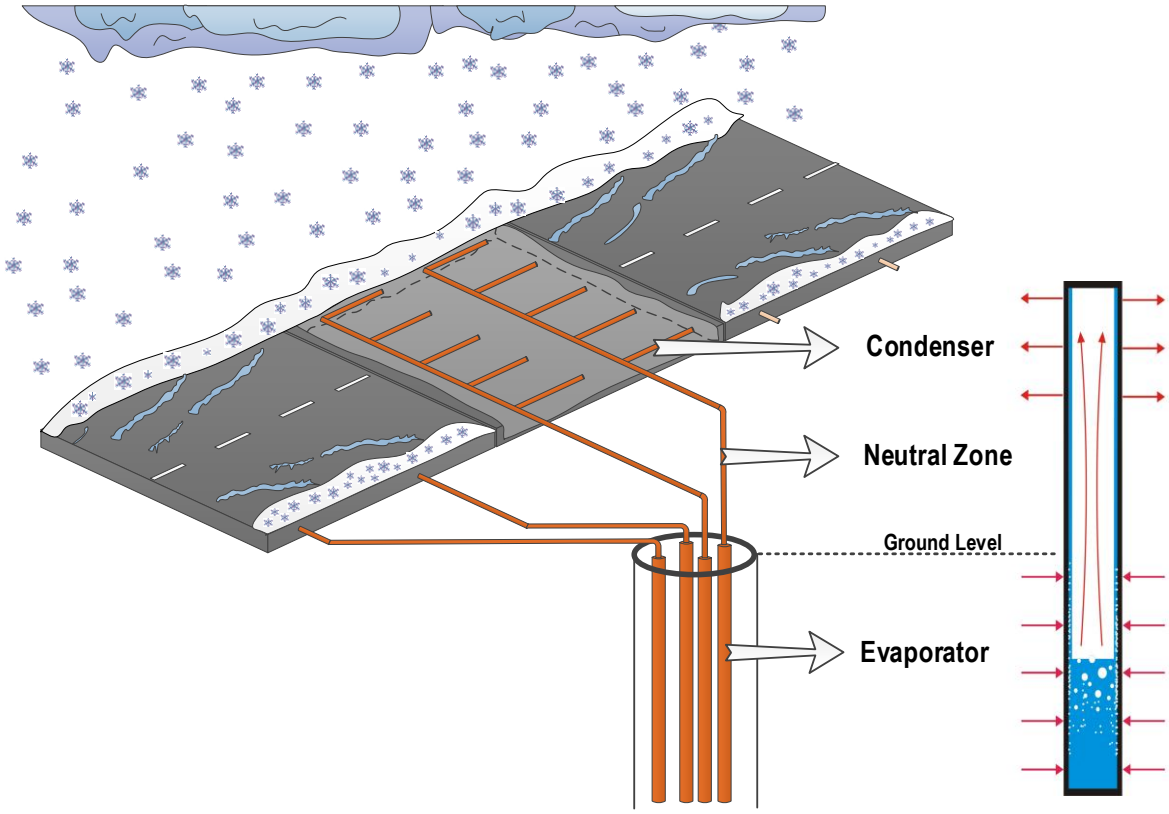
ZAE BAYERN

User handbook

for

GERDPy



The Simulation Tool for Geothermal Heat Pipe Surface Heating Systems

Authors: Yannick Apfel, Meike Martin

Content

Abbreviations

Abbre-viation	**Meaning**
CHS	Cylindrical Heat Source
DWD	Deutscher Wetterdienst
FLS	Finite Line Source
GUI	Graphical User Interface
ILS	Infinite Line Source

Notation

Latin characters

Symbol	Unit	Meaning
A_{he}	[m²]	heating element surface area
a	[m²/s]	thermal diffusivity
B	[m] / [0,8]	horizontal borehole distance / cloudiness
b_m	[-]	coordinate-dependent geometric coefficient
b_{mn}	[-]	coordinate-dependent geometric coefficient
b'_{mn}	[-]	coordinate-dependent geometric coefficient
C	[-]	factor for gas type
C_{WB}	[-]	ratio of emission coefficients of wall and backfill
c_p	[J/kgK]	specific isobaric heat capacity
D	[m]	vertical borehole buried depth
D_{he}	[m]	vertical thickness of heating element
d	[m]	particle diameter
d_{mn}	[m]	horizontal distance between boreholes m and n
d_{pa}	[m]	outer diameter of a thermosiphon pipe
d_{pi}	[m]	inner diameter of a thermosiphon pipe
e_o, e_u	[-]	auxiliary variables for sum-term evaluation
g	[-]	g-function
H	[m]	borehole length/depth
H_{total}	[m]	total borehole length/depth of the borefield
$\Delta h_{Ph,l\rightarrow v}$	[J/kg]	phase-change enthalpy liquid <-> vapour
$\Delta h_{Ph,s\rightarrow l}$	[J/kg]	phase-change enthalpy solid <-> liquid
i	[-]	timestep
k	[-]	total number of timesteps
l_a	[m]	free path length of air molecules
l_{conn}	[m]	total length of connection line between boreholes and heating element
$l_{p,he}$	[m]	total length of condenser piping within heating element

$\boldsymbol{M}$	[kg/mol]	molar mass
$\boldsymbol{m}$	[kg]	mass
$\boldsymbol{N}$	[-]	number of thermosiphons per borehole
$\boldsymbol{N_1}, \boldsymbol{N_2}$	[-]	auxiliary variables for sum-term evaluation
$\boldsymbol{n_b}$	[-]	number of boreholes per borefield
$\boldsymbol{n_q}$	[-]	number of segments per borehole
$\boldsymbol{p}$	[Pa]	pressure
$\boldsymbol{Pr}$	[-]	Prandtl-number
$\dot{\boldsymbol{Q}}$	[W]	thermal power/flux
$\overline{\boldsymbol{Q}}$	[W/m]	average thermal power/flux per borehole length
$\dot{\boldsymbol{Q}}_{\boldsymbol{l}}$	[W/m]	thermal power/flux per metre of condenser length
$\dot{\boldsymbol{q}}$	[W/m²]	thermal power/flux per area
$\boldsymbol{R}$	[J/molK]	universal gas constant
$\boldsymbol{R_f}$	[-]	snow-free area ratio
$\boldsymbol{RR}$	[mm/h]	precipitation rate
$\boldsymbol{R^0_{mn}}$	[NxN] of [m*K/W]	matrix of thermal interaction coefficients
$\boldsymbol{r^0_{mn}}$	[m*K/W]	thermal interaction coefficient
$\boldsymbol{R_{th}}$	[K/W]	system thermal resistance
$\boldsymbol{r_{th}}$	[m*K/W]	(specific) thermal resistance per unit length
$\boldsymbol{r_{iso,b}}$	[m]	outer radius of thermosiphon insulation inside borehole
$\boldsymbol{r_{iso,conn}}$	[m]	outer radius of thermosiphon insulation inside connection line
$\boldsymbol{r_{pa}}$	[m]	outer radius of a thermosiphon pipe
$\boldsymbol{r_{pi}}$	[m]	inner radius of a thermosiphon pipe
$\boldsymbol{r_w}$	[m]	radius of thermosiphon epicentres
$\boldsymbol{S_r}$	[mm/h]	snowfall rate (water equivalent)
$\boldsymbol{s_R}$	[m]	horizontal distance between condenser pipe centres
$\boldsymbol{Sc}$	[-]	Schmidt-number
$\boldsymbol{s_c}$	[m]	thickness of additional concrete layer
$\boldsymbol{T}$	[K]	absolute temperature
$\Delta\boldsymbol{T_b}$	[K]	temperature variation at the borehole wall

T_{MR}	[K]	mean radiant temperature of the surroundings
t	[s]	time
Δt_i	[s]	length of a timestep
t_s	[s]	characteristic borefield time
u_{inf}	[m/s]	ambient wind speed
X_{inf}	[kg_v/kg_a]	humidity ratio of ambient air
$X_{sat,surf}$	[kg_v/kg_a]	humidity ratio of saturated air at surface temperature
x_{min}	[m]	minimum vertical pipe-to-surface distance
x_o, x_u	[m]	upper and lower half of the heating element
z	[m]	elevation above ground
z_{asl}	[m]	elevation above sea level

Greek characters

Symbol	Unit	Meaning
α	[W/m^2K]	coefficient of heat transfer
β_o, β_u	[-]	auxiliary variables for sum-term evaluation
β_m	[m/s]	mass transfer coefficient
γ	[-] / [-]	coefficient of accommodation / auxiliary variable for sum-term evaluation
δ	[m] / [$K^{1,81}/Pa$]	particle surface roughness / binary diffusion coefficient
ϵ	[-]	emissivity coefficient
Θ	[°C]	temperature
κ	[-]	heat transmission coefficient
λ	[W/m*K]	thermal conductivity
ν	[m^2/s]	kinematic viscosity
ρ	[kg/m^3]	density
σ	[W/m^2K^4]	Stefan-Boltzmann constant
σ_λ	[-]	ratio of thermal conductivities
φ	[-] / [%]	contact area ratio / relative humidity

Indices

Symbol	Meaning
WB	wall-to-backfill
a	air
b	borehole (wall)
c	contact / concrete
con	convection
conn	connection line between borehole and heating element
cs	clear sky
eva	evaporation
f	free of snow
g	ground
he	heating element
hp	heat pipe / thermosiphon
i	timestep index
inf	ambient
iso	thermal insulation
j	thermal resistance index / series running index for sum-term evaluation
k	total number of timesteps
l	liquid
lat	latent
melt	melting
n	thermosiphon index
o	heating element surface/upper half
p	thermosiphon pipe
rad	radiation
s	solid / saturation / snow or ice
sen	sensible
sim	simulation
surf	heating element surface

$\mathbf{u}$	heating element underside/lower half
$\mathbf{v}$	vapour
$\mathbf{w}$	water
$\mathbf{0}$	from the prior timestep $(i-1)$

1 Introduction

1.1 Purpose of GERDPy

GERDPy is a Python-based simulation tool designed to model directly geothermally heated surface de-icing systems using a two-phase thermosiphon and was specifically developed for the project GERDI. Thus, this modelling tool represents a very specific system and is not suitable as a general modelling software for borehole heat exchangers. For a detailed description of the use case please refer to chapter 1.3 of this manual.

Given local ground parameters, comprehensive geometric data of the system to be analysed and hourly weather data as user inputs, GERDPy conducts a time series analysis of the surface heating system and produces, among other parameters, hourly data on the surface heat flux and temperatures at the surface and borehole wall and thus provides essential data for dimensioning a surface heating system correctly in relation to the ground source.

1.2 The GERDI project and surface de-icing systems

The GERDI-project encompasses the development and testing of a thermal de-icing surface heating system that makes direct use of shallow geothermal energy with borehole lengths not exceeding 400 metres (Zorn et al. 2015). While most hydraulic systems utilize either a conventionally heated pumped fluid or a heat-pump, the GERDI-system makes direct use of the shallow geothermal heat source by integrating multiple two-phase thermosiphons into a borehole. Possible use cases of a surface heating system with a plane surface include ramps, entrances, platforms, bridge decks, parts of a road or highway, walking/biking paths and parking lots.

A two-phase thermosiphon is a type of wickless heat pipe that uses the gravitational pull as the driving force instead of capillary forces. The system uses CO_2 as a working fluid at an operation pressure of around 45 bar. The working principle of a two-phase thermosiphon also constitutes its main advantages as an effective heat transfer device that works passively without an additional energy supply and can transport high heat fluxes given relatively low temperature deltas between the ground (heat source) and the surface (heat sink). The heat can be carried through small cross-sectional areas over considerable distances. The thermosiphon is composed of a sealed container, usually tube-shaped, that contains a working fluid in two-phase state. The working fluid is vaporized at the heat source in the ground where heat is supplied to the outer wall of the thermosiphon (evaporator). The resulting vapour pressure drives it up where its latent heat is released into the pipe wall and surface heating element through condensation (condenser). Gravitational forces then bring the liquid back down the pipe wall where it re-accumulates in the liquid pool at the evaporator. This endless loop will start if the temperature difference between ground and surface suffices to overcome the frictional forces inside the thermosiphon and will be sustained as long as the condition above is met. With long thermosiphons like the ones used in GERDI, the middle part can often be considered as adiabatic. The following figure provides a schematic of the thermosiphon (Figure 1). (Faghri 2018, S. 2163–2212) (Zohuri 2016, S. 4)

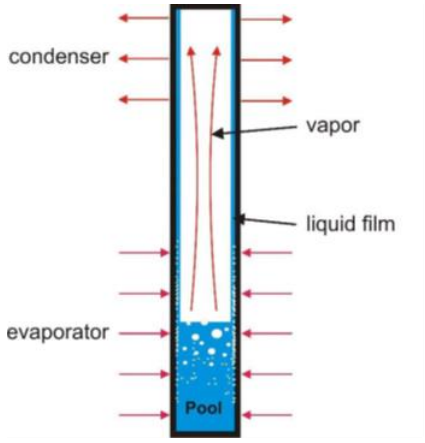


Figure 1: Schematic of a thermosiphon

Constituting the basic heat transfer component of the GERDI surface de-icing system, the thermosiphons need to be integrated into the whole heating system like depicted on the front page of this document. The condenser section of the system is part of the surface that needs to be heated and is located directly below it. It is connected to one or multiple boreholes that each contain a bundle of up to ten or more thermosiphon pipes and reach between 30 to 100 m deep into the ground. The connection lines between the vertical part (borehole) and the horizontal part (heating surface) as well as the pipes inside the condenser section need to be angled towards the surface as to enable the backflow of the liquid phase. A typical angling of the condenser section would be around 1-2°.

Depending on the surface area and thermophysical properties of the ground the system of boreholes and their individual depths directly influence the heating performance on the surface, thus rendering the geometric configuration a very important dimensioning factor for designing such a system.

1.3 Scope of GERDPy

This chapter shall serve as a summary of the capabilities and limits of this simulation tool. After a short description of the entities of the GERDI-system, its basic function is explained. This is followed by the outputs that GERDPy provides.

The GERDI-system, as schematically depicted in Figure 2, is connected to the ground (1) which serves as the heat source. The borehole is made up of the backfill material (2) which can contain up to ten or more heat pipes (4) arranged in a circle within the confines of the borehole wall (3). The heat pipe configuration needs to be identical for all boreholes in the borefield. The heating element (5) of arbitrary surface area (6) serves as the heat sink of the system and gives off the surface heat flux propagated through the thermosiphon pipes. The system starts to transport heat by virtue of a continuous loop of evaporation and condensation like explained in chapter 1.2. The heat flux on the surface is therefore a direct function of the temperature difference between ground and surface. As the surface temperature itself depends both on the heat flux delivered and the ambient weather conditions both temperature and heat flux are interdependent. As hinted above, the thermosiphon system works autonomously without needing an external power source.

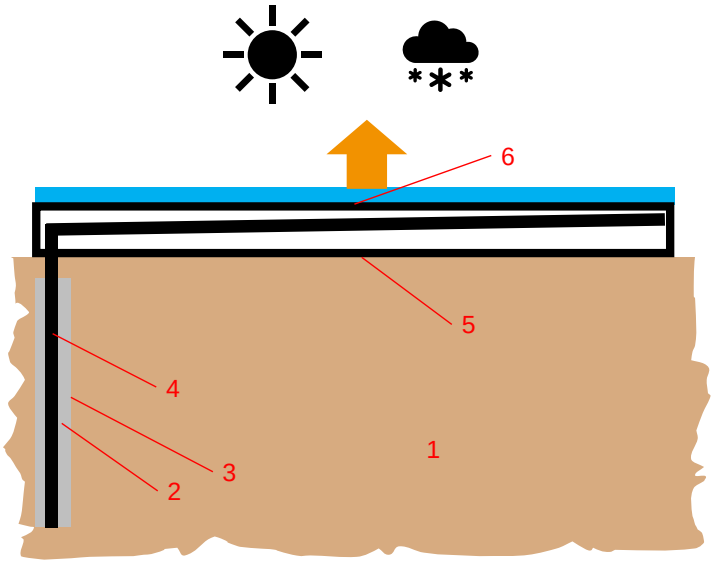


Figure 2: Basic entities of the GERDI-system

GERDPy is a GUI-based tool developed in Python to conduct simulations of the GERDI-system. The physical algorithm was programmed in *Python 3.7* using object-oriented programming. The GUI was developed with *Qt Designer* and the Python module *PySide 6*. The GUI is explained in detail in chapter 3. Basically, the user needs to input all relevant data on multiple input pages of the GUI starting with thermophysical parameters of the ground, followed by thermophysical and geometric parameters of the heat pipes within the borehole and heating element and provide information on the x-y-layout of the boreholes within the borefield. The boreholes cannot be tilted towards vertical and may only have identical diameters. The borehole configuration as well as a time series of hourly weather data need to be formatted in a predefined way and provided as .csv-files. They are then imported via the GUI. The simulation time, which is also provided by the user, cannot fall short of one hour (as this is the timestep) but can be as long as 20 years or more. A reasonable simulation time would be multiple years as that would allow monitoring of the ground temperature level before the start of the annual heating period.

As opposed to some design programs that optimize for instance the borehole length given a heat flux requirement of the system, GERDPy doesn't optimize parameters but instead conducts a system simulation that then returns the change of certain values like the surface temperature of the heating element over time. To optimize for critical variables like the surface or borehole temperatures, the user needs to manually tweak the input parameters and repeat the simulation to obtain a different output.

The tool produces three main parameters as simulation results which are provided as plots of the value over time. These three parameters are the **borehole wall temperature**, the **heating element surface temperature** and the **surface heat flux** for every hour of the simulation. The latter is additionally used to calculate the amount of energy that was extracted from the ground over the simulated period. The plot of the borehole wall temperature over time represents the cooling of the ground and is an important indicator for system sizing as an adequately sized borefield in relation to the surface heat sink should yield constant ground temperatures at the start of the annual heating period to guarantee a constant system performance. The simulation results can be saved as figure plots or to a .csv-file.

1.4 Third party software tools

The development of GERDPy is fully based on open-source toolboxes and libraries within Python. A few of the modules are directly based on the open-source toolbox for the evaluation of thermal response factors for geothermal borehole fields, *Pygfunction (v1.1.2)*[1]. More details on what was taken from the toolbox and how the modules were adapted can be found in the following chapter (2.1.1). The GUI was created using the PySide interface (support for PyQt) created by Wanderson M. Pimenta[2].

