly average ambient temperature. Set to 25 by default.	°C
optimize_network	TRUE	Average Ground Temperature	Yearly average ground temperature. Set to 8 by default.	°C
optimize_network	TRUE	Average Flow Temperature	Yearly average flow temperature. Set to 100 by default.	°C
optimize_network	TRUE	Average Return Temperature	Yearly average return temperature. Set to 70 by default.	°C
optimize_network	TRUE	Heat Capacity	Heat capacity at a specific temperature (average of flow and return temperatures). Set to 4.18 by default.	J/kgK
optimize_network	TRUE	Water Density	Water density at a specific temperature (average of flow and return temperatures). Set to 1000 by default.	kg/m3
optimize_network	TRUE	Fixed Piping Cost	The fixed component of the piping cost. Set to 50 by default.	EUR/m
optimize_network	TRUE	Variable Piping Cost	The fixed component of the piping cost. Set to 700 by default.	EUR/m
optimize_network	TRUE	Exponent Piping	The exponent of the piping cost. Set to 1.3 by default.	-
optimize_network	TRUE	Cost Factor Street vs. Terrain	Determines how much cheaper it is to lay 1 m of pipe into a terrain than a street. Expressed in decimals: 0.1 means it is 10% cheaper.	Decimals
optimize_network	TRUE	Cost Factor Street vs. Overland	Determines how much cheaper it is to place 1 m of the pipe over the ground than putting it into the	Decimals

			street. Expressed in decimals: 0.4 means it is 40% cheaper.	

The outputs of the GIS Module are

- the network solution visualized on Open Street Map,
- the network losses,
- the investment costs.

The GIS module also outputs the potential grid area independent of the network solution. This potential grid area shows all the possible routes for the pipes on OSM.

3.2.3 Simulation

3.2.3.1 Actors

- Platform User
- GIS module
- Knowledge Base
- CF Module
- TEO Module

3.2.3.2 Pre-conditions

- The user must be logged into the platform
- The user must have defined a project area
- The user must have added sources/sinks via the CF module
- The user must have inserted all necessary data for sources/sinks via the CF module
- CF module is run
- The user must have chosen at least one source and one sink
- The user has chosen a simulation that requires the GIS module

3.2.3.3 Basic Flow for the user

A graphical scheme of how the user experiences the GIS module is illustrated below:

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

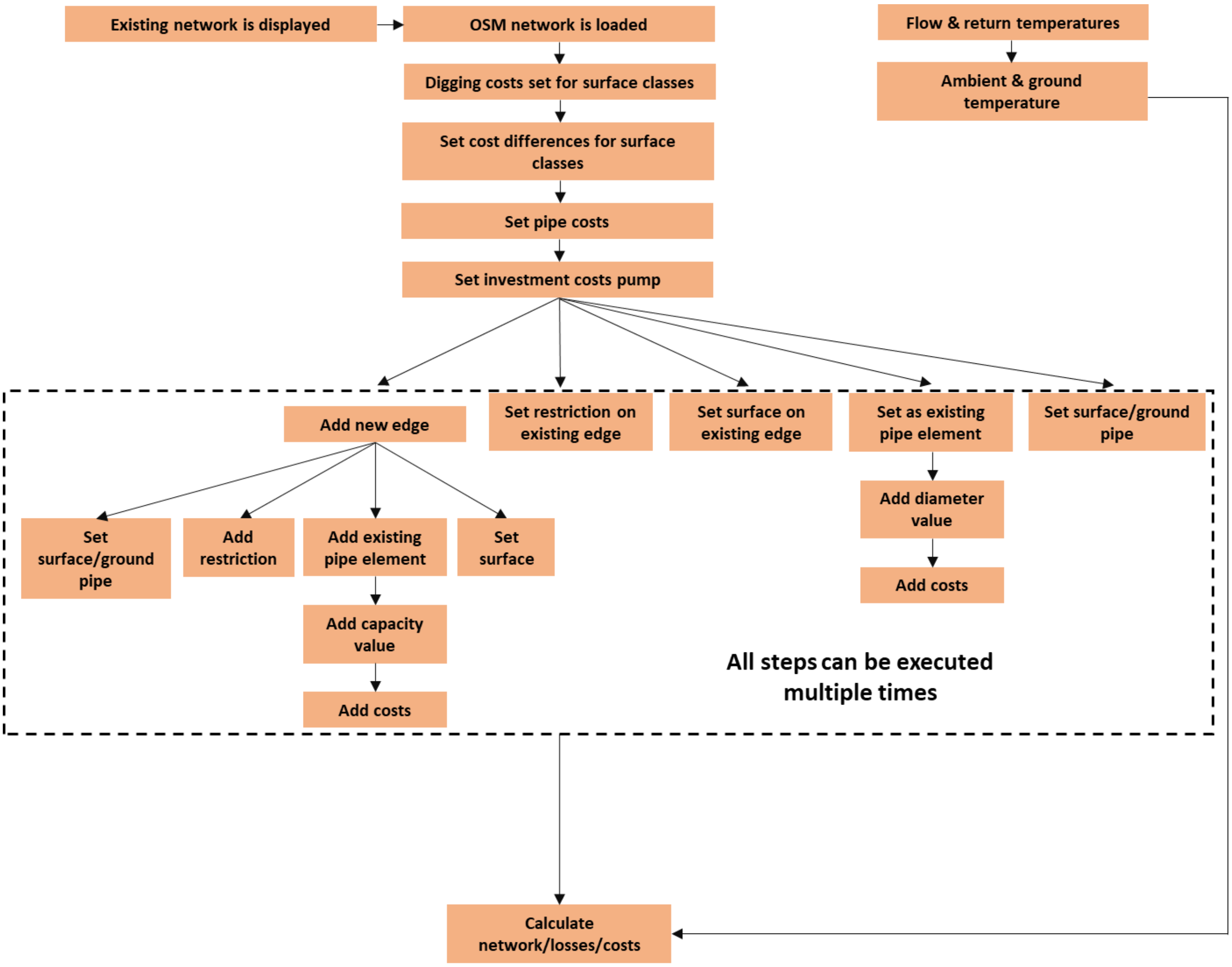


Figure 93: Basic flow for the user.