[1] https://github.com/MassimoCimmino/pygfunction (last accessed 18.07.2022); because the library is subject to constant improvements and updates, the version current during the development of GERDPy was extracted to ensure compatibility. This is release 1.1.2 of Pygfunction.

[2] https://github.com/Wanderson-Magalhaes/Modern_GUI_PyDracula_PySide6_or_PyQt6 (last accessed 18.07.2022)

2 Theory

2.1 Structure of the software tool GERDPy

This chapter dives into the inner workings of GERDPy. In chapter 2.1.1 the module architecture of the software tool is explained. Even though the modules cannot be strictly categorized into different functional groups because some modules serve functions in different parts of the model an attempt is made to sort them into different clusters as to create structure. That chapter is followed by a detailed description of the main snow-melting algorithm (chapter 2.1.2).

2.1.1 Python module-architecture

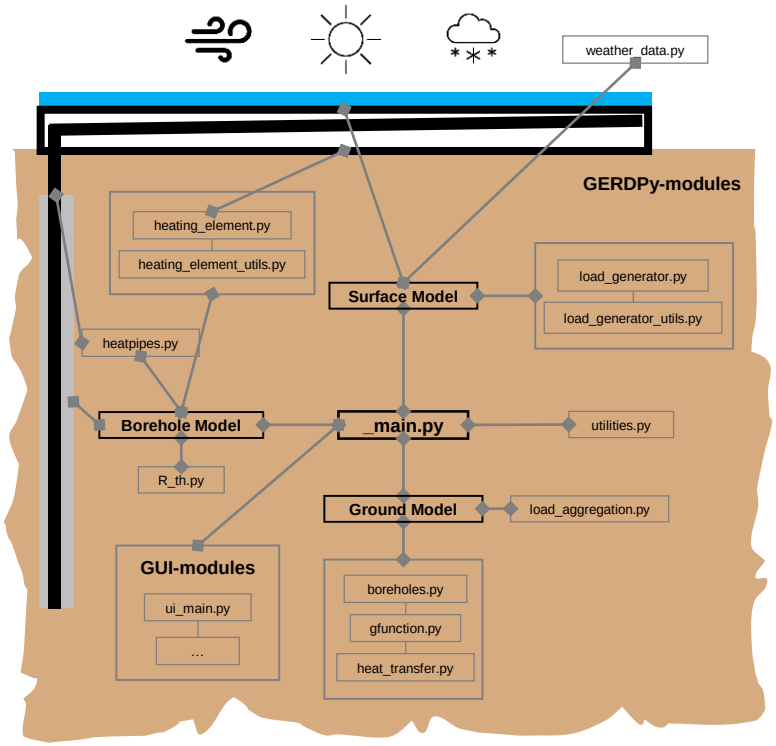


Figure 3: GERDPy module architecture

The GERDPy-tool is comprised of three sub-models, namely the ground model, the borehole model and the surface model.

The role of the **ground model** is to provide a temperature at the borehole wall for each timestep that results from the heat flux extracted from the ground. In essence, the model calculates the cooling reaction of the ground over time when exposed to heat extraction by boreholes in a borefield. This is achieved through the temporal and spatial superposition of g-functions, also known as thermal response functions, that model the thermal interference

of boreholes within said borefield. The modules allocated to the ground model are “borehole.py”, “gfunction.py”, “heat_transfer.py” and “load_aggregation.py”. All four were taken from the open-source tool *Pygfunction* by Massimo Cimmino which is a Python library for the calculation of thermal response factors for arbitrary borefield configurations and depths. In fact, the library was largely adopted and integrated into GERDPy with only minor changes to the code to create an interface with the rest of GERDPy. It relies on the analytical finite line source solution more closely described in chapter 2.2.1.

The **borehole model** serves as the system thermal resistance. By collecting the necessary geometric information from the modules “heating_element.py”, “heating_element_utils.py” and “heatpipes.py” and concatenating the thermal resistances of all components, the thermal resistance of the entire system is calculated within “R_th.py”. Inside the main-function “_main.py” the system thermal resistance is used to calculate the surface temperature for the next timestep by subtracting the product of heat flux and system thermal resistance from the borehole wall temperature, essentially coupling the ground with the surface. The equations used are more closely examined in chapter 2.2.2.

The **surface model** contains the snow-melting algorithm more closely described in chapter 2.1.2. The detailed model description will follow in chapter 2.2. This sub-model determines the surface heat flux/thermal load for each timestep by solving a thermal power balance on the surface of the heating element given known weather conditions like wind speed, ambient temperature, snowfall rate, etc. This sub-model is made up of the modules “load_generator.py” and “load_generator_utils.py”. While the former contains the most essential parts of the snow-melting algorithm, the latter harbours the detailed physical modelling equations. The weather data is imported by the module “weather_data.py”.

The modules within the three sub-models are linked and controlled via the module “_main.py”. It plays the main role in the execution of the algorithm. The GUI-modules include the GUI-template used and additional functionality to increase robustness and usability, e.g. to check for incompatible geometric configurations and produce an error.

2.1.2 Snow-melting algorithm

Like mentioned earlier, the heat flux for each timestep depends on the weather conditions present at the surface and thus is not a controlled parameter. To determine the heat flux a thermal power balance, which stands at the centre of the whole algorithm, is solved at every timestep. The power balance sums up all thermal phenomena that happen at the surface as a result of the latter interacting with the surroundings. Table 1 provides a list of heat transfer mechanisms that can be modelled and shows which ones fall within the scope of GERDPy. The thermal power balance used as well as the underlying physical empirical correlations are well established in pertinent literature, e.g. (ASHRAE 2015) or (VDI 2013). The correlations used are documented in detail in chapter 2.2.

Table 1: Heat transfer mechanisms on the surface of a de-icing system

	Snow/ice melting			***Surface losses/gains***				
Thermal mechanism	latent heat	sensible heat	sublimation	convection	radiation		evaporation	condensation
					direct	diffuse		
Covered by GERDPy?	✓	✓	×	✓	×	✓	✓	×

The heat transfer mechanisms considered in GERDPy include latent and sensible heat for melting snow/ice and surface losses through convection, diffuse radiation and evaporation of surface water. Among the effects for “snow/ice melting” sublimation, which represents direct evaporation of snow/ice from solid state, wasn’t covered because the effect is negligible at lower altitudes. Radiation can in fact be divided into two separate components, direct and diffuse radiation. In case of cloudy conditions, the interaction via diffuse radiation is dominant which causes a net cooling effect on the surface due to interaction with the colder cloud layer. On a sunny day direct radiation is dominant and the sun will heat up the surface element. Direct radiation was neglected because it doesn’t lead to a net positive heat extraction from the ground. The same goes for condensation which is in some way the inverse of evaporation and leads to a heat input into the surface element.

Summing up the five mentioned mechanisms one yields the following heat balance equation on the surface, see equation (1). A net positive system heat flux equates to heat being drawn from the ground. A net negative system heat flux $\dot{Q}$ would lead to a heat up of the surface element as the GERDI-system is not designed to transport heat back into the ground.

$$\dot{Q} = \dot{Q}_{lat} + \dot{Q}_{sen} + \dot{Q}_{con} + \dot{Q}_{rad} + \dot{Q}_{eva} \tag{1}$$

Within the snow-melting algorithm this equation is solved in two different variants, one where $\dot{Q} \geq 0$ (net positive heat extraction from the ground) and one where $\dot{Q} \coloneqq 0$ (no heat extraction from the ground, power set to zero). Additionally, a factor called the snow-free area ratio (ranging from 0 to 1) is introduced which diminishes the effect of the loss terms convection, radiation and evaporation. It is set to $\coloneqq 1$ for a surface free of snow and set to $\ll 1$ to account for the insulating effect in case a layer of snow is present. As the insulating effect of a possible snow layer is unknown, this factor can/should be parametrized by the user with experimental or some other form of empirical data. The default value within GERDPy in case of snow cover was set to 0,2. Figure 4 is an illustration of the snow-free

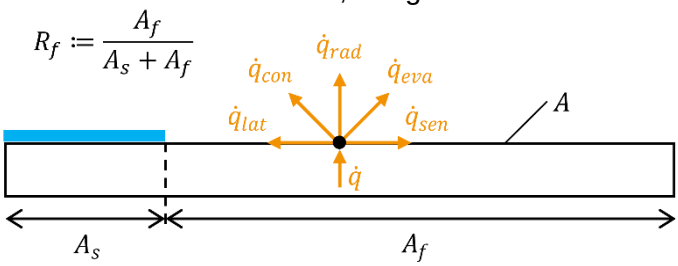


Figure 4: Illustration of the thermal balance

area ratio and the heat transfer terms. $\dot{q}$ denotes a heat flux per area of heating element. The index “s” denotes an area covered in snow; “f” denotes an area free of snow.

For reasons of computational robustness and speed the thermal balance is solved only for steady state, meaning that the inherent thermal capacity of the system is neglected. Thusly, the sums of all heat transfer mechanisms entering versus leaving the heating element are equal at any given timestep and the whole equation is set to zero. Equations (2) and (3) represent the two aforementioned formulations of the thermal balance F. The first one is used in case of a net positive heat flux from the ground ($\dot{Q} \geq 0$) and parametrized with and solved for $\dot{Q}$. The surface temperature is formulated as a function of the heat flux

and determined later within the algorithm (see below). The second one represents the simplified/reduced version with $\dot{Q} \coloneqq 0$ (the ground isn't involved). It is parametrized with and solved for the surface temperature. For both versions only the latent heat term is non-dependent on the parameters $\dot{Q}$ or θ. Both are interdependent via the system thermal resistance: $\theta_{surf} = \theta_b - \dot{Q} \cdot R_{th}$.

$$F(\dot{Q}) = -\dot{Q} + \dot{Q}_{lat} + \dot{Q}_{sen}(\dot{Q}) + R_f \left(\dot{Q}_{con}(\dot{Q}) + \dot{Q}_{rad}(\dot{Q}) + \dot{Q}_{eva}(\dot{Q})\right) \coloneqq 0 \quad (2)$$

$$F(\theta_{surf}) = \dot{Q}_{lat} + \dot{Q}_{sen}(\theta_{surf}) + R_f \left(\dot{Q}_{con}(\theta_{surf}) + \dot{Q}_{rad}(\theta_{surf}) + \dot{Q}_{eva}(\theta_{surf})\right) \coloneqq 0 \quad (3)$$

Figure 5 depicts the snow-melting algorithm designed for GERDPy. It revolves around the evaluation of the thermal balance equations introduced above. Some of the symbols used in the figure include the temperature θ on the surface (index "surf") and at the borehole wall (index "b"), the system thermal resistance R_{th}, the snow mass m_s, timestep i, snowfall rate S_r and the snow-free area ratio R_f.

The program starts by importing all relevant data, including the g-function of the borefield, R_{th} of the system, other system parameters and pre-allocated empty time series variables to store the simulation results. All weather data is imported as time series tuples of the same length in hourly units, e.g. with 730 entries for a month-long simulation. The snow-melting algorithm is then started and repeated in a loop of hourly iterations (e.g. 730 iterations for a month-long simulation).

In contrast to other surface de-icing models that calculate the necessary heat flux based on an instantaneous melting of the snow, GERDPy allows a build-up of snow between timesteps by keeping track of snow/water accumulation on the surface, e.g. if $(\partial m_s)/\partial t > 0$ the rate of snowfall S_r surpasses the volume of snowmelt within the timestep, resulting in an effective reduction of the surface heat flux during those timesteps by virtue of an insulating snow layer. This modelling approach is necessitated for the (usually few) cases of timesteps where the system performance is insufficient to melt the snow within the timestep that it falls. The amount of snow or water present on the surface at a given moment is monitored by mass balancing, it is not physically modelled. A possible snow layer can thus only be reduced by means of thermal power from the heating system. Any other means, like melting by the sun or wind, is neglected. Also, the algorithm isn't capable of distinguishing between ice and snow but treats it as one medium.

Once inside the algorithmic loop, the de-icing algorithm is commenced by evaluating the current surface conditions through combining the parameters of the prior timestep with the current weather parameters. Rule-based approaches are made to determine which one of five simulation modes is entered (see Figure 5). If there is snowfall in combination with a surface temperature < 0 °C or if there was a build-up of snow in the prior timestep, the algorithm will proceed in simulation modes 1, 2 or 3 which cover the case of an existing layer of snow on the surface element (left side of diagram). In that case the snow-free area ratio is set to << 1 to account for the insulating effect of the snow layer when solving the thermal balance. Before entering simulation modes 1, 2 or 3 the total available heating power $\dot{Q}_0$ is determined based on the prior timestep $(i-1)$. In case $\dot{Q}_0 < 0$ the ground cannot supply any heat to the surface and the thermal balance in equation (3) is solved for the surface temperature while the thermal load $\dot{Q}$ is zero. This state is covered by **simulation mode 1** which could be coined the "Siberian Mode" because the ground temperature will most likely be below 0 °C for this case to apply. On the contrary if $\dot{Q}_0 \geq 0$ a temperature

difference exists and thus a usable heat flux is available. In that case a thin layer of meltwater should form under the snow layer and the surface temperature is set to 0 °C. However, this approach can only be seen as an approximation because the surface loss mechanisms convection, radiation and evaporation – given cold enough weather conditions – can eat up the available power, leading to further cooling of the surface below 0 °C. To determine whether that is the case the available melting power $\dot{Q}_{melt}$ is calculated by subtracting the loss mechanisms from the total available power $\dot{Q}_0$, both evaluated for a surface temperature of 0 °C. For the case of $\dot{Q}_{melt} < 0$ the surface loss mechanisms would in fact cool the surface temperature to below 0 °C. In that case the algorithm proceeds with **simulation mode 2** where the thermal balance (equation (2)) is solved for the surface load. For instances of timesteps with $\dot{Q}_{melt} \geq 0$ the thermal power available in the ground is enough to actually melt away part of the existing snow layer (or all of it). This case applies to **simulation mode 3** which is also the only mode that doesn't necessitate solving the thermal balance as the thermal load is set to $\dot{Q}_0$ and the volume of melted snow is approximated using the melting heat flux $\dot{Q}_{melt}$ and the solid-to-liquid phase change enthalpy $\Delta h_{Ph,s\rightarrow l}$ of snow/ice.

For the timesteps where the algorithm doesn't detect a build-up of snow or a pre-existing snow layer simulation modes 4 & 5 become active and the assumption of instantaneous snowmelt (within the timestep) applies. For an adequately dimensioned heating system these two simulation modes ought to be the most common by far because the system should in fact be able to melt all the snow that falls within one timestep. While the melting power for modes 1-3 is estimated using the available temperature difference in the ground, modes 4 & 5 identify the melting power directly by calculating the latent and sensible heat fluxes using the snowfall rate. The snow-free area ratio is now = 1 for there is no insulating snow layer. **Simulation mode 4** solves the full thermal balance in equation (2) and the thermal heat flux $\dot{Q}$ is calculated. In case $\dot{Q} < 0$ **simulation mode 5** is entered and overwrites the heat flux to $\dot{Q} \coloneqq 0$. The simplified thermal balance in equation (3) is solved to yield the surface temperature. Simulation mode 5 can be seen as the "summer mode" because the surface temperature surpasses the ground temperature and the ground is thermally inactive.

Simulation modes 1-5 are followed by an update of the snow and water balances and additional thermal losses in connecting pipes between boreholes and heating elements and at the underside of the heating element are estimated using the calculated load $\dot{Q}$. The latter is then propagated back into the ground where it serves as an input to the ground model's g-function which in turn updates the borehole wall temperature for the next timestep. Finally, the surface temperature is updated using R_{th} and the new borehole wall temperature for cases where the thermal balance $F(\dot{Q})$ was solved (simulation modes 2 & 4).

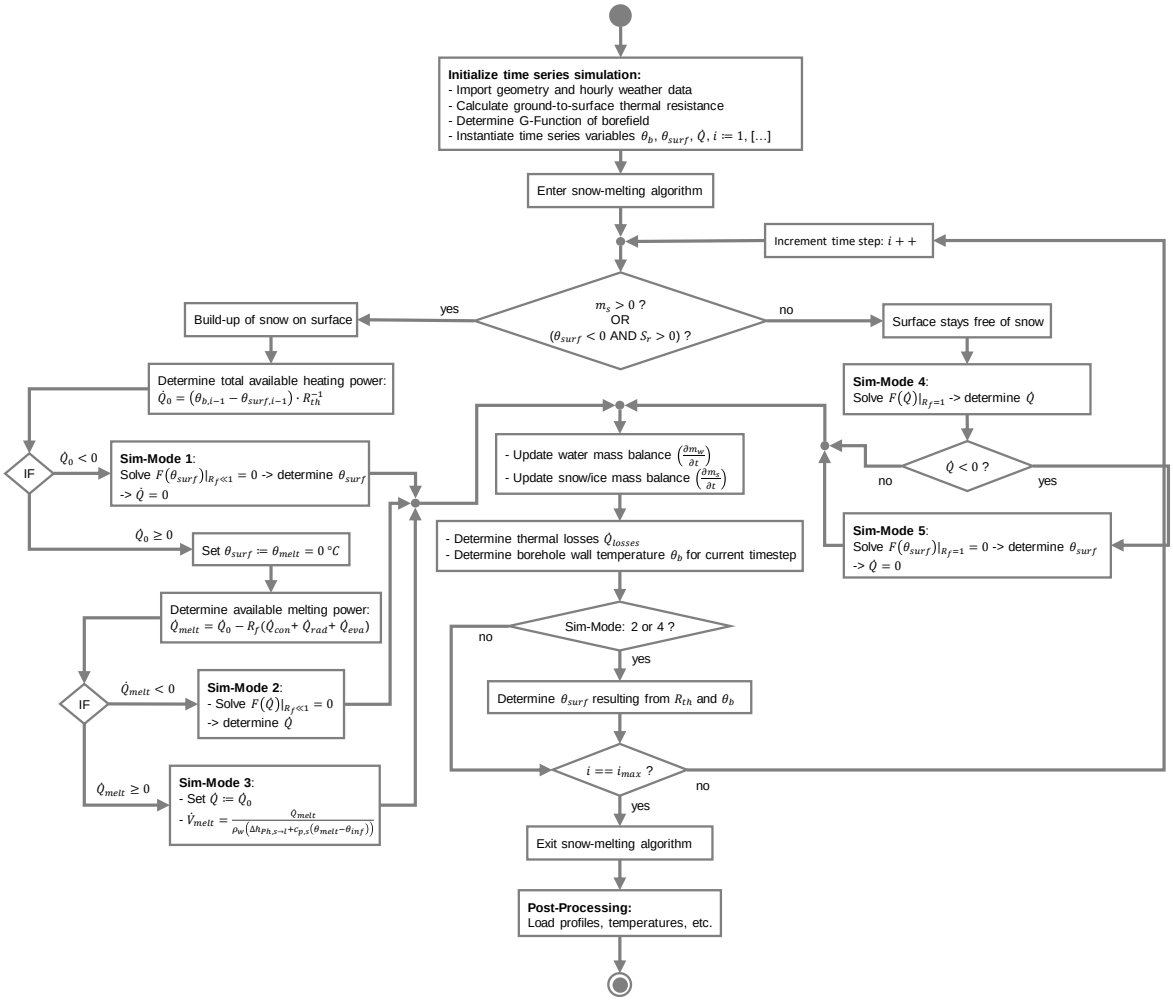


Figure 5: GERDPy snow-melting algorithm

The following table shall serve as a summary of the snow-melting algorithm and especially its five simulation modes, delineating if a net positive thermal heat flux from the ground is present as well as which one of the two power balances (see equation (2) and (3)) is solved, if any. In addition, information is provided whether the five heat transport mechanisms latent and sensible heat for snowmelt, convection, radiation and evaporation are active, e.g. for simulation mode 1 latent and sensible heat flux for melting snow is deactivated.

Table 2: Summary of simulation modes 1-5

Simulation mode	$\dot{Q}$ extracted from ground	thermal power balance	heat transfer mechanisms active				
			lat	sen	con	rad	eva
1	set to 0	$F(\theta_{surf})_{\lvert R_f \ll 1} = 0$	-	-	✓	✓	✓
2	> 0	$F(\dot{Q})_{\lvert R_f \ll 1} = 0$	-	-	✓	✓	✓
3	> 0	-	✓	✓	✓	✓	✓
4	> 0	$F(\dot{Q})_{\lvert R_f := 1} = 0$	✓	✓	✓	✓	✓
5	set to 0	$F(\theta_{surf})_{\lvert R_f := 1} = 0$	-	-	✓	✓	✓

2.2 Detailed physical model description

The structure of the GERDPy-tool was established on a modular and algorithmic basis in chapter 2.1. Now a detailed description of all the physical equations underlying the software tool shall be presented. Chapter 2.2.1 deals with the basics of the ground model which is based on the calculation of thermal response factors, so called g-functions, to predict the ground temperatures in a geothermal borefield of multiple boreholes. Chapter 2.2.2 addresses the borehole itself which – in case of the GERDI-system – can be interpreted as the interface between the ground (heat source) and the surface (heat sink). In contrast to what the name “borehole model” suggests, this sub-model actually is the system thermal resistance in itself, thus also including the thermal resistance of the heating element which is not part of the borehole. Finally, in chapter 2.2, the equations that make up the surface model, which is described in detail in chapter 2.1.2 (now-melting algorithm), are provided.

It is important to mention that while the equations within the three sub-models are evaluated hourly, only the thermal capacity of the ground – the ground’s ability to store heat – is actually modelled. The physics of the other parts of the system, from boreholes via thermosiphons to the heating element are realized in steady state, meaning there is no change in their stored energy levels. This was done due to reasons of computational efficiency. Also, the thermal capacity of the ground outweighs the combined thermal capacity of the other parts by multiple orders of magnitude, rendering their ability to store heat negligible in comparison to the ground[3].

2.2.1 Ground model – G-functions for fields of boreholes

GERDPy’s ground model is, as mentioned before, based on an open-source toolbox *Pygfunction* to generate thermal response factors, so-called g-functions, of the ground for geothermal borehole systems. The *Pygfunction* library is designed to generate g-functions for geothermal borefields of arbitrary layout with vertical boreholes in homogenous isotropic ground without groundwater advection.

2.2.1.1 G-functions

G-functions are widely established within energy simulation software to evaluate the performance of geothermal systems. The concept of g-functions was introduced in the mid-1980s by Per Eskilson[4]. A common use case of this mathematical construct is to calculate the temperature drop at the borehole wall due to a constant heat flux. According to the following formal mathematical definition, g-functions give the relation between the total heat extraction rate in the borefield and the average temperature variation at the borehole walls. In equation (4) $\bar{Q}$ is the average heat extraction rate per borehole length and g represents

[3] Considering a ground volume with a radius of just 5 m around a borehole and a borehole radius of 0,1 m, the ratio of thermal capacities ground:borehole is 2500:1 (given identical heat capacities c_p).

[4] For more information please refer to (Eskilson 1986) and (Eskilson 1987).

the g-function. T_b is the borehole wall temperature common to all boreholes. Ground parameters carry an index "g": T_g, undisturbed ground temperature; λ_g, thermal conductivity.

$$T_b = T_g - \frac{\bar{Q}}{2\pi\lambda_g} \cdot g\left(\frac{t}{t_s}, \frac{r_b}{H}, \frac{B}{H}, \frac{D}{H}\right) \tag{4}$$

It is important to note that g-functions depend on the geometric configuration of the boreholes, their lengths and diameters and their layout and are thus unique to a borefield. Each g-function curve is associated with a specific set of four non-dimensional parameters: t/t_s the non-dimensional time[5], r_b/H the non-dimensional borehole radius, B/H the borefield aspect or spacing ratio and D/H the non-dimensional buried depth of the boreholes. Any change in these parameters requires a re-evaluation of the g-function. (Cimmino & Bernier 2014) (Cimmino 2018b)

G-functions present themselves in the form of non-dimensional curves. The left plot in Figure 6 represents g-functions for a borefield of 3 x 2 boreholes, with different curves for different borehole aspect ratios (spacing to depth ratios). The y-axis represents the non-dimensional value of the g-function while the x-axis displays the non-dimensional time. The bottom curve corresponds to infinite borehole spacing which equates to a single borehole. One can identify four distinctive regions with unique heat transfer characteristics of the g-functions: (I) corresponds to a short time of heat extraction with radial 1-D heat transfer when axial heat conduction is negligible. In this region the temperature disturbance hasn't reached far into the ground yet. Interaction with neighbouring boreholes, the surface or the far-field below the boreholes is non-existent. Region (II) sees the detachment of the curves as the boreholes start to interact with each other thermally. As the bottom curve only represents one borehole, it doesn't exhibit this behaviour and flattens out upon reaching steady state behaviour. The smaller the aspect ratio, the earlier detachment happens and the steeper the g-function curve becomes. Region (III) represents radial-axial 2-D heat transfer while region (IV) sees the nearing of the curves towards steady state operation. Given enough time the borefield will always reach an equilibrium with the surrounding ground.

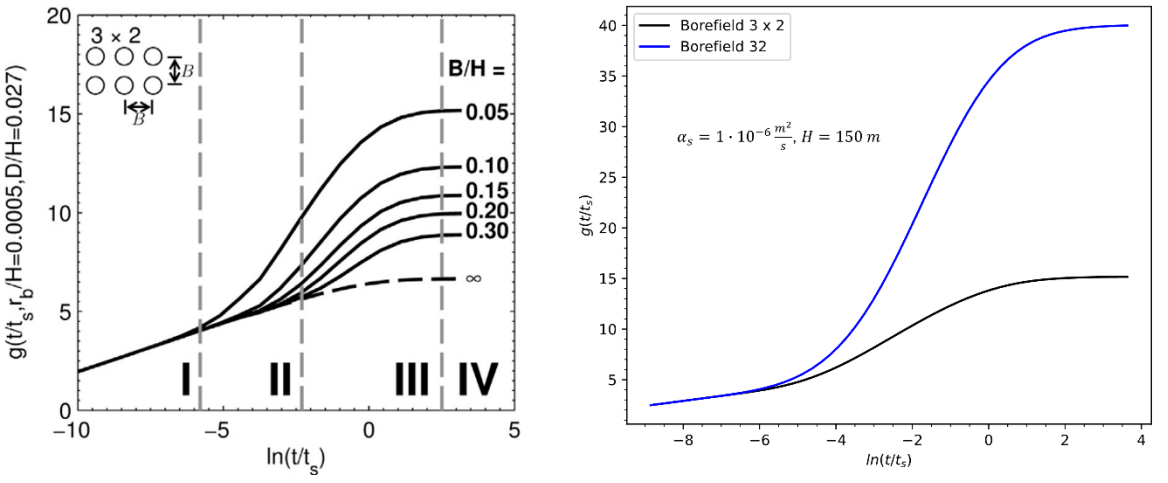


Figure 6: G-function curves for different borefield aspect ratios (Cimmino & Bernier 2014)

[5] where $t_s = \frac{H^2}{9a_g}$ is the characteristic time of the borefield and a_g is the ground thermal diffusivity

Region (IV) translates to a constant g-function value which means that the borehole wall temperature starts to stabilize to a constant value (see equation (4)). On the right, the blue curve, the g-function of a borefield of 32 boreholes detaches from the lower curve at $\ln(t/t_s) \approx -6$ which equates to around 72 days in "normal" time. It is interesting to note that the g-function value is the temperature drop itself if the heat extracted is $\bar{Q} = 2\pi\lambda_g$.

Eskilson's g-functions – which are often used as a benchmark for other g-function schemes – were obtained numerically using an explicit finite difference method with individually modelled boreholes within a radial-axial 2-D grid. This means a dedicated code was necessary to implement different borefield configurations. Nowadays, databases exist in common energy software tools with a large number of g-functions for pre-set borefield configurations using Eskilson's numerical g-functions. While this provides the advantage of avoiding g-function calculations, the borefield layout is limited to standard configurations, thus limiting design choices. Analytical methods have the advantage of allowing the calculation of irregular arbitrary borefield configurations at relatively small calculation times. The three main analytical methods include the infinite line source (ILS) solution, the cylindrical heat source (CHS) solution and the finite line source (FLS) solution. The ILS gives the temperature distribution around a line source of infinite length resulting from a constant heat extraction. This represents an early approach and only models 1-D radial heat transfer. The CHS solution is particularly useful when modelling the borehole wall temperature response at shorter timesteps when the borehole interior cannot be approximated by a line source. The finite line source (FLS) solution gives the temperature distribution around a line source of finite length H buried at a distance D below the ground. Here, the temperature distribution of another finite line source that is the original line source mirrored at the surface is superposed to create a uniform temperature boundary condition for the surface. GERDPy utilizes an FLS-approach which is presented below. (Cimmino 2018b) (Cimmino & Bernier 2014) (Cimmino et al. 2013)

2.2.1.2 Boundary conditions

To provide an impression of how heat flux and temperature can behave in relation to each other at the borehole wall, (Cimmino & Bernier 2014) provide a figure (shown below) to demonstrate the most common borehole wall boundary conditions used for the generation of g-functions. As can be seen in Figure 7, either the heat flux or the borehole wall temperature is kept uniform along the borehole wall and/or equal to all boreholes. In the context of the GERDPy-tool the boundary condition on the right (III) of equal and uniform borehole wall temperature was chosen. Firstly, because it corresponds to the original modelling approach pursued by Eskilson and secondly, because the fact that the thermosiphons

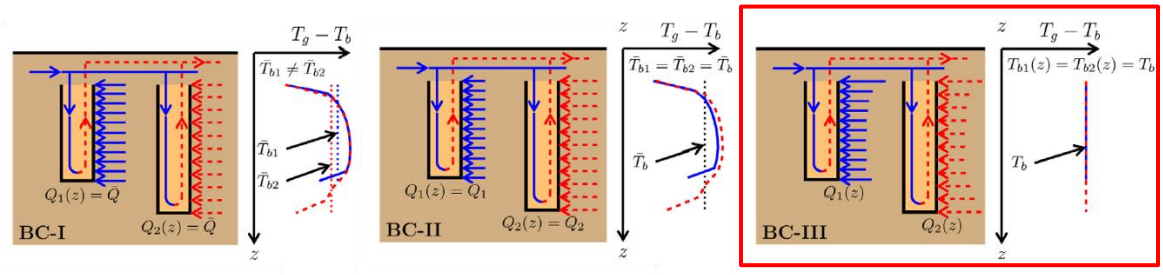


Figure 7: Three boundary conditions for the evaluation of g-functions (final choice: BC III): BC-I: uniform & equal heat extraction rates; BC-II: uniform & unequal heat extraction rates; BC-III: uniform & equal borehole wall temperature (Cimmino & Bernier 2014)

inside the borehole work nearly isothermally lends itself to using the boundary condition of uniform wall temperatures. The high thermal conductivity of the thermosiphon material (usually copper or steel) only supports this approach as axial gradients are levelled. This inevitably leads to a non-uniform heat extraction rate along the borehole depth. In reality, both the heat flux and temperature vary along the length of a borehole. The boundary conditions just constitute approximations that facilitate modelling approaches. Additionally, it shall be noted that all three boundary conditions have in common that $\bar{Q}$, the average heat extraction per unit borehole length of the entire borefield, has to be constant to comply with the g-function definition in equation (4).

2.2.1.3 Finite line source solution

Now that the concept of g-functions and different approaches to solutions is established in general, the mathematical formulation of the analytic FLS solution used within GERDPy shall be presented. The solution was created by (Cimmino & Bernier 2014) based on prior work by (Claesson & Javed 2011) and constitutes a generalization of (Cimmino et al. 2013).

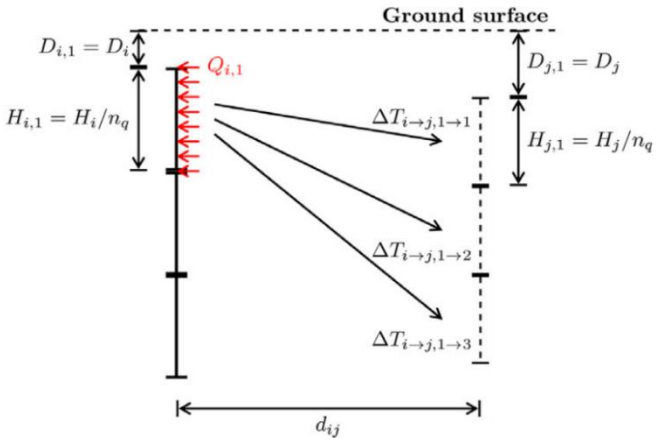


Figure 8: Two interacting boreholes modelled with three stacked finite line source segments (Cimmino & Bernier 2014)

Each borehole is divided into segments of finite line sources stacked on top of each other (see Figure 8). The number of segments per borehole is identical despite possibly differing lengths. The solution below models the temperature drop at the v-th segment of the n-th borehole caused by a heat flux $\dot{Q}_{m,u}$ extracted at the u-th segment of the m-th borehole. The variable *s* is the result of a substitution in the integral limits and shall not be further discussed. The solution shall now be presented without further explanation. The detailed derivation can be looked up in (Cimmino & Bernier 2014). The used variables are identical to the ones introduced in this work. “ierf()” is the integral error function. d_{mn} represents the horizontal distance between borehole m and n. To achieve a good trade-off between accuracy of the solution and computational speed, a value of 12 segments per borehole was deemed ideal by the authors. This number of segments n_q was thus chosen for GERDPy.

$$\Delta T_{m\to n,u\to v}(t) = -\frac{\dot{Q}_{m,u}}{2\pi\lambda_g} \cdot h_{m\to n,u\to v}(t) \tag{5}$$

$$
\begin{aligned}
h_{m\to n,u\to v}(t) = \frac{1}{2H_{n,v}} \int_{1/\sqrt{4a_g t}}^{\infty} \frac{1}{s^2} \exp(-d_{mn}^2 s^2) \Big[& ierf\left(\left(D_{n,v} - D_{m,u} + H_{n,v}\right)s\right) \\
& - ierf\left(\left(D_{n,v} - D_{m,u}\right)s\right) + ierf\left(\left(D_{n,v} - D_{m,u} - H_{m,u}\right)s\right) \\
& - ierf\left(\left(D_{n,v} - D_{m,u} + H_{n,v} - H_{m,u}\right)s\right) + ierf\left(\left(D_{n,v} + D_{m,u} + H_{n,v}\right)s\right) \\
& - ierf\left(\left(D_{n,v} + D_{m,u}\right)s\right) + ierf\left(\left(D_{n,v} + D_{m,u} + H_{m,u}\right)s\right) \\
& - ierf\left(\left(D_{n,v} + D_{m,u} + H_{n,v} + H_{m,u}\right)s\right) \Big] ds
\end{aligned} \tag{6}
$$

$$
d_{mn} = \begin{cases} r_b : for\ m = n \\ \sqrt{(x_m - x_n)^2 + (y_m - y_n)^2} : for\ m \neq n \end{cases} \tag{7}
$$

Equation (5) and (6) provide the solution for the temperature drop of one segment of a borehole due to a constant heat extraction rate from another borehole segment. To account for a time variation in the heat extraction rate, equation (6) is superposed in time (temporal superposition of k timesteps). $\dot{Q}_{m,u}(t_i)$ denotes the heat extraction rate increment in timestep i while the latter is also $\Delta t_i = t_i - t_{i-1}$. For the first timestep $\dot{Q}_{m,u}(t_{i=0}) = 0$ applies.

$$
\Delta T_{m\to n,u\to v}(t_k) = -\sum_{i=1}^{k} \frac{\dot{Q}_{m,u}(t_i)}{2\pi\lambda_g} \cdot h_{m\to n,u\to v}(t_k - t_{i-1}) \tag{8}
$$

To yield the total temperature variation at a borehole segment the temperature drops inflicted on segment v of borehole n by all segments of all boreholes have to be summed up (spatial superposition). n_b represents the total number of boreholes in the field. The resulting equation reads as follows (equation (9)). It can be evaluated for any segment v of any borehole n and forms a set of $n_b n_q$ equations with $2n_b n_q$ unknowns ($\dot{Q}_{m,u}$ and $\Delta T_{b,n,v}$).

$$
\Delta T_{b,n,v}(t_k) = -\sum_{i=1}^{k} \sum_{m=1}^{n_b} \sum_{u=1}^{n_q} \frac{\dot{Q}_{m,u}(t_i)}{2\pi\lambda_g} \cdot h_{m\to n,u\to v}(t_k - t_{i-1}) \tag{9}
$$

To further explain the g-function generation process would exceed the scope of this manual. The equations above shall mainly serve as a primer to understand the concept of the ground model of GERDPy. Basically, the g-function is obtained by assembling three sets of equations:

1. the FLS solution superposed in time and space (see above)
2. a global energy balance (constant total heat extraction rate $\bar{Q}$ in the borefield) and
3. the boundary condition at the borehole walls (see Figure 7).