The more detailed sequence of user activities can be found below; graphical illustrations of the steps are described in the Visual Model section.

1. The user enters the GUI and defines the area where the OSM network should be loaded.
2. The OSM street network is loaded automatically for the user-defined area (Figure 94). The OSM network is automatically connected to the existing grid if added. All sources and sinks are connected to the closest road junction (Figure 95).
3. The inputs for the general settings illustrated in Figure 96 are displayed to the user.
4. The user sets the flow, return, ambient, and ground temperatures. Predefined values are loaded from the KB.
5. The user sets the pipe cost factor difference between street and terrain surface as well as between street and no surface. Predefined values are loaded from the KB.

6. The user defines the digging costs as fixed and variable for street and terrain surfaces, including the formula exponent. Predefined values are loaded from the KB if the user does not specify the parameters.
7. The user defines the fixed and variable pipe costs, including the formula exponent. Predefined values are loaded from the KB if the user does not specify the parameters.
8. The user may input investment costs for pumps.
9. The user may add a new network element by clicking on the "add network element" button and connecting a new edge to the road network graph (Figure 97).
10. The user may change the surface class of each edge; by default, the surface will be defined as a street (Figure 98).
11. The user may set a restriction on each edge; by default, this value will be set to optional (Figure 99).
12. The user may set a road element to an existing pipe element; by default, this value will be set to No (Figure 100).
13. The user may set a road network as a potential surface pipe pathway (Figure 101).
14. The user exits the GUI.

3.2.3.4 Alternate/Exception Flows

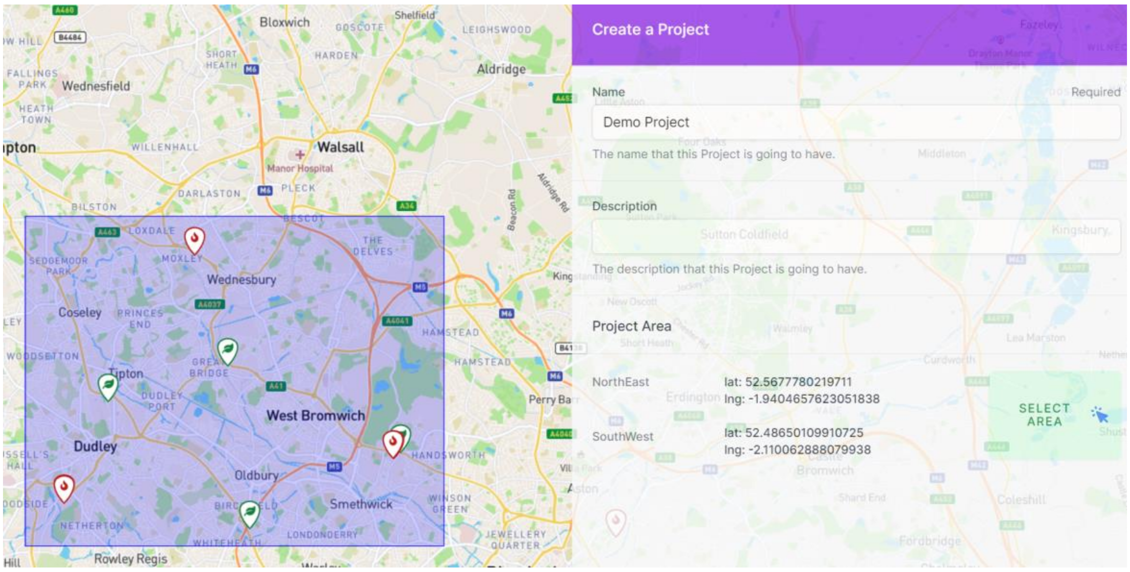
1a – user inputs information about the existing grid
12a – if the user has set a road element to an existing pipe element, the user needs to add a diameter value. Further, the user can add costs for that element if wanted (Figure 100)
13a – if the user has set a road network element to a surface pipe, the digging costs attribute of that edge will be set automatically to None.

3.2.3.4.1 Post Conditions

- The graphical network is stored
- An array with the losses/costs/network length between each source/sink pair is forwarded to the other modules
- The user can check the costs and losses of each element by clicking on the graphical representation of the grid

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121

3.2.3.5 Visual Model



Create a Project

Name — Required

Demo Project

The name that this Project is going to have.

Description

The description that this Project is going to have.

Project Area

NorthEast — lat: 52.5677780219711, lng: -1.9404657623051838

SouthWest — lat: 52.48650109910725, lng: -2.110062888079938

Figure 94: User defines the area for loading OSM graph.