1. - 3. form a complete system of equations that is converted to non-dimensional form and solved in the spectral domain using Laplace-transformation and a numerical FFT algorithm. The solution gives the normalized heat extraction rate increments $\dot{Q}_{m,u}/\bar{Q}$ and the non-dimensional temperature drop Θ_b at the borehole wall corresponding to the thermal response factor – the g-function – of the borefield.

Conclusively, it shall be noted that the FLS-solution constitutes an approximation of a borehole as a line, like the name implies. Because a borehole extends radially and has a

borehole radius, to model the borehole as a line creates a significant error for short simulation times when a temperature disturbance from a heat pulse has not yet reached the borehole wall. One way to ameliorate this circumstance would be to model the thermal capacity of the borehole, adding a delay time to the ground model heat flux which was not implemented for GERDPy to reduce model complexity, thus subjecting the model to this error. Eskilson defined a time criterion to address this issue and to validate his numerical model using line source solutions, stating that the error between his g-function (that corresponds to a cylindrical, more realistic model) and the ILS-solution falls below 10 % for simulation times $t > t_{10}$ (see equation (10)). For a borehole with a radius of $r_b = 0{,}1\ m$ and a ground thermal diffusivity of $1 \cdot 10^{-6}\ \mathrm{m}^2/s$ this would yield a $t_{10} = 13{,}\overline{8}\ h$. Different authors provide different measures of comparison between long-term and short-term g-functions. The suggestion for GERDPy-users that can be made at this point is to avoid short simulation times. Ideally, the minimum simulation time should be at least one to two days to avoid this issue.

$$t_{10} = \frac{5 r_b^2}{a_g} \tag{10}$$

For further information on the basic derivations of the FLS-solution and the methodology of *Pygfunction* the following literature can be referred to:

- (Claesson & Javed 2011) for the FLS-solution;
- (Claesson & Javed 2012) for the load aggregation technique for time series simulations and
- (Cimmino 2018a) for the method of deriving the g-function through similar geometric sub-groups within a borefield. Both techniques shall not be further discussed in this manual.

2.2.1.4 GERDPy-parameters

As a chapter summary, the following table provides the parameters a GERDPy-user can manipulate that influence the ground model:

Table 3: Ground model parameters

a_g	ground thermal diffusivity [m²/s]
r_b	borehole radius [m]
θ_g	undisturbed ground temperature [°C]
λ_g	ground thermal conductivity [W/mK]
(x, y)	coordinates of boreholes within a borefield [m]
H	borehole length/depth [m]
D	borehole buried depth [m]
t_{sim}	simulation time [s]

2.2.2 Borehole model – The system thermal resistance

The borehole model constitutes all thermal resistances j that are modelled within GERDPy and connects them in series. Given a heat extraction rate from the ground, the

borehole wall temperature is connected to the surface temperature via this thermal resistance in terms of a temperature drop from ground to surface expressed in [K/W]. The basic equation is given below (equation (11)). For reasons of computational speed the thermal resistance was implemented as non-dependent from the heat extraction rate: $R_{th} \neq R_{th}(\dot{Q})$, made possible by simplifying the thermosiphon model (see corresponding subchapter).

$$\dot{Q} = \frac{(\theta_b - \theta_{surf})}{\sum_j R_{th,j}} \tag{11}$$

The following four resistances are addressed and their series connection further illustrated in Figure 9. The red colour symbolizes the heat source (ground), the blue colour the heat sink (surface).

1. ground-to-grout thermal contact resistance ($R_{th,c}$)
2. borehole thermal resistance ($R_{th,b}$)
3. thermosiphon/heat pipe thermal resistance ($R_{th,hp}$)
4. heating element thermal resistance ($R_{th,he}$)

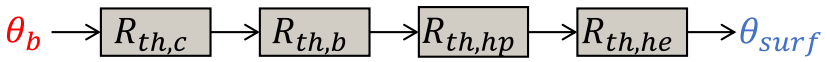


Figure 9: Series connection of thermal resistances within the borehole model

2.2.2.1 Ground-to-grout thermal contact resistance

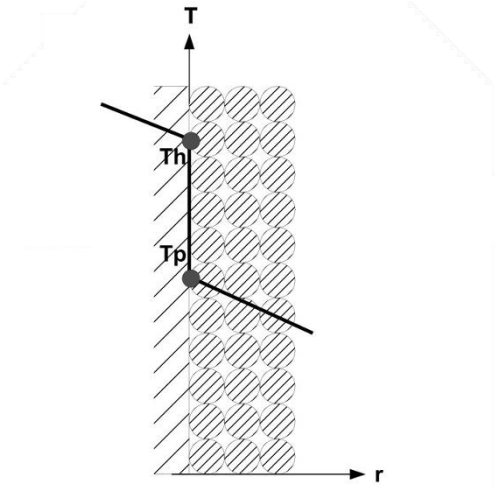


Figure 10: Contact resistance between ground and backfill material

Thermal contact resistances are generally not easy to model because the ground conditions in the contact region between two solids are usually unknown. Nevertheless, the main thermal contact resistance of the system, from the ground to the borehole grout, was modelled for GERDPy. For this a correlation from (VDI 2013) to calculate the thermal resistance from a solid wall to a backfill made of spherical particles was utilized, modelling the ground as solid and the backfill as the spherical particles. The contact zone described here is shown

in Figure 10. The temperature discontinuity constitutes the thermal contact resistance in the borehole wall. The indices used in the figure shall be ignored. The higher temperature level on the “ground” side is in fact the borehole wall temperature designated T_b in this document.

The correlation models two heat transfer mechanisms happening simultaneously in the gas-filled gaps: heat conduction as α_{con} and radiative heat transfer as α_{rad}. Both phenomena happen within the gas-filled interstitial space. φ denotes the contact area ratio between wall and particles. The basic equation for the combined heat transfer coefficient is as follows:

$$\alpha_{WB} = \varphi \cdot \alpha_{con} + \alpha_{rad} \tag{12}$$

For heat conduction through the gaseous medium equation (13) is used. It models the conductive heat transfer coefficient with the thermal conductivity of the gas λ_a and the free path length of the gas molecules within the air pockets l_a. Furthermore, δ and d are used to describe the particle surface roughness and their diameter, respectively.

$$\alpha_{con} = \frac{4\lambda_a}{d}\left(\left(1 + \frac{2(l_a + \delta)}{d}\right)\ln\left(1 + \frac{d}{2(l_a + \delta)}\right) - 1\right) \tag{13}$$

To derive the free molecule path length l_a (see equation (15)) the coefficient of accommodation γ [0,1] (see equation (14)) is necessary which describes impulse propagation within the medium. The contact zone temperature, factor for the type of gas, universal gas constant, molar mass of the gas, gas pressure and specific isobaric heat capacity of the gas are described by T, C, R, M, p, and $c_{p,a}$, respectively.

$$\gamma = \left(10^{\left(0{,}6 - \frac{\frac{1000K}{T}+1}{C}\right)} + 1\right)^{-1} \tag{14}$$

$$l_a = 2 \cdot \frac{2-\gamma}{\gamma}\sqrt{\frac{2\pi RT}{M}}\frac{\lambda_a}{p\left(2c_{p,a} - \frac{R}{M}\right)} \tag{15}$$

The radiative heat transfer coefficient is described by equation (16). C_{WB} describes the ratio of emission coefficients of wall and backfill whereas σ and ϵ are the Stefan-Boltzmann constant and emissivity, respectively.

$$\alpha_{rad} = 4 \cdot C_{WB} \cdot T^3 \tag{16}$$

$$C_{WB} = \frac{\sigma}{\frac{1}{\epsilon_W} + \frac{1}{\epsilon_B} - 1} \tag{17}$$

Conclusively, α_{WB} from equation (12) is multiplied by the circumference of a borehole and inverted to form a thermal resistance per unit length of borehole depth [mK/W] (equation (18)). The latter is now divided by the total borehole length in the borefield to yield the total thermal contact resistance $R_{th,c}$ (equation (19)).

$$r_{th,c} = \frac{1}{2r_b\pi \cdot \alpha_{WB}} \tag{18}$$

$$R_{th,c} = \frac{r_{th,c}}{H_{total}} \tag{19}$$

The values for the relevant parameters to calculate the ground-to-grout thermal resistance for GERDPy are pre-set within the software and their manipulation by the user is not intended as they already constitute a reasonable parametrization for the problem. The following enumeration is a list of chosen values for the parameters that can be checked in (VDI 2013): $\varphi \coloneqq 0{,}8$; $\lambda_a \coloneqq 0{,}025\ W/mK$; $d \coloneqq 10^{-3}\ m$; $\delta \coloneqq 250 \cdot 10^{-6}\ m$; $C \coloneqq 2{,}8$; $R = 8{,}314\ J/molK$; $M \coloneqq 0{,}02896\ kg/mol$; $T \coloneqq 283\ K$; $\epsilon_{W/B} \coloneqq 0{,}2$; $\sigma = 5{,}67 \cdot 10^{-8}\ W/m^2K^4$; $p \coloneqq 1 \cdot 10^5\ Pa$; $c_{p,a} \coloneqq 1007\ J/kgK$.

2.2.2.2 Borehole thermal resistance

The borehole thermal resistance usually constitutes the major thermal resistance within borehole heat exchanger systems and constitutes the resistance of all entities within the borehole, from the borehole wall through the grout, possible insulation of the thermosiphons, the thermosiphon material itself and heat transfer onto the heat transfer fluid. All components are ideally connected thermally and not subject to a contact resistance. The convective heat transfer from the thermosiphon pipe material onto the fluid was neglected in this part as all processes within the thermosiphons are already covered by the simplified thermosiphon model in the following chapter. An illustration of the general layout within the borehole is depicted in Figure 11. As can be seen, $N = 4$ identical thermosiphons are uniformly distributed on a circle with radius r_w through their epicentres.

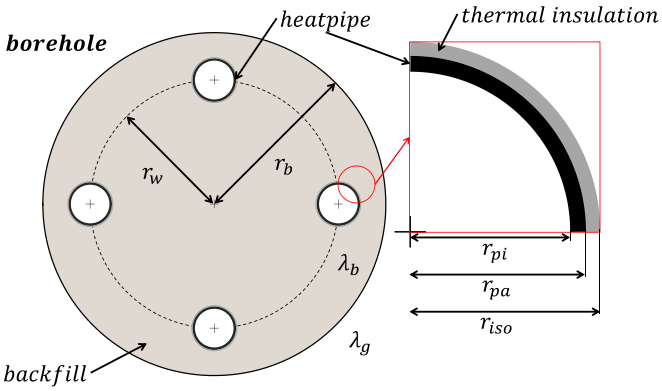


Figure 11: Cross-section of a borehole

The GERDPy-model uses a line source approximation by (Hellström 1991) to model the borehole thermal resistance. Detailed derivations can be read in the reference. The model makes a simplifying assumption of 2-D heat conduction from the borehole wall to the thermosiphons (axial heat conduction is neglected) and thus fits well with the ground model boundary condition of constant and uniform borehole wall temperatures. In addition, the grout/backfill material as well as the surrounding ground are modelled as homogenous and isotropic.

To derive the borehole thermal resistance a symmetric[6] matrix of interaction coefficients R^0_{mn} of dimension N x N is struck up. It describes the thermal interaction of each line sink with the other line sinks ($m \neq n$) and with the borehole wall ($m = n$). By superposing the thermal flux of the m-th line sink and its corresponding thermal interaction coefficient r^0_{mn} a general mathematical solution for the temperature difference between the grout-side borehole wall and the thermosiphon fluid can be expressed in matrix notation as follows. It shall be mentioned that the coefficients r^0_{mn} carry the unit of a thermal resistance but don't represent a physical quantity in and of themselves.

$$\Delta T = \boldsymbol{R^0_{mn}} \cdot \boldsymbol{\dot{q}^T_m}; m = 1 \dots N, n = 1 \dots N \tag{20}$$

The coefficients within R^0_{mn} are to be calculated as follows. "m" denotes the row of the matrix whereas "n" denotes the column. Coefficients r^0_{mm} make up the main diagonal of the matrix ($m = n$), r^0_{mn} make up the rest ($m \neq n$).

$$r^0_{mm} = \frac{1}{2\pi\lambda_b}\left(\ln\left(\frac{r_b}{r_{pa}}\right) - \sigma_\lambda \ln(1 - b_m^2)\right) + r_{th,pm} \tag{21}$$

$$r^0_{mn} = -\frac{1}{2\pi\lambda_b}(\ln(b_{mn}) - \sigma_\lambda \ln(b'_{mn})) \tag{22}$$

b_m, b_{mn} and b'_{mn} are coordinate-dependent geometric coefficients that contain positional information of the line sinks (=thermosiphon centres) as (x, y)-coordinates. These coordinates are determined implicitly within the tool and need not be defined by the GERDPy-user. σ_λ denotes the ratio of thermal conductivities of the grout/backfill and the ground[7]. An additional term added to r^0_{mm} is necessary to account for the thermal resistance of a possible insulation layer and the thermosiphon material itself. It is added in series and is the basic Péclet-equation (Polifke & Kopitz 2013) for radial heat transfer through cylinders, the term is denoted $r_{th,pm}$.

$$b_m = \frac{\sqrt{x_m^2 + y_m^2}}{r_b} \tag{23}$$

$$b_{mn} = \frac{\sqrt{(x_n - x_m)^2 + (y_n - y_m)^2}}{r_b} \tag{24}$$

$$b'_{mn} = \sqrt{(1 - b_m^2)(1 - b_n^2) + b_{mn}^2} \tag{25}$$

$$\sigma_\lambda = \frac{\lambda_b - \lambda_g}{\lambda_b + \lambda_g} \tag{26}$$

[6] $r^0_{mn} = r^0_{nm}$ (matrix symmetry)

[7] The solution used is derived from an exact solution of the steady-state temperature distribution T(x, y) inside and outside of the borehole and thus includes the thermal conductivity λ_g of the ground outside the borehole. This means that λ_g has an effect on $R_{th,b}$ (!)

$$r_{th,pm} = \frac{\ln\left(\frac{r_{iso,b}}{r_{pa}}\right)}{2\pi\lambda_{iso}} + \frac{\ln\left(\frac{r_{pa}}{r_{pi}}\right)}{2\pi\lambda_{p}} \tag{27}$$

To derive the specific and total borehole thermal resistance, the entries of the inverse matrix of R^0_{mn} are summed up and inverted to yield the specific thermal borehole resistance $r_{th,b}$, provided as [mK/W]. Analogous to the ground-to-grout contact resistance a division by the total borehole length yields the total borehole thermal resistance $R_{th,b}$.

$$r_{th,b} = \frac{1}{\sum_{n=1}^{N}\sum_{m=1}^{N}(r^0_{mn})^{-1}} \tag{28}$$

$$R_{th,b} = \frac{r_{th,b}}{H_{total}} \tag{29}$$

2.2.2.3 Thermosiphon thermal resistance

The thermosiphon thermal resistance covers the temperature difference created by the flow of heat transfer fluid inside the closed volume of the thermosiphon, between the evaporator in the ground and the condenser in the heating element. As previously mentioned, the thermosiphon thermal resistance is in reality a function of the thermal heat flux itself because of flow phenomena inside the thermosiphon pipe. A full analytical model would include complex correlations for convective heat transfer, condensation and evaporation at the condenser and evaporator, respectively, and other two-phase flow phenomena and highly transient behaviour like the entrainment limit. While this resistance is the most complex to model analytically, it also has the smallest impact on the overall thermal resistance of the system R_{th} because thermosiphons/heat pipes are by design conveyors of heat at low temperature differences. Due to this circumstance the thermosiphon thermal resistance within GERDPy is a very simple fixed value for the temperature drop in [K] caused by a thermal heat flux in [W]. The value (see equation (30)) is based on experimental data acquired in similar projects. N and n_b are the number of thermosiphons per borehole and the number of boreholes in the borefield, respectively.

$$R_{th,hp} \neq R_{th,hp}(\dot{Q}) = \frac{1}{500\ [W] \cdot N \cdot n_b} \tag{30}$$

2.2.2.4 Heating element thermal resistance

The heating element thermal resistance constitutes the last of the four thermal resistances considered within the GERDPy-tool. It covers the heat transfer path from inside the thermosiphon pipes in the condenser to the heating element surface. As the thermosiphons follow the borehole inside the ground towards the surface, they leave the confines of the borehole upon arriving at the surface. There, they are bundled into a connection line that connects one or multiple boreholes with a heating element which they feed into horizontally at a uniform horizontal distance s_R. Thus, they become the condenser pipes. The thermosiphons' dimensions stay the same as in the borehole, without the thermal insulation.

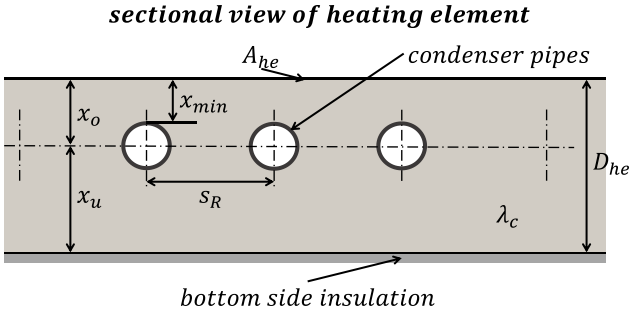


Figure 12: Cross-section of heating element

The implementation in GERDPy uses an exact analytical solution of the 2-D Fourier heat equation (see equation (31)) used for the calculation of heat losses in heated walls with parallel pipe registers which can be found in (VDI-2055-1). The following graphic depicts variable parameters and the geometric configuration (see Figure 12). The model only covers parallel piping inside the heating element as opposed to the illustration on the cover page that suggests orthogonal piping.