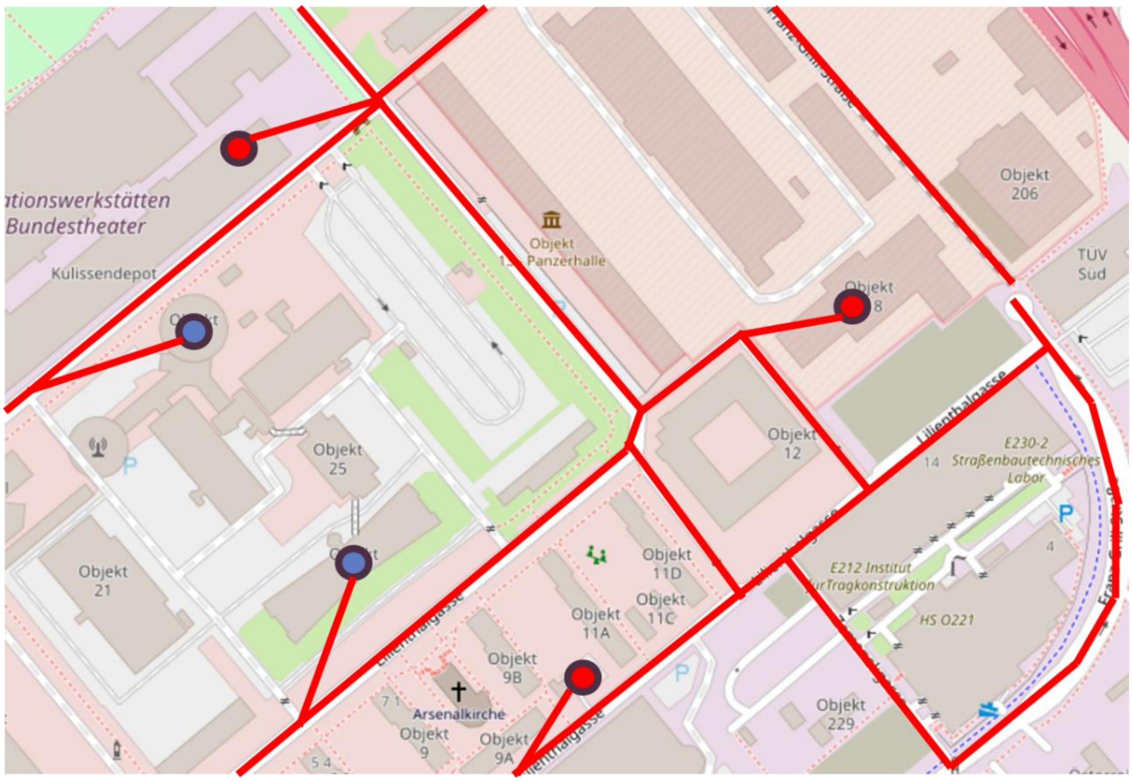
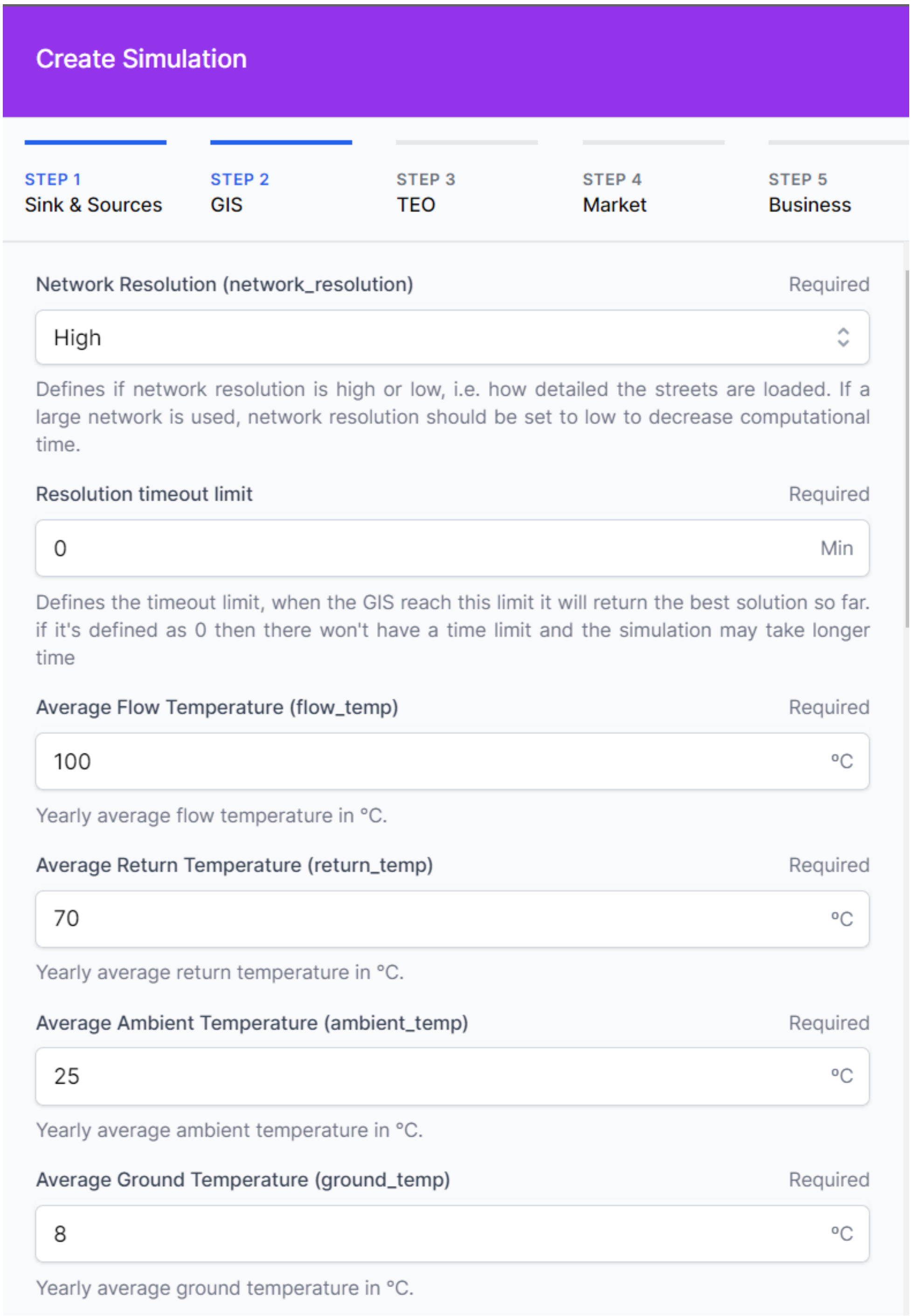