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0 \tag{31}$$

The following equations have been derived from the original, more complex solution in (VDI-2055-1) by eliminating superfluous parameters to meet the requirements of the GERDI-system. Still, some additional parameters are necessary to stabilize the iterative component “sum” of the solution shown in equation (33) which yields the thermal heat flux per metre of condenser piping inside the heating element. Some of those parameters have been pre-set to certain values within GERDPy (see below). Because the original equation considers a heat flux on both sides of a wall or floor element, but GERDPy only considers the upward heat flux for thermal resistance modelling, the corresponding heat transfer coefficients are set to 0 for the bottom ($\alpha_u \to 0\, W/m^2$) and infinite for the heating element surface ($\alpha_o \to \infty\, W/m^2K$) to emulate a temperature boundary condition. An additional parameter is set to $s_c \coloneqq 0\, m$ for a lack of additional layers in the heating element. Equation (33) is evaluated for a temperature difference of $\Delta T \coloneqq 1\, K$ and multiplied with the total length of condenser piping in the heating element to yield the heating element thermal resistance $R_{th,he}$ in equation (32). The κ and $\tilde{\kappa}$ constitute heat transmission coefficients with indices “o” and “u” for the upper and lower half of the heating element. x_o and x_u are the respective halves of the heating element corresponding to upward and downward heat conduction (see centre line in Figure 12 for clarity). To eliminate downward heat conduction it is necessary to set $x_u \to \infty\, m$ for an exact mathematical solution. The contact resistance between the condenser pipes and the concrete of the heating element is modelled by adding a thin layer of air of $\lambda_a = 0{,}0262\, W/mK$ and thickness of 0,1 mm between the two, so that $d_a \coloneqq d_{pa} + 0{,}0002\, m$.

$$R_{th,he} = \left(l_{p,he} \cdot \dot{Q}_l\right)^{-1} \tag{32}$$

$$\dot{Q}_l = 2\pi\lambda_c \frac{\Delta T}{\frac{\lambda_c}{\lambda_p} ln\left(\frac{d_{pa}}{d_{pi}}\right) + \frac{\lambda_c}{\lambda_a} ln\left(\frac{d_a}{d_{pa}}\right) + ln\left(\frac{s_R}{\pi d_a}\right) + \frac{2\pi\lambda_c}{s_R(\tilde{\kappa}_o + \tilde{\kappa}_u)} + sum} \tag{33}$$

$$\kappa_o = \left(\frac{1}{\alpha_o} + \frac{s_c}{\lambda_c}\right)^{-1};\ \kappa_u = \left(\frac{1}{\alpha_u} + \frac{s_c}{\lambda_c}\right)^{-1} \quad \text{(34) \& (35)}$$

$$\tilde{\kappa}_o = \left(\kappa_o^{-1} + \frac{x_o}{\lambda_c}\right)^{-1};\ \tilde{\kappa}_u = \left(\kappa_u^{-1} + \frac{x_u}{\lambda_c}\right)^{-1} \quad \text{(36) \& (37)}$$

Finally, to yield a stable and exact mathematical solution, the sum-term is evaluated iteratively. The following table shows the algorithm of the series solution:

Table 4: Iterative series solution for sum-term in equation (33)

1)	$\beta_o = \frac{\kappa_o s_R}{\lambda_c}$	(38)
	$\beta_u = \frac{\kappa_u s_R}{\lambda_c}$	(39)
2)	Set $j \coloneqq 1$, $sum \coloneqq 0$	
3)	$N_1 = 1 - \frac{\beta_u + 2\pi j}{\beta_u - 2\pi j} e^{\frac{4\pi j}{s_R} s_c}$	(40)
4)	$N_2 = 1 - \frac{\beta_u - 2\pi j}{\beta_u + 2\pi j} e^{-\frac{4\pi j}{s_R} s_c}$	(41)
5)	$\gamma = \frac{\beta_o - 2\pi j}{\beta_o + 2\pi j} e^{-\frac{4\pi j}{s_R}(x_o + x_u)}$	(42)
6)	$e_o = \frac{\left(\lambda_c + \frac{\lambda_c}{N_1} - \frac{\lambda_c}{N_2}\right)\left(e^{-\frac{4\pi j x_u}{s_R}} - \gamma\right)}{\lambda_c(1+\gamma) + \left(\frac{\lambda_c}{N_2} - \frac{\lambda_c}{N_1}\right)(1-\gamma)}$	(43)
7)	$e_u = -\frac{\beta_o - 2\pi j}{\beta_o + 2\pi j} e^{-\frac{(4\pi j x_o)}{s_R}}(1 + e_o)$	(44)
8)	Set $sum_0 \coloneqq sum$	
9)	$sum = sum_0 + \frac{e_o(j) + e_u(j)}{j}$	(45)
10)	If $abs(sum - sum_0) > \varepsilon$, set $j \coloneqq j + 1$, jump to 3)	
11)	$sum = \sum_{j=1}^{\infty} \frac{e_o(j) + e_u(j)}{j}$	(46)

2.2.2.5 GERDPy-parameters

As a chapter summary, the following table provides the parameters a GERDPy-user can manipulate that influence the borehole model:

Table 5: Borehole model parameters

λ_g	ground thermal conductivity [W/mK]
H_{total}	total borehole length/depth of the borefield [m]
r_b	borehole radius [m]
n_b	number of boreholes per borefield [-]
N	number of thermosiphons/heat pipes per borehole [-]
r_w	radius of thermosiphon epicentres [m]
$r_{iso,b}$	outer radius of thermosiphon insulation inside borehole [m]
r_{pa}	outer radius of thermosiphon pipe [m]
r_{pi}	inner radius of thermosiphon pipe [m]
λ_b	thermal conductivity of borehole grout/backfill [W/mK]
λ_{iso}	thermal conductivity of insulation layer [W/mK]
λ_p	thermal conductivity of heat pipe/thermosiphon material [W/mK]
A_{he}	heating element surface area [m²]
x_{min}	minimum vertical pipe-to-surface distance [m]
λ_c	thermal conductivity of heating element material (concrete) [W/mK]
s_R	horizontal centre-line distance between condenser pipes [m]
$l_{p,he}$	total length of condenser piping within heating element [m]
D_{he}	vertical thickness of heating element [m]

2.2.3 Surface model – Determining the thermal load

The algorithmic inner workings of the surface model have been described in detail in chapter 2.1.2. The present chapter focusses on the physical equations behind each of the heat flux terms in equation (1). Also, the solver for the thermal balance equations in (2) and (3) will be covered in detail as well as the mass balancing for snow and ice on the heating element surface. To recap, the purpose of the surface model is to couple the ground (heat source) and the ambient weather conditions at the surface (heat sink) in order to determine the thermal heat flux/heat extraction rate of the system: the larger the temperature difference between ground and surface, the larger the heat flux per timestep. Because the system is modelled in steady state, the ground heat flux equates to the sum of all surface heat transfer mechanisms (the whole system is in a state of equilibrium).

2.2.3.1 Weather data

The GERDPy-tool makes use of weather data files that are imported by the user via the GUI. Information about how this weather data file needs to be formatted is given in chapter 3. The user is required to work with generic year-long weather data files of hourly averages to design a GERDI-system as those are more representative of a certain location[8]. For multi-year simulations a single weather data file of 8760 entries for each hour of the

[8] Such weather data can be found on the website of the DWD: https://opendata.dwd.de/climate_environment/

year is looped. The following weather parameters are necessary to describe the surface conditions:

- u_{inf}: ambient wind speed [m/s]
- θ_{inf}: ambient air temperature [°C]
- S_r: snowfall rate [mm/h]
- RR: precipitation rate [mm/h]
- B: cloudiness [0,8] – 0: clear skies, 8: fully overcast
- φ: relative humidity [%]

Of these parameters u_{inf}, θ_{inf}, RR, B and φ are original weather parameters while S_r is derived from the precipitation rate: precipitation is defined as snowfall if the ambient temperature at the time of precipitation $\theta_{inf} < 1\,°C$. Otherwise, RR constitutes rainfall. This also means that S_r is always a subset of RR. The unit for both kinds of precipitation is [mm] of water equivalent per hour.

2.2.3.2 Material and substance values

The material and substance values used within the surface model of GERDPy can be found in pertinent literature like (ASHRAE 2013) or (VDI 2013). A list of used parameters and their values is as follows. Possible temperature, pressure, etc. dependencies of the fixed values have been neglected for ease of computation and/or because of the low impact on results. The values apply to conditions of $p = 1\,bar$ and $T = 0\,°C$.

- $\rho_w = 999{,}84\,kg/m^3$
- $\Delta h_{Ph,s\to l} = 333 \cdot 10^3\,J/kg$
- $\Delta h_{Ph,l\to v} = 2500{,}9 \cdot 10^3\,J/kg$
- $\theta_{melt} = 0\,°C$
- $\rho_a = 1{,}276\,kg/m^3$
- $\lambda_a = 0{,}0244\,W/mK$
- $\nu_a = 13{,}5 \cdot 10^{-6}\,m^2/s$
- $c_{p,s} = 2106\,J/kgK$
- $c_{p,w} = 4219\,J/kgK$
- $c_{p,a} = 1006\,J/kgK$

To calculate the humidity ratios for the evaporation heat flux (see chapter 2.2) the dew point temperature is necessary. It is determined within GERDPy using the thermodynamic library CoolProp[9].

[9] http://www.coolprop.org/

2.2.3.3 Surface heat transfer mechanisms

This sub-chapter deals with the five surface heat transfer mechanisms introduced in chapter 2.1.2. Convection, radiation and evaporation represent the three surface loss mechanisms while latent heat and sensible heat are the two contributors to melting of snow/ice. For each heat transfer mechanism two basic equations are used within the software, one for the thermal balance in equation (2), and one for the thermal balance in equation (3). To recap, the first is used for the case of a net positive heat flux from the ground and is solved explicitly for $\dot{Q}$ while the latter is used when there is no heat extraction from the ground and solved explicitly for the surface temperature. The power terms are provided as heat flux per area, $\dot{q}$. They represent the variant fed into equation (3). For the case of equation (2) the surface temperature is substituted using the borehole wall temperature from the prior timestep as follows: $\theta_{surf} \coloneqq \theta_{b,0} - \dot{q} \cdot R_{th}$, so that $\dot{q}(\theta_{surf})$ becomes $\dot{q}(\theta_{b,0} - \dot{q} \cdot R_{th})$.

Convection

Convective heat transfer on the heating element surface constitutes one of the three surface loss mechanisms. It describes heating or cooling of the surface through forced convection by airflow. The basic mechanism of convection can be described by Newton's law of cooling which states that the heat flux through convection is directly proportional to the temperature difference of the streaming fluid and the solid that is warmed/cooled and a heat transfer coefficient α. The complexity of convection modelling lies in the determination of the latter which is usually dependent on a multitude of geometric, hydraulic and thermal conditions. To use a complex approach using Nusselt-correlations, information on the wind direction and exact dimensions of the heating element would be necessary which is not the case. For the case of GERDPy a very simple empirical model established for flat surface heating elements was used (Bentz Dale P. 2000). It models the heat transfer coefficient as solely dependent on the wind speed:

$$\dot{q}_{con} = \alpha_{con}(u_{inf}^{*}) \cdot (\theta_{surf} - \theta_{inf}) \tag{47}$$

The heat transfer coefficient α_{con} is determined by means of two simple cases for a lower and higher wind speed. u_{inf}^{*} denotes a wind-shear-corrected wind speed.

$$\alpha_{con} = 5{,}6 + 4 \cdot u_{inf} \text{: for } u_{inf} \leq 5\,\frac{m}{s} \tag{48}$$

$$\alpha_{con} = 7{,}2 \cdot u_{inf}^{0,78} \text{: for } u_{inf} > 5\,\frac{m}{s} \tag{49}$$

The wind-shear-corrected wind speed is used to reduce the wind speed provided by the weather file as weather data is usually gathered from small weather stations at an elevation of about 10 m above ground. For convection modelling, on the other hand, the wind speed near the heating element surface is needed. A reference height of 1 m and a topographic roughness level of 0,005 m are chosen. The equation can be looked up in literature on atmospheric physics and is applicable within 100 m of elevation above ground. The low topographic roughness level applies to an open area without houses or trees and therefore might overestimate the wind speed for areas with nearby obstacles.

$$u_{inf}^{*} = u_{inf} \frac{\lg_{10}(z = 1\,m) - \lg_{10}(z = 0{,}005\ m)}{\lg_{10}(z = 10\,m) - \lg_{10}(z = 0{,}005\ m)} \tag{50}$$

The following graphic shows a GERDPy-simulation using only convection. The weather data used is from Hamburg, Germany. During summertime the surroundings warm up the heating element and the heat extracted from the ground is zero.

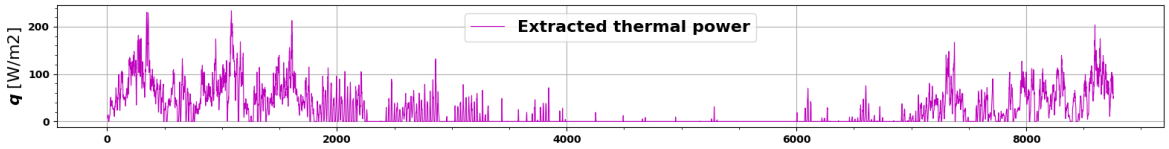


Figure 13: Year-long convection simulation using weather data of Hamburg, Germany

Radiation

The radiation model used for GERDPy can be found in (ASHRAE 2015), in the chapter for “Snow melting and freeze protection”. It should be stressed that direct radiation from the sun is not considered as the weather data intended for the software tool usually doesn’t contain data on direct radiation. Instead, the sky is modelled as a blackbody with a temperature T_{MR} which stands for “mean radiant temperature”. The wavelength mostly falls into the regime of infrared radiation. The radiative exchange between the surface and the surroundings is modelled as non-obstructed by any possible objects. σ is the Stefan-Boltzmann constant. The emission coefficient used for a concrete surface is set to $\epsilon_{surf} \coloneqq 0{,}94$.

$$\dot{q}_{rad} = \sigma \cdot \epsilon_{surf} \cdot \left(T_{surf}^{4} - T_{MR}^{4}\right) \tag{51}$$

The amount of cloud cover has a strong influence on the mean radiant temperature of the surroundings T_{MR}. Under snowfall conditions, the surroundings are approximated to assume the ambient air temperature. Without snowfall the approximation is more complex (see equation below) and a weighted average between the temperature of the sky and the temperature of the clouds is calculated based on the amount of cloudiness.

$$T_{MR} = \begin{cases} T_{inf} : for\ S_r > 0 \\ \left(T_{cloud}^{4} \cdot \frac{B}{8} + T_{cs}^{4} \cdot \left(1 - \frac{B}{8}\right)\right)^{\frac{1}{4}} : for\ S_r = 0 \end{cases} \tag{52}$$

The equivalent blackbody temperature of the clear sky, T_{cs}, is a function of the ambient air temperature and the humidity and provided in the form of a well-established curve fit.

$$T_{cs} = T_{inf} - \left(1{,}1058 \cdot 10^{3} - 7{,}562 \cdot T_{inf} + 1{,}333 \cdot 10^{-2} \cdot T_{inf}^{2} - 31{,}292 \cdot \varphi + 14{,}58 \cdot \varphi^{2}\right) \tag{53}$$

The temperature of the clouds, T_{cloud}, is determined for an elevation of the clouds at 3000 m using the ambient air temperature and an average atmospheric lapse rate. In case $T_{cs} > T_{cloud}$, T_{cloud} is set equal to the clear sky temperature T_{cs}.

$$T_{cloud} = T_{inf} - 19{,}2\ K \tag{54}$$

The following graphic shows a GERDPy-simulation using only radiative exchange of the heating element surface and the surroundings. One can see that radiative cooling correlates with colder temperatures in winter but pretty much exists throughout the whole year. One can also observe that, compared to convection (see Figure 13), radiative losses are more evenly distributed over the year, peaking also in winter.

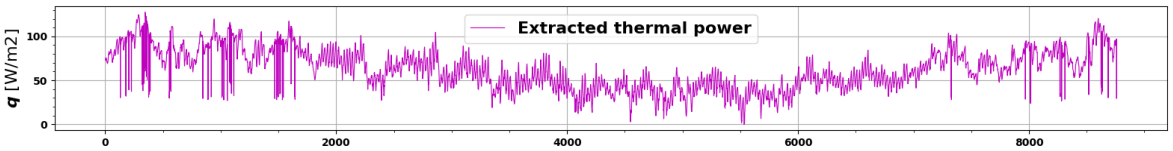


Figure 14: Year-long radiation simulation using weather data of Hamburg, Germany

Evaporation

According to (ASHRAE 2015), the heat flux required for evaporation of water from a wet surface is given in equation (55). Evaporation constitutes a mass transfer phenomenon, making use of a mass transfer coefficient β_m which is analogous to the heat transfer coefficient α for convection. $X_{sat,surf}$ and X_{inf} are the humidity ratio of saturated air at heating element surface temperature and the humidity ratio of ambient air, respectively. The first is evaluated for a saturation vapour pressure at surface temperature $p_v = p_s(T_{surf})$, assuming saturation at the surface, while the latter uses the dew point temperature at ambient conditions $p_v = p_s\left(T_{dew}(T_{inf}, \varphi)\right)$. (ASHRAE 2013)

$$\dot{q}_{eva} = \rho_a \cdot \beta_m \cdot \left(X_{sat,surf} - X_{inf}\right) \cdot \Delta h_{Ph,l\to v} \tag{55}$$

$$X = \frac{m_v}{m_a} = 0{,}622 \cdot \left(\frac{p_v}{p_{inf} - p_v}\right) \tag{56}$$

Equation (57) shows how the mass transfer coefficient is coupled with the heat transfer coefficient, as both phenomena are related in a sense that a bulk of external flow is charged or loaded at the interface to a convective surface, with material or with heat, respectively. The non-dimensional coefficients Schmidt-number and Prandtl-number are set to $Sc = 0{,}6$ and $Pr = 0{,}7$. Sc depends on the binary diffusion coefficient δ.

$$\beta_m = \left(\frac{Pr}{Sc}\right)^{2/3} \frac{\alpha_{con}}{\rho_a \cdot c_{p,a}} \tag{57}$$

$$Pr = \frac{\nu_a}{a};\ Sc = \frac{\nu_a}{\delta} \tag{58}$$

$$\delta = \frac{2{,}252}{p_{inf}} \cdot \left(\frac{T_{inf}}{273{,}15}\right)^{1{,}81} \tag{59}$$

It shall be noted that a liquid film of water on the heating element surface has to exist for evaporation to happen. Therefore, the algorithm deactivates the evaporation term if the surface is dry or the surface temperature drops below 0 °C (direct sublimation of ice was neglected for GERDPy (see Table 1)). Also, the reverse phenomenon to evaporation – condensation – on the heating element surface was neglected, rendering $\dot{q}_{eva} < 0\ W$ an impossibility.

The following figure shows what a simulation result considering only evaporation looks like. As with the other heat extraction terms the power flux is well distributed across the year but a discontinuous nature becomes evident. GERDPy stops evaporation from the surface if the condition of a wet surface doesn't apply. This is achieved through a constant evaluation of the surface temperature and water volume balancing on the surface. The latter is explained later in this document.

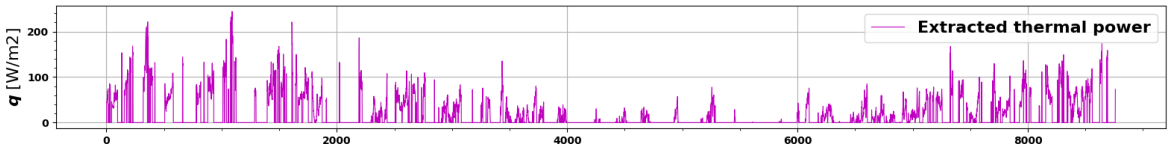


Figure 15: Year-long evaporation simulation using weather data of Hamburg, Germany

Latent & sensible heat

Latent and sensible heat are the only heat transfer components of the heat flux that actually directly contribute to melting of snow and ice. The latent and sensible heat fluxes are both directly proportional to the snowfall rate S_r for simulation cases in which the heating element surface stays above 0 °C (this should be the case in most situations). This corresponds to an instantaneous snowmelt (within the current timestep). In case of a build-up of snow/ice on the surface, the latent and sensible heat fluxes are diminished and a ΔT-approach is used (see chapter 2.1.2). The latent heat makes up the phase-change energy necessary to transform the snow/ice from solid into liquid form. The sensible heat component exists for the fact that the temperature level of the fallen snow is first raised to the melting point before the melting process and then up to the surface temperature in liquid form afterwards. The equations for the standard case are as follows:

$$\dot{q}_{lat} = \rho_w \cdot S_r \cdot \Delta h_{Ph,s\to l} \tag{60}$$

$$\dot{q}_{sen} = \rho_w \cdot S_r \left(c_{p,s} \cdot \left(\theta_{melt} - \theta_{inf} \right) + c_{p,w} \cdot \left(\theta_{surf} - \theta_{melt} \right) \right) \tag{61}$$

Latent and sensible energy make up only a few percent of the total extracted energy when looking at annual simulations. Actual snowmelt happens only seldom but the heat flux may be bigger than for the other mechanisms. The figure below shows a simulation done only considering latent and sensible heat for snow-melt. They was no build-up of snow on the surface for any timestep.

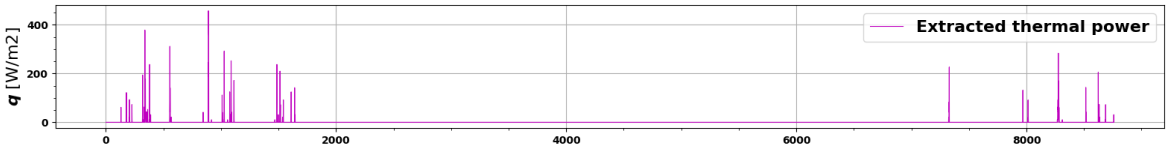


Figure 16: Year-long snowmelt (latent & sensible heat) simulation using weather data of Hamburg, Germany

2.2.3.4 Mass balancing

Mass balancing is necessary within GERDPy to model the amount of accumulated snow/ice and water (two balances).

Snow/ice volume balancing is used for all cases where fallen snow cannot be melted within the timestep. As this should not be the case for most times, the snow mass m_s is mostly zero. The value of m_s therefore is an indicator for the current activate simulation mode (see chapter 2.1.2). The balance is updated after every timestep i using the mass of the prior timestep $(i-1)$, the snowfall rate S_r and the latent heat power $\dot{Q}_{lat}$. The basic equation is:

$$m_{s,i} = m_{s,i-1} + (S_r \cdot \rho_w \cdot A_{he}) - \frac{\dot{Q}_{lat}}{\Delta h_{Ph,s\to l}} \tag{62}$$

The amount of water accumulated on the surface is an important constraint for the evaporation term and accounts for both kinds of precipitation, rain and snow. The latter will eventually become water that can be evaporated and is thus implicitly modelled into the equation using the precipitation rate RR. The maximum amount of water allowed corresponds to the water level at which the gravitational pull will cause the water to run off from the surface. This value is set to 2 mm of water. The basic equation is:

$$m_{w,i} = m_{w,i-1} + (RR \cdot \rho_w \cdot A_{he}) - \frac{\dot{Q}_{eva}}{\Delta h_{Ph,l\to v}} \tag{63}$$

2.2.3.5 Pressure correlations

The tool uses two kinds of pressure correlations that have been taken from (ASHRAE 2013). The first one yields the altitude-corrected ambient pressure (equation (64)). The second one is for the water vapour saturation pressure p_s, for both conditions over ice (-100 to 0 °C) and over liquid water (0 to 200 °C). Both terms are necessary to determine the evaporation power flux (see equation (55)).

$$p_{inf} = 101325 \cdot (1 - 2{,}25577 \cdot 10^{-5} \cdot z_{asl})^{5{,}2559} \tag{64}$$

$$\ln(p_s) = \begin{cases} \frac{C_1}{T} + C_2 + C_3 T + C_4 T^2 + C_5 T^3 + C_6 T^4 + C_7 \ln T : for\ T[-100;0] \\ \frac{C_8}{T} + C_9 + C_{10} T + C_{11} T^2 + C_{12} T^3 + C_{13} \ln T : for\ T[0;200] \end{cases} \tag{65}$$

The constants for the p_s-correlation are as follows: $C_1 = -5{,}6745359e3$, $C_2 = 6{,}3925247e0$, $C_3 = -9{,}6778430e-3$, $C_4 = 6{,}2215701e-7$, $C_5 = 2{,}0747825e-9$, $C_6 = -9{,}4840240e-13$, $C_7 = 4{,}1635019e0$, $C_8 = -5{,}8002206e3$, $C_9 = 1{,}3914993e0$, $C_{10} = -4{,}8640239e-2$, $C_{11} = 4{,}1764768e-5$, $C_{12} = -1{,}4452093e-8$, $C_{13} = 6{,}5459673e0$.

2.2.3.6 Thermal losses

A distinction needs to be made between the heat transport phenomena convection, radiation and evaporation that are treated as surface losses and additional thermal losses that occur at the connecting pipes between borehole and heating element and at the underside of the heating element. The latter are the topic within this subchapter. It shall be added that these thermal losses are not included within the thermal balance on the surface but treated separately as a lump sum and added to the heat extraction rate $\dot{Q}$ later within the algorithm. The following equation gives the thermal loss term:

$$\dot{Q}_{losses} = \dot{Q}_{losses,conn} + \dot{Q}_{losses,u} \tag{66}$$

The losses of the connection line $\dot{Q}_{losses,con}$ are modelled by means of the Péclet-equation (Polifke & Kopitz 2013) for cylindrical shells (piping + insulation) by estimating the thermal losses through natural convection separately for every thermosiphon pipe inside the connection line. l_{conn} is the total length of connection piping between boreholes and heating element. The connection piping contains the separate thermosiphon pipes as a bundle. The $\alpha_{con,conn}(\Delta T)$ is approximated using a simple correlation by (Löser et al. 2018) for air around insulated pipes. ΔT is the difference between the gas temperature inside the piping (calculated via the chain of thermal resistances until that point before the heating element) and the ambient temperature.

$$\dot{Q}_{losses,con} = \Delta T \cdot \frac{2\pi(l_{conn} \cdot N)}{\frac{\ln\left(\frac{r_{pa}}{r_{pi}}\right)}{\lambda_p} + \frac{\ln\left(\frac{r_{iso,conn}}{r_{pa}}\right)}{\lambda_{iso}} + \frac{1}{\alpha_{con,conn}(\Delta T) \cdot r_{iso,conn}}} \tag{67}$$

$$\alpha_{con,conn}(\Delta T) = 9{,}4 + 0{,}052 \cdot \Delta T \tag{68}$$

$$\Delta T = \theta_{gas} - \theta_{inf} = \left(\theta_{b,0} - \dot{Q} \cdot \left(R_{th} - R_{th,he}\right)\right) - \theta_{inf} \tag{69}$$

$\dot{Q}_{losses,u}$ is the thermal loss term on the underside of the heating element. The modelling approach considers 2-D heat conduction from inside the condenser piping (gas temperature) to the underside of the heating element using the analytical solution for parallel pipe registers introduced in chapter 2.2.2.4 ($R_{th,he,u}$), 1-D heat conduction through the thermal insulation layer (middle term inside brackets) and finally a convective term to approximate unknown conditions on the underside using $\alpha_{con,he,u} = 10\ W/m^2K$ (Löser et al. 2018).

$$\dot{Q}_{losses,u} = \Delta T \cdot \left(R_{th,he,u} + \frac{1}{\lambda_{iso}} \cdot \frac{D_{iso,he}}{A_{he}} + \frac{1}{\alpha_{con,he,u} \cdot A_{he}}\right)^{-1} \tag{70}$$

$$\alpha_{con,he,u} = 10 \tag{71}$$

2.2.3.7 Solver

Finally, the solver that was written for the thermal balances on the surface is presented. Basically, the solver iteratively approaches the solution for a steady-state system. Figure 5 illustrates GERDPy's snow-melting algorithm. There, all instances where the thermal balance equation is solved are documented within the "Simulation modes 1-5". The two different thermal balances solved are equation (2) and (3). The first one is used for the case of a positive heat extraction rate, the latter in case there is no heat extraction from the ground.

It can be mathematically shown that the thermal balances F in equation (2) and (3) are fourth-degree polynomials[10] that – in the vicinity of the origin – are monotonically decreasing

[10] The 4th degree arises from the radiation term which contains T^4.

or monotonically rising, respectively. While they can have up to four zero-crossings, the one nearest to the origin represents the desired solution for $\dot{Q}$ or θ_{surf}, respectively.

The algorithm thus starts by setting the first value to zero. The algorithm then enters the iteration phase where the x-value is increased or decreased with a fixed step size until it overshoots the zero-crossing. Upon doing so, the algorithm reverses the running direction using a refined, smaller step size until it overshoots again, continuing the process until $abs(F) < res$, a pre-defined residual.

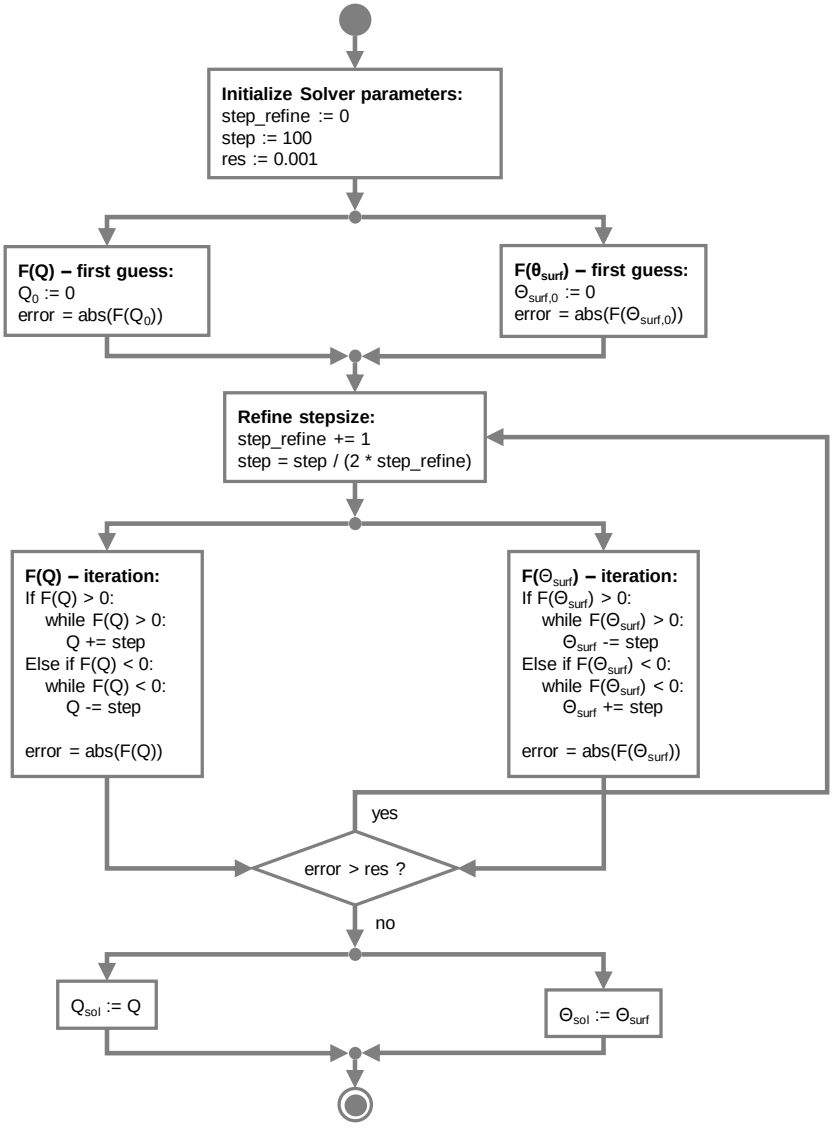


Figure 17: GERDPy-solver

2.2.3.8 GERDPy-parameters

As a chapter summary, the following table provides the parameters a GERDPy-user can manipulate that influence the surface model. The weather parameters and elevation over sea-level of the simulated location can only be indirectly influenced by changing the location.

Table 6: Surface model parameters

u_{inf}	ambient wind speed [m/s]
θ_{inf}	ambient temperature [°C]
S_r	snowfall rate [mm/h]
B	cloudiness [octal units]
φ	relative humidity [%]
RR	precipitation rate [mm/h]
z_{asl}	elevation above sea level [m]
l_{conn}	total length of connection line between boreholes and heating element [m]
N	number of thermosiphons/heat pipes per borehole [-]
$l_{p,he}$	total length of condenser piping within heating element [m]
r_{pa}	outer radius of thermosiphon pipe [m]
r_{pi}	inner radius of thermosiphon pipe [m]
$r_{iso,conn}$	outer radius of thermosiphon insulation inside connection line [m]
λ_p	thermal conductivity of heat pipe/thermosiphon material [W/mK]
λ_{iso}	thermal conductivity of insulation layer [W/mK]
$D_{iso,he}$	vertical thickness of insulation layer on heating element underside [m]
A_{he}	heating element surface area [m²]
λ_c	thermal conductivity of heating element material (concrete) [W/mK]
s_R	horizontal centre-line distance between condenser pipes [m]

3 Using GERDPy

3.1 Installing GERDPy

First download the GERDPy project from GitHub (LINK). For using GERDPy Simulation Tool, Python 3.7 must be used. If not installed, click the link below and install Python 3.7.

https://www.python.org/downloads/

Inside your preferred terminal navigate to the folder, where GERDPy is saved and run the commands below depending on your system, remembering before installing Python 3.7 and PySide6 "pip install PySide6".

Windows:

```
py -3.7 guimain.py
```

MacOS and Linux:

```
python3.7 guimain.py
```

3.2 Usage

The GERDPy simulation tool is operated via a graphical user interface (GUI). It starts on the tab “Home” as shown in Figure 18 below. Every tab and all input parameters will be described in the following chapters.

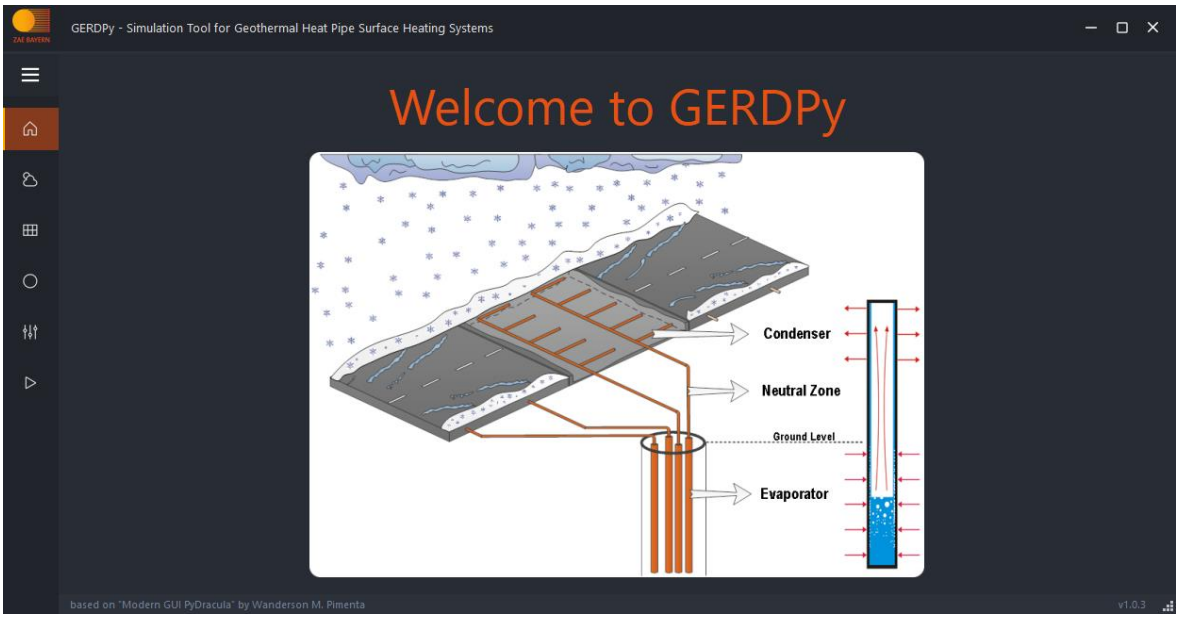


Figure 18: GERDPy GUI tab ‘Home’

3.2.1 Weather & local parameters

The tab “Weather & local parameters” is about defining the local conditions, where a heating system is to be simulated.