Figure 95: OSM graph is loaded for the project area (mock-up).

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121



Network Resolution (network_resolution) Required

High

Defines if network resolution is high or low, i.e. how detailed the streets are loaded. If a large network is used, network resolution should be set to low to decrease computational time.

Resolution timeout limit Required

0 Min

Defines the timeout limit, when the GIS reach this limit it will return the best solution so far. if it's defined as 0 then there won't have a time limit and the simulation may take longer time

Average Flow Temperature (flow_temp) Required

100 ºC

Yearly average flow temperature in °C.

Average Return Temperature (return_temp) Required

70 ºC

Yearly average return temperature in °C.

Average Ambient Temperature (ambient_temp) Required

25 ºC

Yearly average ambient temperature in °C.

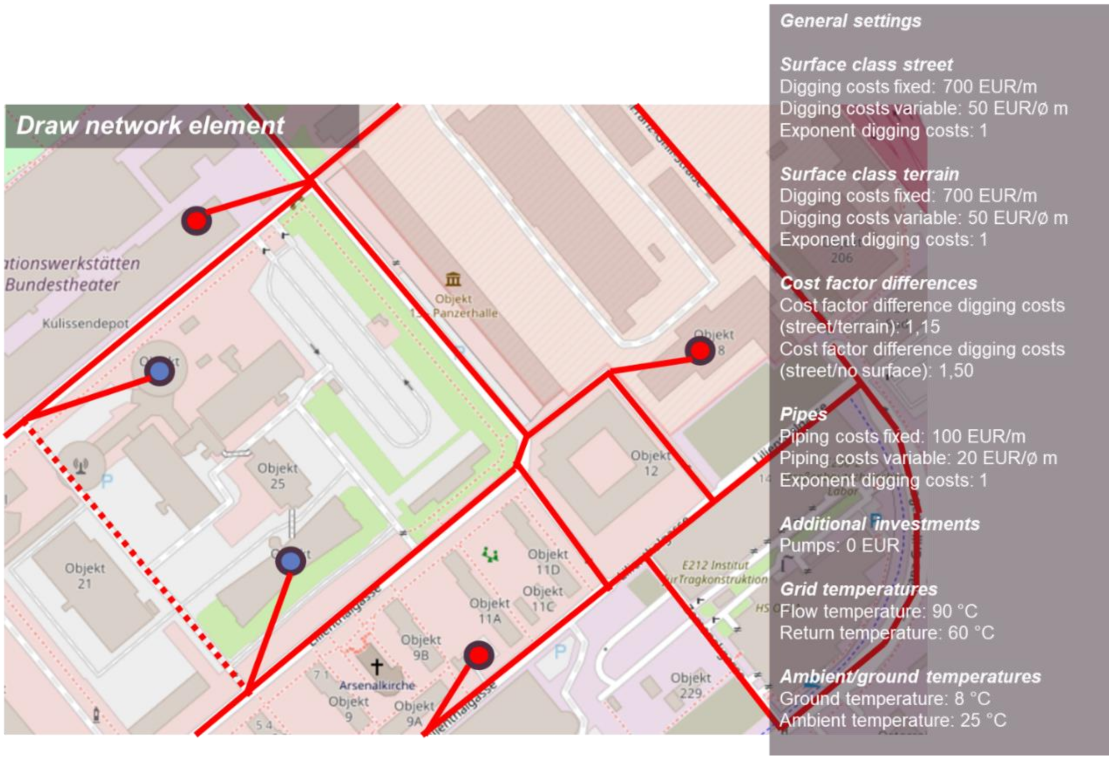
Average Ground Temperature (ground_temp) Required

8 ºC

Yearly average ground temperature in °C.

Figure 96: Input window for GIS module (mock-up).

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement N°847121



Draw network element

General settings

Surface class street
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Surface class terrain
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Cost factor differences
Cost factor difference digging costs (street/terrain): 1,15
Cost factor difference digging costs (street/no surface): 1,50

Pipes
Piping costs fixed: 100 EUR/m
Piping costs variable: 20 EUR/ø m
Exponent digging costs: 1

Additional investments
Pumps: 0 EUR

Grid temperatures
Flow temperature: 90 °C
Return temperature: 60 °C

Ambient/ground temperatures
Ground temperature: 8 °C
Ambient temperature: 25 °C

Figure 97: Adding network element to the graph (mock-up).

Draw network element

Element xy

Surface type: street
Existing grid element: No
Overland pipe: No
Restriction: must build

General settings

Surface class street
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Surface class terrain
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Cost factor differences
Cost factor difference digging costs (street/terrain): 1,15
Cost factor difference digging costs (street/no surface): 1,50

Pipes
Piping costs fixed: 100 EUR/m
Piping costs variable: 20 EUR/ø m
Exponent digging costs: 1

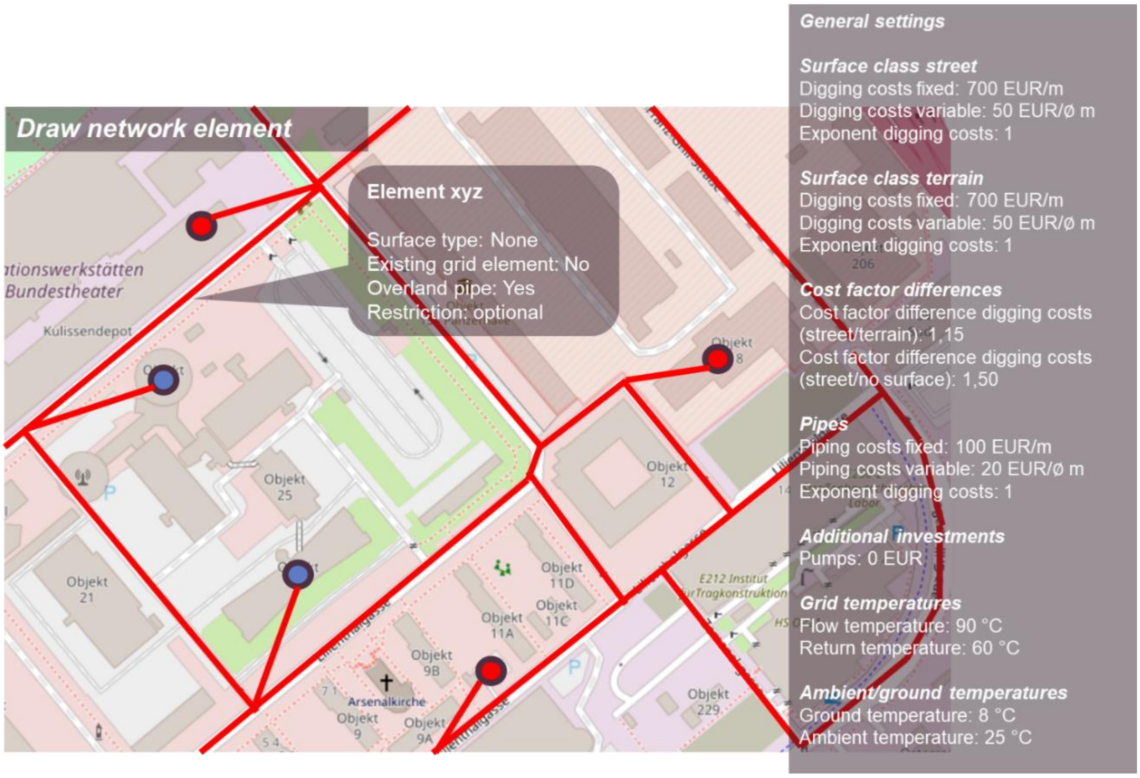
Additional investments
Pumps: 0 EUR

Grid temperatures
Flow temperature: 90 °C
Return temperature: 60 °C

Ambient/ground temperatures
Ground temperature: 8 °C
Ambient temperature: 25 °C

Figure 98: Changing surface type of road element (mock-up).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121



Draw network element

Element xyz

Surface type: None
Existing grid element: No
Overland pipe: Yes
Restriction: optional

General settings

Surface class street
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Surface class terrain
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Cost factor differences
Cost factor difference digging costs (street/terrain): 1,15
Cost factor difference digging costs (street/no surface): 1,50

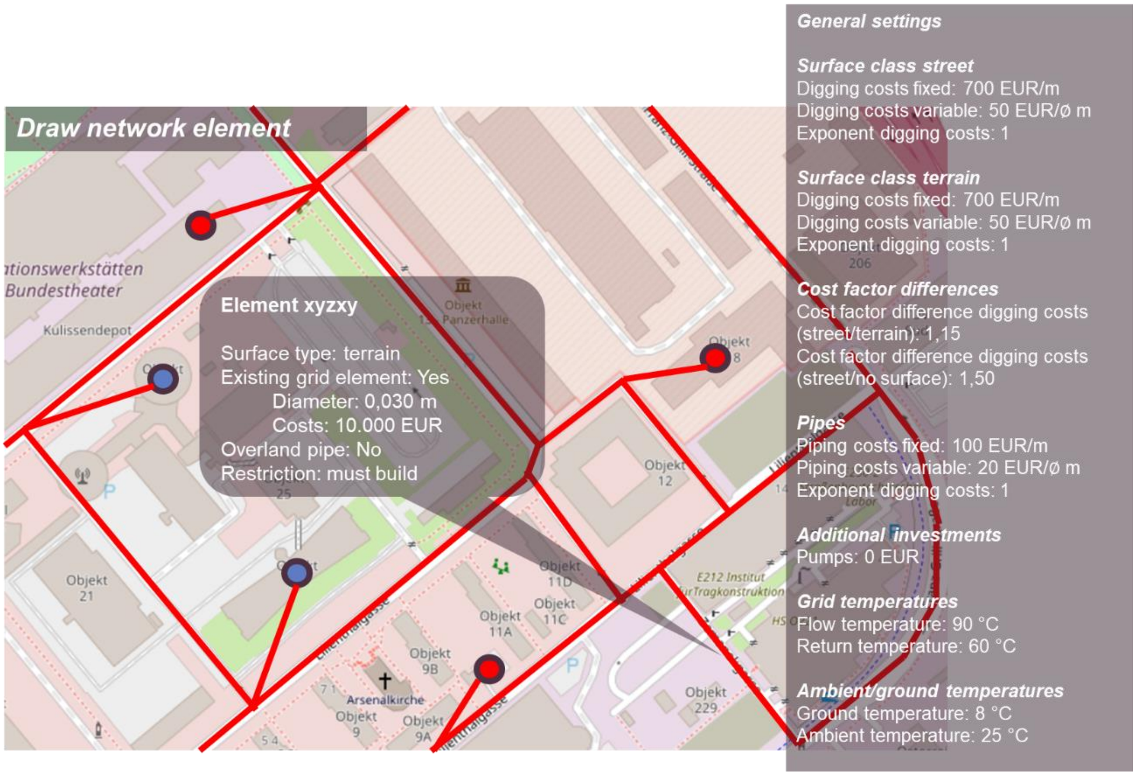
Pipes
Piping costs fixed: 100 EUR/m
Piping costs variable: 20 EUR/ø m
Exponent digging costs: 1