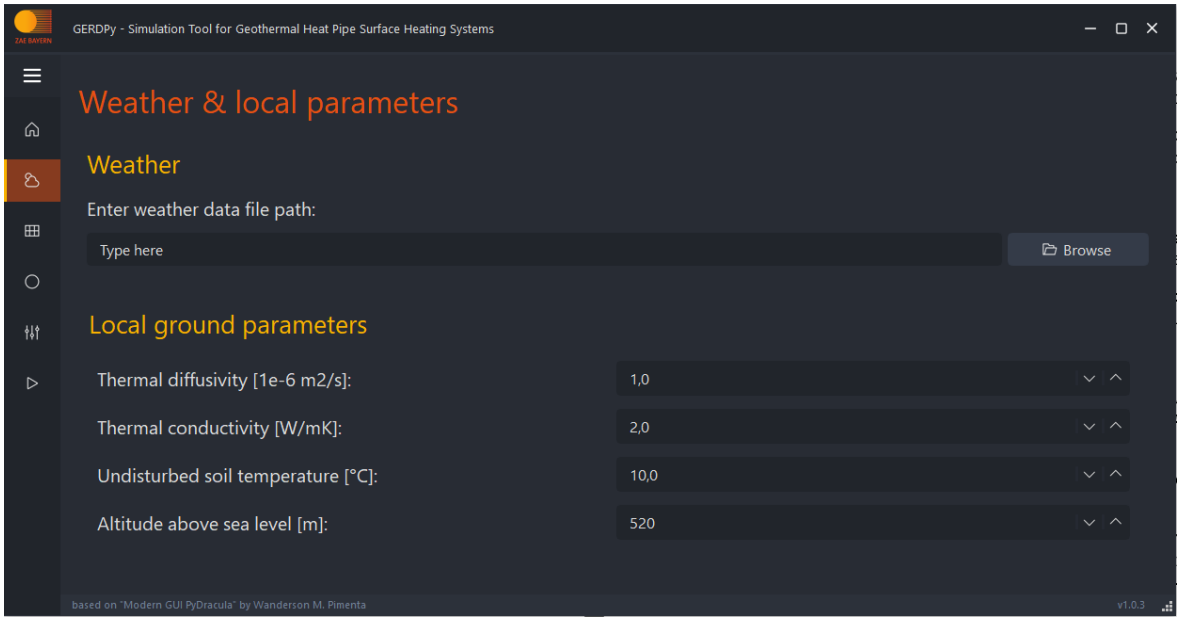


Figure 19: GERDPy GUI tab ‘Weather & local parameters’

For the weather data, a .xlsx-file is required. It is recommended to use test reference years of weather data at the desired location. The file must contain hourly data of the following variables in shown format and units in brackets.

- month [-]
- day [-]
- hour [-]
- precipitation [mm]
- temperature [°C]
- relative humidity [%]
- wind speed [m/s]
- cloudiness [octal units]

1	Month				Day	Hour	Precipitation	Temperature	Relative humidity	Wind speed	Cloudiness
2	Munich/Riem						[mm]	[°C]	[%]	[m/s]	[octal units]
3	48,133	11,7	529	1							
4											
5	m				dm	h	RR	Ta	RH	FF	N
6	1				1	1	0.0	6.3	85	2.5	5
7	1				1	2	0.0	6.3	86	2.0	5
8	1				1	3	0.0	6.2	84	1.6	5
9	1				1	4	0.0	6.2	89	2.0	5
10	1				1	5	0.0	6.2	86	1.9	5
11	1				1	6	0.0	6.3	86	2.3	5
12	1				1	7	0.0	6.3	83	1.9	5
13	1				1	8	0.0	6.4	79	2.5	5
14	1				1	9	0.0	6.9	77	2.2	5
15	1				1	10	0.0	8.8	70	4.0	5
16	1				1	11	0.0	10.5	59	3.3	0
17	1				1	12	0.0	11.8	54	3.0	0
18	1				1	13	0.0	11.3	56	3.5	0
19	1				1	14	0.0	11.3	54	4.0	0
20	1				1	15	0.0	11.3	54	3.5	5
21	1				1	16	0.0	11.3	56	4.7	5
22	1				1	17	0.0	11.6	55	3.5	5
23	1				1	18	0.0	11.1	56	4.0	5
24	1				1	19	0.0	10.6	56	3.3	5

Figure 20: Example of a weather data file, first 24 hours (Munich-Riem)

The structure of the file must be like shown in Figure 20. This is an example of a test reference year of weather data at the location Munich-Riem. For creating a test reference year of weather data it is recommended to use the software Meteonorm[11]. By clicking on the “Browse” button, one can search for the file and provide the path to said weather file. In case of multi-year simulations, the annual weather file is used multiple times within the software tool by means of concatenation.

The “local ground parameters” are related to the desired location, the “thermal diffusivity” and “thermal conductivity” depend on the natural composition of the ground. The “undisturbed soil temperature” is taken as the starting condition for the simulation. The “altitude above sea level” for the chosen location must also be specified.

[11] https://meteonorm.com/en/

3.2.2 Heating elements

The tab “Heating elements” is about the heating elements’ geometry.

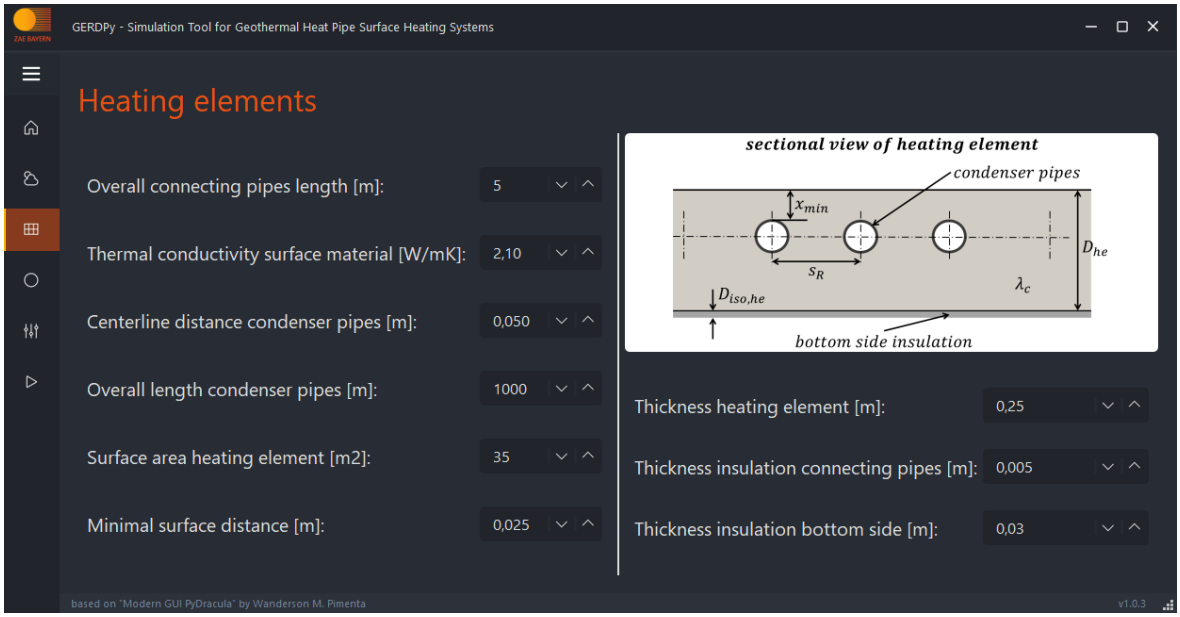


Figure 21: GERDPy GUI tab ‘Heating elements’

The “overall connecting pipes length” refers to l_{conn} and constitutes the overall length of all pipes connecting heating elements and boreholes with one connection line per borehole. These connection lines contain the thermosiphon pipes as a bundle. The “thermal conductivity of surface material” is by default the thermal conductivity of concrete which is the usual surface material. The “centreline distance” of condenser pipes refers to the distance between the condenser pipes measured from pipe centre to pipe centre within the heating element (s_R). The “minimal surface distance” is the shortest vertical distance from condenser pipes to the surface (x_{min}). The “overall length condenser pipes” refers to the total length of condenser piping in the heating element, denoted $l_{p,he}$ in this document.

3.2.3 Borehole parameters

The tab “Borehole parameters” encompasses geometric and thermal parameters within the bounds of the borehole.

By activating the radio button “Activate varying borehole depths”, the borehole depths will be taken from the .txt-file provided in tab “Borefield geometry & Simulation parameters” (see 3.2.4). If the radio button is not activated, the borehole depths will be all equal and set to the value provided in the field below the radio button. For having no insulation between heat pipes and backfill the insulation radius must be set equal to the outer radius of thermosiphon piping: $r_{iso} \coloneqq r_{pa}$.

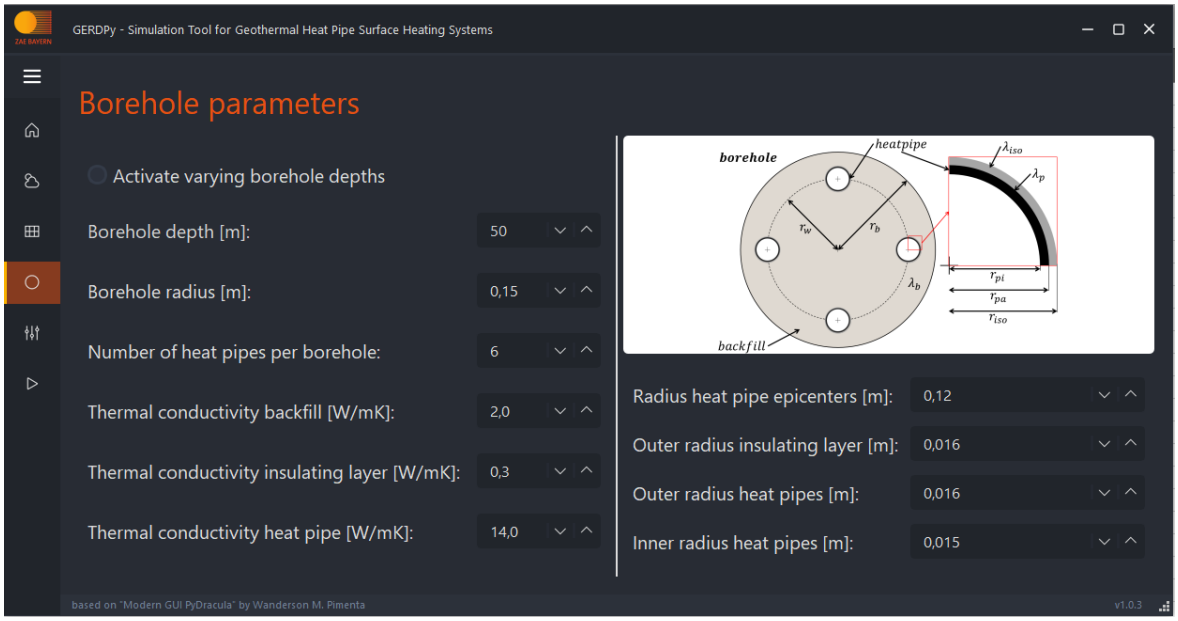


Figure 22: GERDPy GUI tab 'Borehole parameters'

3.2.4 Borefield geometry & Simulation parameters

The tab "Borefield geometry & Simulation parameters" is about the borefield layout and simulation duration.

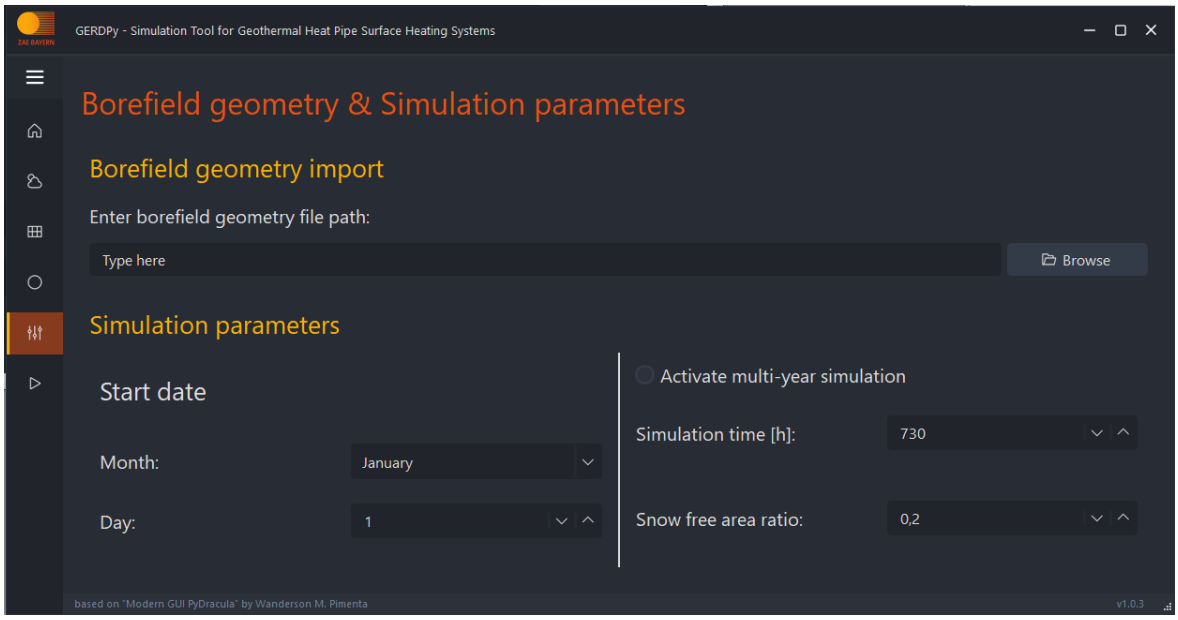


Figure 23: GERDPy GUI tab 'Borefield geometry & Simulation parameters'

The borefield geometry file has to be a .txt-file, tab-separated, with a header and structured as follows. The relative location of a borehole is given as (x, y)-coordinates, H gives the borehole length and D gives the buried depth of the respective borehole. For instance, a borehole with $D := 5\ m$ and $H := 70\ m$ will reach down to 75 metres. All parameters must be specified in metres.

```
# x     y       H       D
0.0     10.0    73      4.0
4.0     0.0     50      4.0
9.0     0.0     50      4.0
14.0    0.0     50      4.0
19.0    0.0     50      4.0
```

Figure 24: Example of a bore-field geometry file

If the radio button “Activate varying borehole depths” in tab “Borehole parameters” is checked, the column containing H will be ignored. It can also be omitted in the borefield geometry file.

The simulation time must be given in hours. If the radio button “Activate multi-year simulation” is checked, the unit changes to years. For multi-year simulations the minimum timespan is 2 years. For the start date of the simulation a month and a day can be chosen.

The snow free area ratio is a parameter for estimating the extent to which a possible build-up of snow will act as thermal insulation of the heating surface. Theoretical aspects of this parameter are covered in chapter 2.1.2, Figure 4 shall also be referred to. The value of the snow free area ratio must be specified between 0 and 1 and gives the estimated insulating effect of a snow cover. 0 means the snow cover works as a perfect thermal insulator, 1 means the snow cover has no insulating effect. This parameter largely depends on the extend of snow coverage in terms of area and the type of snow and is largely unknown prior to the simulation. It should therefore be interpreted as a means of covering different scenarios of snow coverage, e.g. a simulation may be run three times, with snow-free area ratios of $R_{f,1} \coloneqq 0$, $R_{f,2} \coloneqq 0{,}5$ and $R_{f,3} \coloneqq 1$.

3.2.5 Simulation

In the tab “Simulation”, the simulation can be started and results can be saved.

When starting the simulation, the button “START SIMULATION” will change to “ABORT SIMULATION”. If “ABORT SIMULATION” is pushed, the simulation will terminate and the GUI will close. This can take a few moments.

After the simulation has finished, the results can be saved as a .csv-file by pushing the button “SAVE RESULTS” (requires path and file name).

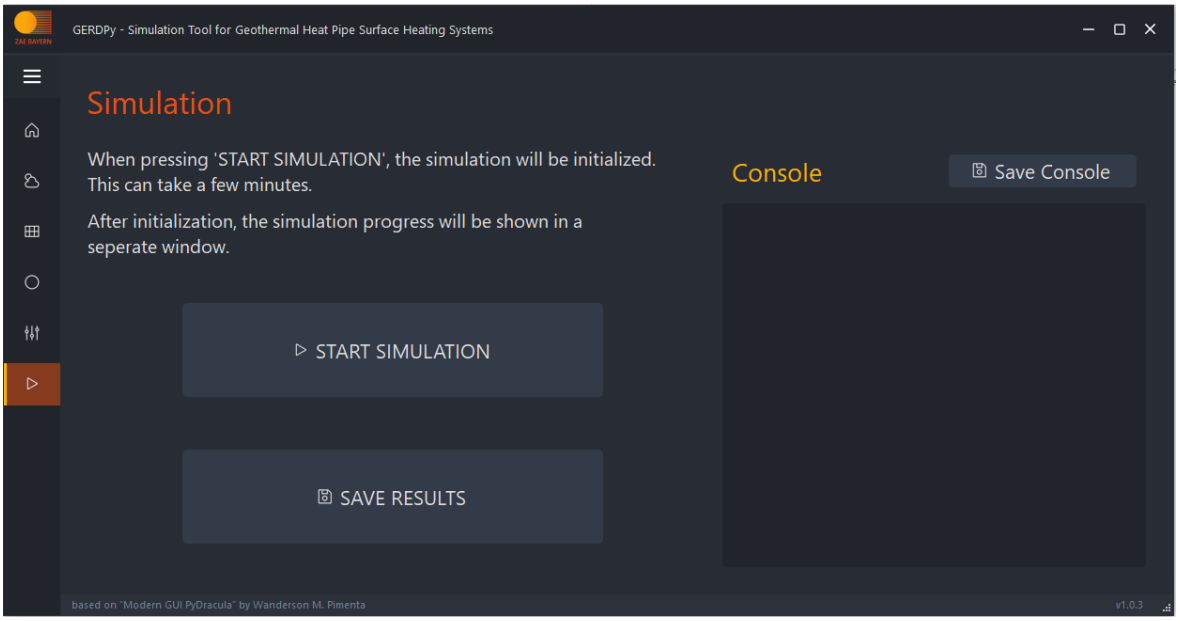


Figure 25: GERDPy GUI tab 'Simulation'

The console gives some information about the progress while simulating and the overall heat extraction from the ground and can be saved after completion as .txt-file by pushing the button "Save Console".