Additional investments
Pumps: 0 EUR

Grid temperatures
Flow temperature: 90 °C
Return temperature: 60 °C

Ambient/ground temperatures
Ground temperature: 8 °C
Ambient temperature: 25 °C

Figure 99: Setting restrictions on the element (mock-up).



Draw network element

Element xyzxy

Surface type: terrain
Existing grid element: Yes
Diameter: 0,030 m
Costs: 10.000 EUR
Overland pipe: No
Restriction: must build

General settings

Surface class street
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Surface class terrain
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/ø m
Exponent digging costs: 1

Cost factor differences
Cost factor difference digging costs (street/terrain): 1,15
Cost factor difference digging costs (street/no surface): 1,50

Pipes
Piping costs fixed: 100 EUR/m
Piping costs variable: 20 EUR/ø m
Exponent digging costs: 1

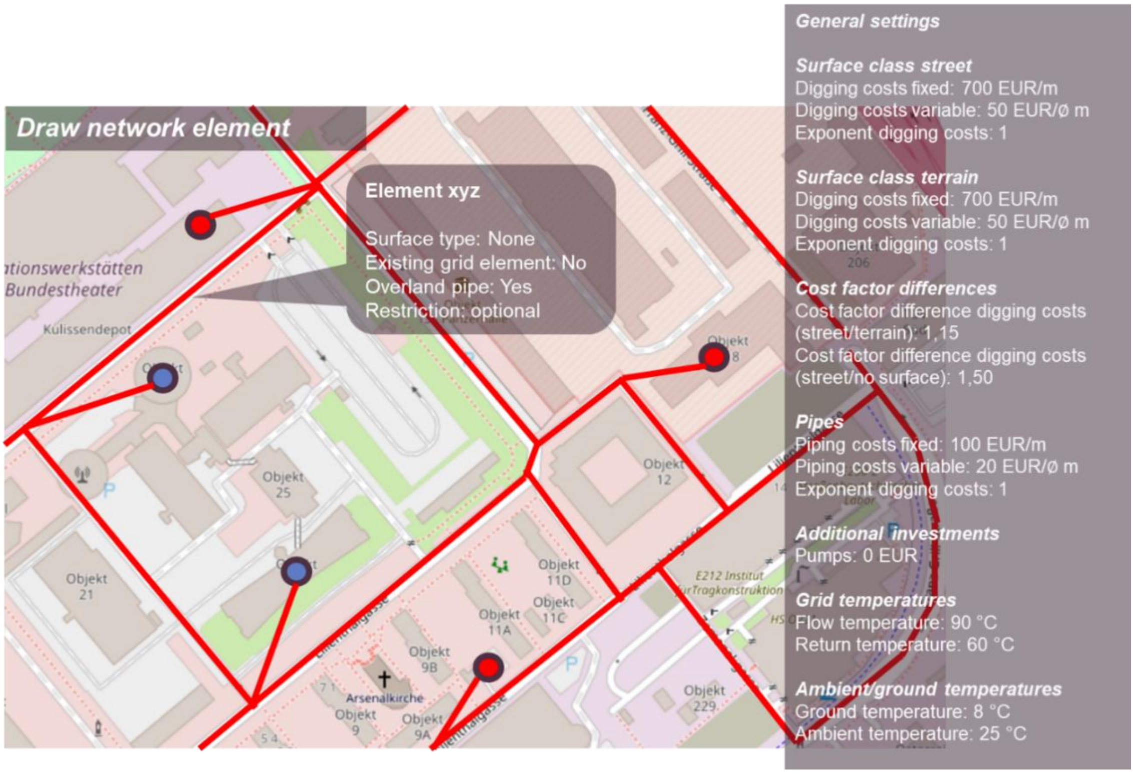
Additional investments
Pumps: 0 EUR

Grid temperatures
Flow temperature: 90 °C
Return temperature: 60 °C

Ambient/ground temperatures
Ground temperature: 8 °C
Ambient temperature: 25 °C

Figure 100: Adding existing network element (mock-up).

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N°847121



Element xyz

Surface type: None
Existing grid element: No
Overland pipe: Yes
Restriction: optional

General settings

Surface class street
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/Ø m
Exponent digging costs: 1

Surface class terrain
Digging costs fixed: 700 EUR/m
Digging costs variable: 50 EUR/Ø m
Exponent digging costs: 1

Cost factor differences
Cost factor difference digging costs (street/terrain): 1,15
Cost factor difference digging costs (street/no surface): 1,50

Pipes
Piping costs fixed: 100 EUR/m
Piping costs variable: 20 EUR/Ø m
Exponent digging costs: 1

Additional investments
Pumps: 0 EUR

Grid temperatures
Flow temperature: 90 °C
Return temperature: 60 °C

Ambient/ground temperatures
Ground temperature: 8 °C
Ambient temperature: 25 °C

Figure 101: Setting overland pathway for pipes (mock-up).

3.2.4 Running a test case using the standalone version of the GIS

3.2.4.1 Description of the Test Case

The user could run a simple model using only the GIS module in this test case. Users could also run their test cases using the same structure. Since this is a standalone version of the module, there is no GUI. Therefore, inputs are provided in a python script. Users could change these inputs to run the module with their data. The user will have the following outputs after running the test case:

- **An output map (HTML file) that illustrates the found network solution. The user can click on certain pipe elements and check the attributes of individual pipe elements. A pipe element is always defined as a section between two street junctions or a street junction and a source or sink.**
- **An Excel (xlsx) file that shows the network length, losses, accumulated installed pipe capacity, and network costs between all source/sink pairs. The sum row shown at the bottom of the table illustrates the costs of the whole network accounting for the fact of not double counting any costs or other factors as certain individual connections may share the same pipe elements.**

3.2.5 Data and Instruction to run the Model

This simple test case considers a single heat source and two demand points (sinks). The data used for sources and sinks are given in Table 111.

Table 111: Data of sources and sink in the test case.

Type	Coordinates (x,y)	Peak Capacity
Source	(47.78022, 13.03961)	30
Sink	(47.78159, 13.03819)	1

Sink	(47.78151, 13.03918)	1

The project area where the Open Street Map data is loaded is defined by the following data in the test case:

- northern latitude = 47.783075,
- southern latitude = 47.780213,
- eastern longitude = 13.040456,
- western longitude = 13.036935.

The cost and environmental data used in the test case and relevant to the user are given in Table 112. Here relevant to the user means these variables have an effect on the solution. The remaining cost and environmental data used in the test case are given in Table 113. These variables are called not relevant because all surfaces are considered streets, and all pipes are considered ground pipes in the simple test case. In other words, these variables have no effect on the solution.

Table 112: The cost and environmental data used in the test case – relevant to the user.

Variable	Explanation	Value
fc_dig_st	fixed share of digging costs along a street in [EUR/m]	350
vc_dig_st	variable share of digging costs along a street in [EUR/m2]	700
vc_dig_st_ex	the exponent of the cost formula of the street surface	1
fc_pip	fixed share of piping costs along a street in [EUR/m]	50
vc_pip	variable share of piping costs along a street in [EUR/m2]	700
vc_pip_ex	the exponent of the piping cost formula	1
ground_temp	yearly average ground temperature in °C	8
invest_pumps	investment cost of pumps	0
water_den	water density at a specific temperature (average of flow and return temperatures) in kg/m3.	1000
heat_capacity	heat capacity at a specific temperature (average of flow and return temperatures) in J/kgK.	4.18

Table 113: The cost and environmental data used in the test case – not relevant to the user.

Variable	Explanation	Value
factor_street_terrain	cost difference factor of street and terrain surfaces	1
factor_street_overland	cost difference factor of street surface and overland pipes	1
fc_dig_tr	fixed share of piping costs along a terrain in [EUR/m]	200
vc_dig_tr	variable share of piping costs along a terrain in [EUR/m2]	500
vc_dig_tr_ex	the exponent of the cost formula of the terrain surface	1
ambient_temp	yearly average ambient temperature in °C	25

If the users want to use their data, the GIS expects the following inputs from the user

- Dictionary of source points. Thus input defines the information of sources. All source points need to have the following structure:

1: {"coords": (47.78022, 13.03961), "cap": 30} where

  - 1 represents the ID of the source. This always needs to be an integer and not a string.
  - “cords” represents the x,y coordinates,
  - “cap” represents the peak capacity in MW.

The input for sources is as follows for the test case:

```
N_supply_dict = {
1: {"coords": (47.78022, 13.03961), "cap": 30}
}
```

- Dictionary of sink points. Thus, input defines the information of sinks. All sink points need to have the following structure:

3: {"coords": (47.78159, 13.03819), "cap": 1} where

  - 3 represents the ID of the sink. This always needs to be a number and not a string,
  - “cords” represents the x,y coordinates,
  - “cap” represents the peak capacity in MW.

The input for sinks is as follows for the test case:

```
N_demand_dict = {
3: {"coords": (47.78159, 13.03819), "cap": 1},
2: {"coords":(47.78151, 13.03918), "cap": 1}
}
```

- A list of the project area’s coordinates. This input defines the boundary box where the Open Street Map data will be loaded. It needs to have the following structure:

project_area = [northern latitude, southern latitude, eastern longitude, western longitude] where all the latitude and longitude values are floats.