3.2.6 Output

After the simulation has finished, some figures and graphs will open. The first figure shows the borefield configuration as provided in the borefield geometry file in the "Borehole parameters" tab shown in Figure 26.

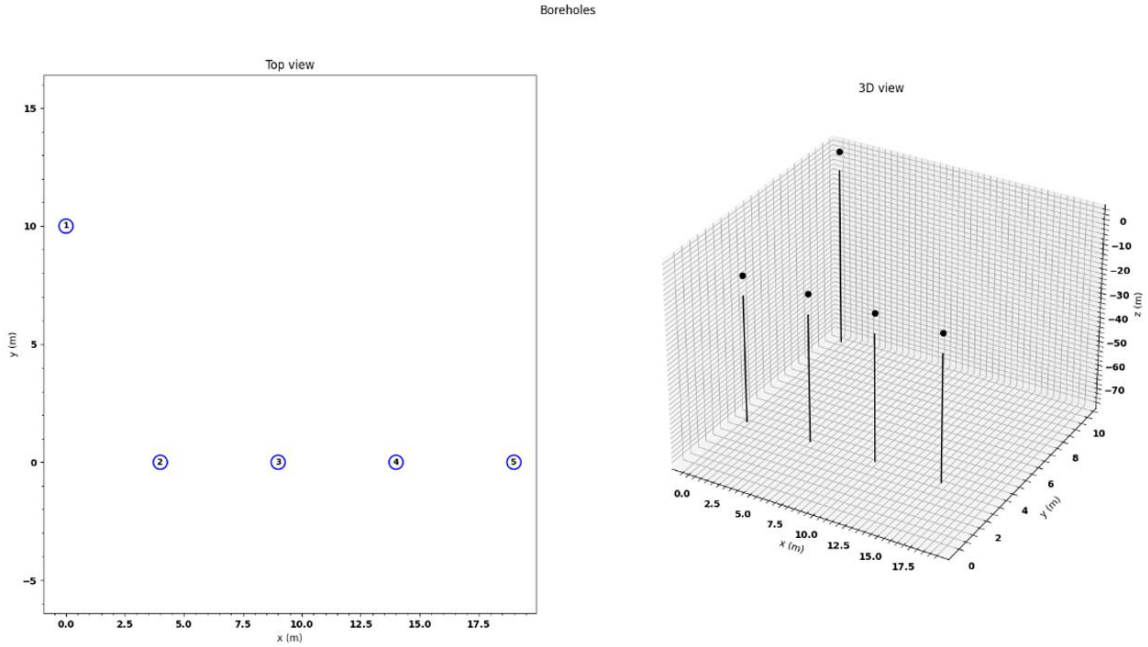


Figure 26: Results figure 1 – borefield layout

The second figure shows the heat pipe layout inside the boreholes (see Figure 27). Both figures visualize geometric data that the user provides and thus should be used to double-check the system geometry.

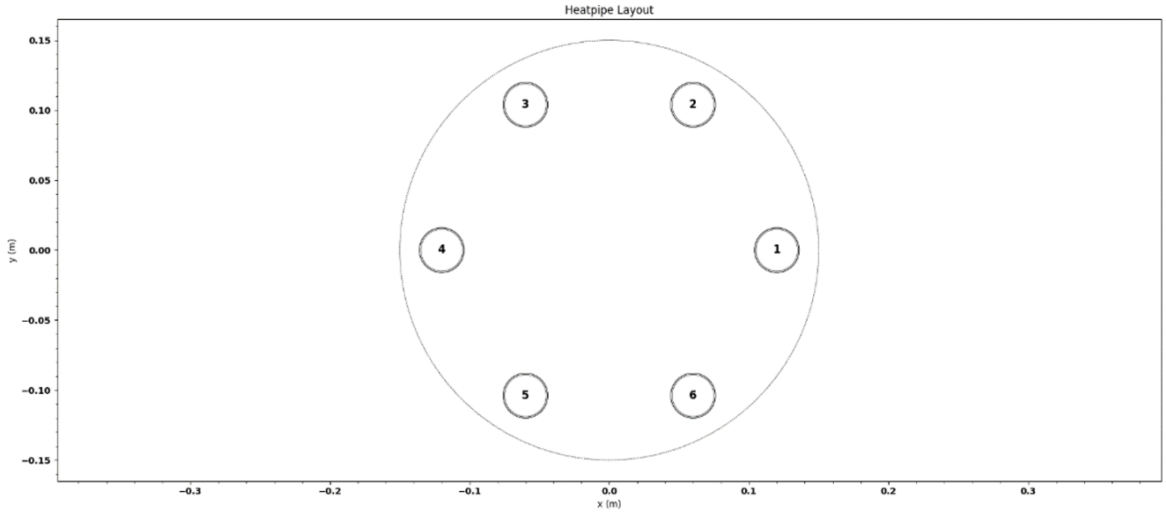


Figure 27: Results figure 2 – heat pipe layout

Figure 28 shows an example of the result plots window. The x-axis is formatted as [mm-dd-hh], month-day-hour. The simulation done in the figure starts on the 15th May and runs a full year.

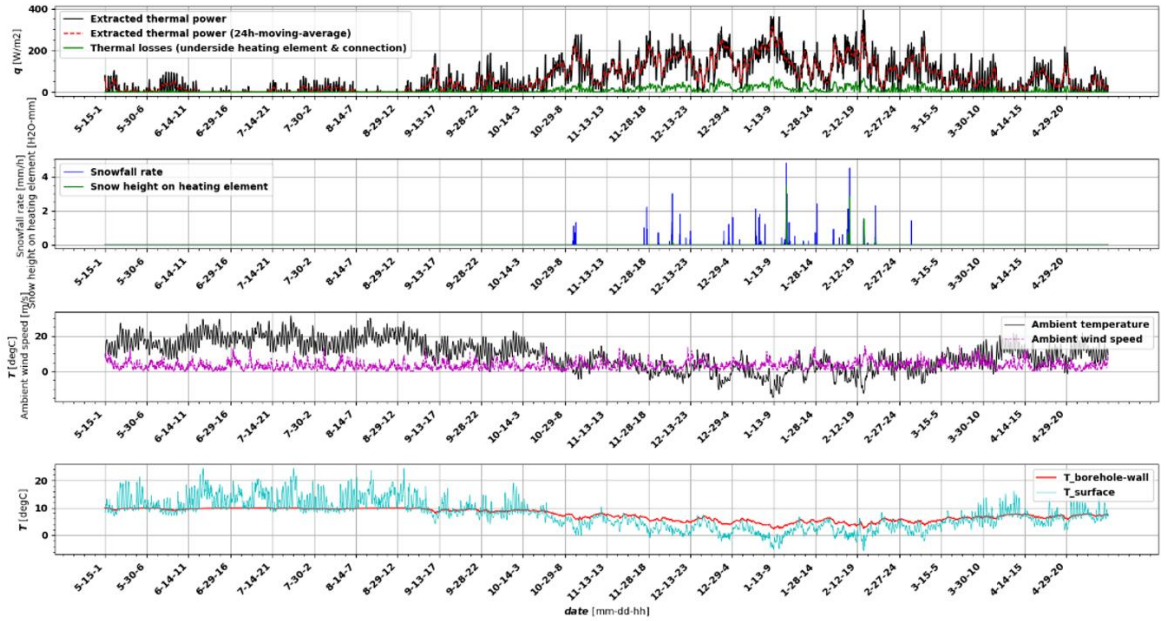


Figure 28: Results figure 3 – extracted power, weather data & temperatures

The first plot in Figure 28 shows the extracted thermal power $\dot{Q}$ and its 24-hour-moving-average as well as the thermal losses incurred at the underside of the heating element and connection lines. The second plot shows the snowfall rate S_r, a predefined array of values priorly imported via the weather file, in combination with the projected snow height on the heating element which is a modelled value. The third plot presents the two important weather parameters of wind speed and temperature above the heating element. The fourth plot constitutes the two temperature profiles at the borehole wall and the heating element

surface that are a direct result of ambient weather conditions and the heat extracted over time (first plot). Details concerning the physical modelling of these parameters can be found in chapter 2.

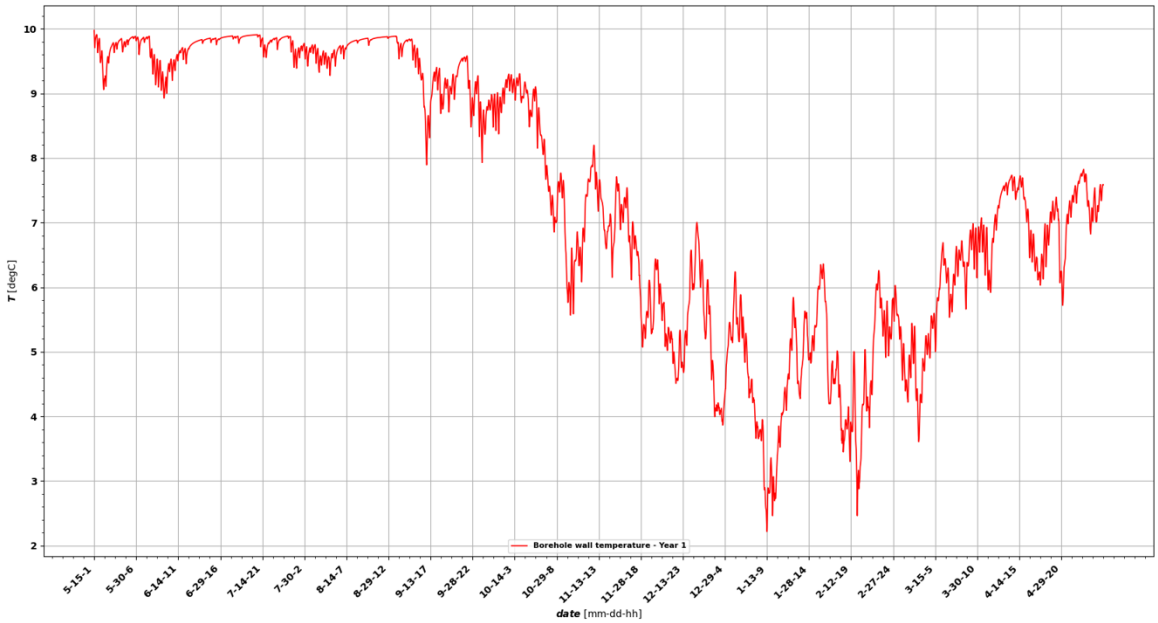


Figure 29: Results figure 4 – borehole wall temperature (single year)

A final result window contains the plots in Figure 29 and Figure 30. In case of a multi-year simulation, the graphs in Figure 30 will appear. Otherwise, Figure 29 is shown. The latter is the isolated plot of the borehole wall temperature over the simulated time period. In the upper plot of Figure 30 the annual borehole wall temperature plots are stacked on top of each other to visualize the cooling of the ground. The bigger temperature delta between the first year and the following years stems from the undisturbed ground temperature that is fed into the simulation as an initial condition. The lower plot in that same figure will appear only in case of a multi-year simulation. It shows the borehole wall temperature at the beginning of the annual heating period which is defined to start on the 1st September. This plot should be used as an indicator of a well dimensioned heating system. If the latter is the case, the plot should level off and asymptotically approach a constant value that represents the long-term thermal performance of the system. In case this value is too low, the usable temperature delta from ground to surface might not be sufficient to keep the surface ice-free during times of high thermal load in winter after multiple simulation years.

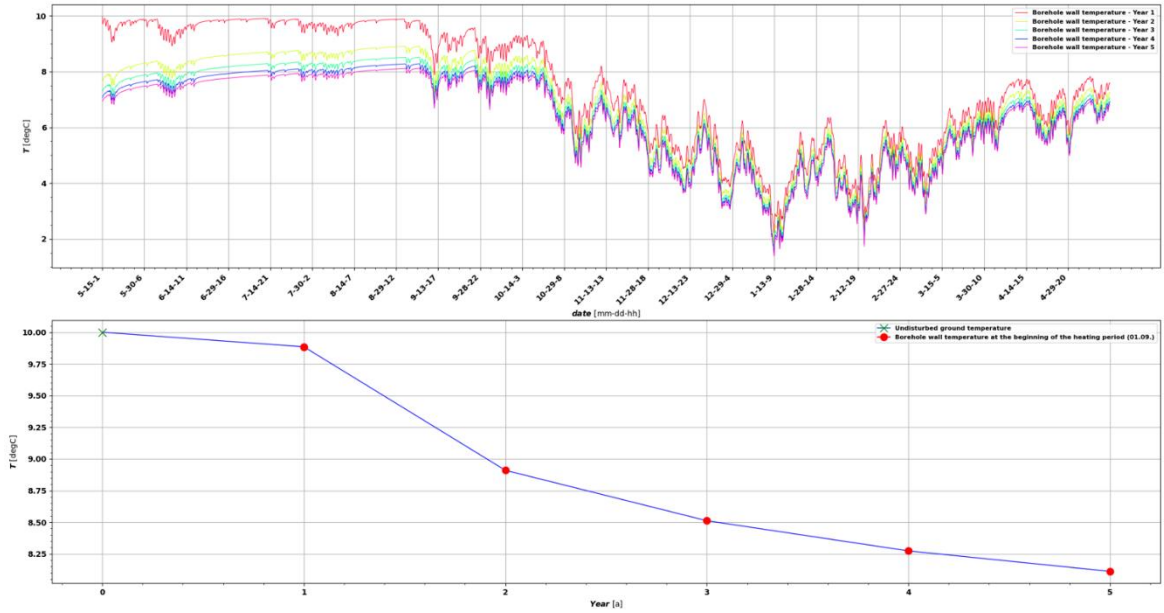


Figure 30: Results figure 4 – borehole wall temperature (multi-year)

All figures can be saved as .png-files.

List of References

ASHRAE (2013): ASHRAE handbook fundamentals 2013. SI edition. Atlanta, GA: ASHRAE.

ASHRAE (2015): ASHRAE Handbook--HVAC Applications (SI Edition). Heating, ventilating, and air-conditioning applications. SI edition. Atlanta: ASHRAE.

Bentz Dale P. (2000): A Computer Model to Predict the Surface Temperature and Time-of-Wetness of Concrete Pavements and Bridge Decks. Hg. v. U. S. Department of Commerce. National Institute of Standards and Technology.

Cimmino, Massimo (2018a): Fast calculation of the g -functions of geothermal borehole fields using similarities in the evaluation of the finite line source solution. In: *Journal of Building Performance Simulation* 11 (6), S. 655–668. DOI: 10.1080/19401493.2017.1423390.

Cimmino, Massimo (2018b): pygfunction: an open-source toolbox for the evaluation of thermal response factors for geothermal borehole fields. In: *Proceedings of eSim 2018, the 10th conference of IBPSA-Canada*.

Cimmino, Massimo; Bernier, Michel (2014): A semi-analytical method to generate g-functions for geothermal bore fields. In: *International Journal of Heat and Mass Transfer* 70 (2), S. 641–650. DOI: 10.1016/j.ijheatmasstransfer.2013.11.037.

Cimmino, Massimo; Bernier, Michel; Adams, François (2013): A contribution towards the determination of g-functions using the finite line source. In: *Applied Thermal Engineering* 51 (1-2), S. 401–412. DOI: 10.1016/j.applthermaleng.2012.07.044.

Claesson, J.; Javed, S. (2011): An Analytical Method to Calculate Borehole Fluid Temperatures for Time-scales from Minutes to Decades. In: *ASHRAE Transactions* 117 (2), S. 279–288.

Claesson, Johan; Javed, Saqib (2012): A Load-Aggregation Method to Calculate Extraction Temperatures of Borehole Heat Exchangers. In: *ASHRAE Transactions* 118 (1), S. 530–539.

Eskilson, P. (1986): Superposition borehole model: manual for computer code. Lund University. Lund, Sweden.

Eskilson, P. (1987): Thermal analysis of heat extraction boreholes. Dissertation. Lund University, Department of Physics. Online verfügbar unter https://portal.research.lu.se/en/publications/thermal-analysis-of-heat-extraction-boreholes, zuletzt geprüft am 17.06.2022.

Faghri, Amir (2018): Handbook of Thermal Science and Engineering. Unter Mitarbeit von F. A. Kulacki. Cham: Springer International Publishing.

Hellström, G. (1991): Ground heat storage. Thermal analyses of duct storage systems., Lund.

Löser, Jan; Klemm, Marco; Hiller, Andreas (2018): Technische Thermodynamik in ausführlichen Beispielen. München: Fachbuchverlag Leipzig im Carl Hanser Verlag.

Polifke, W.; Kopitz, J. (2013): Wärmetransportphänomene, Wärme und Stoffübertragung. Arbeitsunterlagen zur Vorlesung. Skriptum. Technische Universität München, Garching bei München. Lehrstuhl für Thermodynamik, Professur für Thermofluiddynamik.

VDI (2013): VDI-Wärmeatlas. Berlin, Heidelberg: Springer Berlin Heidelberg.

VDI-2055-1, August 2019: Wärme- und Kälteschutz von betriebstechnischen Anlagen. Online verfügbar unter https://www.vdi.de/richtlinien/details/vdi-2055-blatt-1-waerme-und-kaelteschutz-von-betriebstechnischen-anlagen-berechnungsgrundlagen, zuletzt geprüft am 29.06.2020.

Zohuri, Bahman (2016): Heat Pipe Design and Technology. Cham: Springer International Publishing. Online verfügbar unter https://link.springer.com/book/10.1007/978-3-319-29841-2, zuletzt geprüft am 29.06.2020.

Zorn, R.; Steger, H.; Kölbel, T. (2015): De-Icing and Snow Melting System with Innovative Heat Pipe Technology. In: *Proceedings World Geothermal Congress 2015*. Online verfügbar unter https://www.hindawi.com/journals/tswj/2018/4343167/, zuletzt geprüft am 29.06.2020.