The project area input is as follows for the test case:

```
project_area = [47.783075, 47.780213,13.040456, 13.036935]
```

- The cost and environmental data given in Table 112 and Table 113 are defined as follows for the test case:

```
•  fc_pip = 50
•  vc_pip = 700
•  fc_dig_st = 350
•  vc_dig_st = 700
•  ground_temp = 8
•  factor_street_terrain = 1
•  factor_street_overland = 1
•  fc_dig_tr = 200
•  vc_dig_tr = 500
•  invest_pumps = 0
•  vc_dig_st_ex = 1
•  vc_dig_tr_ex = 1
•  vc_pip_ex = 1
•  water_den = 1000
•  heat_capacity = 4.18
•  ambient_temp = 25
```

3.2.6 Results of the Test Case

If all inputs are given as described above, the user can run the code and create the following outputs to the working directory:

- **An HTML file called GIS.html: illustrates the network solution. By clicking on a particular network element, the user gets displayed the attributes of that network element. Sources are marked red, while sinks are marked blue (Figure 102 and Figure 103).**
- **An xlsx file called Results_GIS.xlsx: shows the network length, losses, accumulated installed pipe capacity, and the network costs between all source/sink pairs (Figure 104). Please note that the first column “From/to” defines where the pipe starts and where the pipe ends. In other words, it indicated the IDs of the source and the sink connected by the respective pipe.**

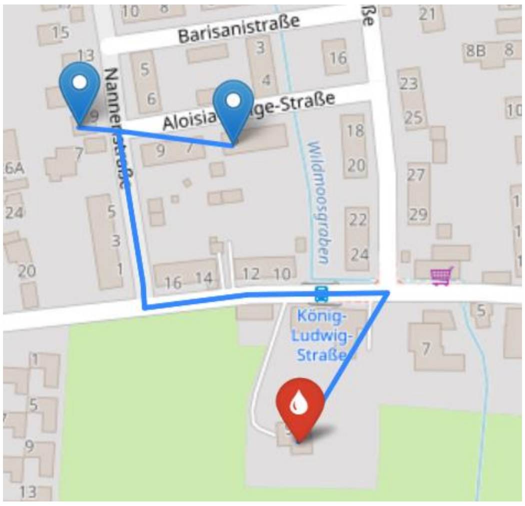
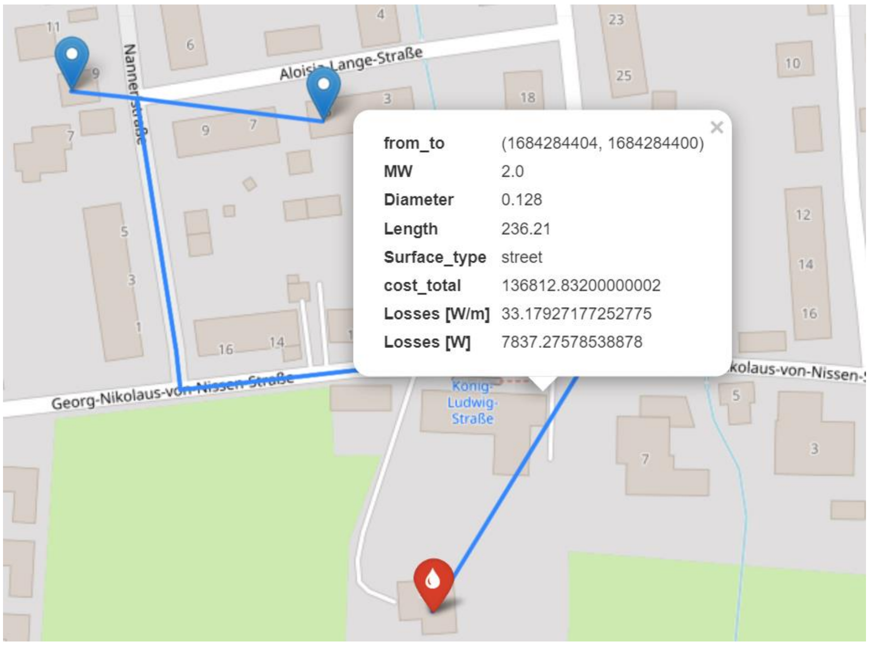
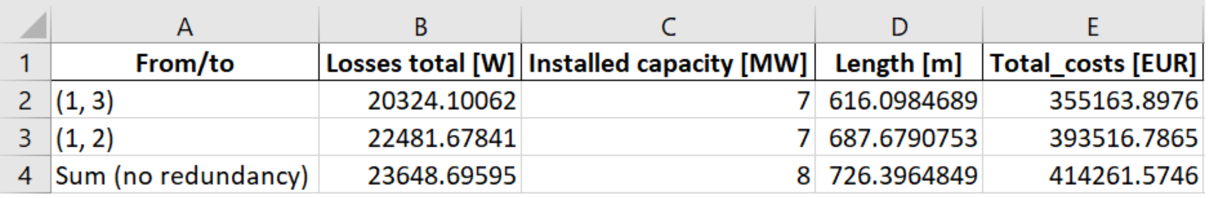


Figure 102: Illustration of network solution in HTML file.



from_to	(1684284404, 1684284400)
MW	2.0
Diameter	0.128
Length	236.21
Surface_type	street
cost_total	136812.83200000002
Losses [W/m]	33.17927177252775
Losses [W]	7837.27578538878

Figure 103: Attributes of the network element in HTML file.



From/to	Losses total [W]	Installed capacity [MW]	Length [m]	Total_costs [EUR]
(1, 3)	20324.10062	7	616.0984689	355163.8976
(1, 2)	22481.67841	7	687.6790753	393516.7865
Sum (no redundancy)	23648.69595	8	726.3964849	414261.5746

Figure 104: Excel file illustrating the network losses, capacities, length, and costs.

3.3 References

[1] T. Nussbaumer, S. Thalmann, A. Jenni, and J. Ködel, “Handbook on Planning of District Heating Networks.” Aug. 21, 2020. Accessed: Feb. 01, 2021. [Online]. Available: http://www.verenum.ch/Dokumente/Handbook-DH_V1.0.pdf

[2] THERMOS, “THERMOS Project.” 2020. Accessed: Dec. 20, 2020. [Online]. Available: https://www.thermos-project.eu