IGSHPA Research Track
Las Vegas, December 6-8, 2022

pygfunction 2.2 : New Features and Improvements in Accuracy and Computational Efficiency

Massimo Cimmino **Jonathan C. Cook**

ABSTRACT

Recent improvements to pygfunction, an open-source tool for the calculation of g-functions, are presented. The latest version 2.2 enables the calculation of g-functions for fields containing inclined boreholes. Various changes introduced in versions 2.0, 2.1 and 2.2 have led to significant decreases in calculation time and memory usage while increasing the accuracy of calculations. This allows the calculation of g-functions of fields comprised of thousands of boreholes. A development roadmap towards a version 3.0 is presented. This future release will increase the scope of the tool to consider aditional physical processes.

INTRODUCTION

Thermal response factors, or g-functions, are dimensionless step-response functions that give the effective temperature variation at the borehole walls in a bore field due to constant heat extraction from the bore field. g-Functions are utilized in ground heat exchanger (GHE) simulations by superposition of building heating and cooling loads to predict variations of borehole wall temperatures throughout time. During the design process, GHE simulations are iteratively performed to determine a suitable design for installation. Therefore, quick and accurate evaluation of g-functions enables the development of optimal designs and is critical to the efficient operation of a ground source heat pump (GSHP) system.

Eskilson (1987) introduced the concept of g-functions and developed a numerical finite difference model to evaluate them. g-Functions are defined by the relation:

$$T_b^*(\tau) = T_g - \frac{\bar{Q}'}{2\pi k_s} \cdot g\left(\tau, \frac{r_b}{H}, \frac{B}{H}, \frac{D}{H}\right) \tag{1}$$

where T_b^* is the effective borehole wall temperature, τ is the dimensionless time, T_g is the undisturbed ground temperature, $\bar{Q}'$ is the constant average heat extraction rate per unit borehole length, k_s is the soil thermal conductivity, g is the g-function, r_b is the borehole radius, B is the borehole spacing, D is the borehole buried depth, and H is the borehole length. Figure 1 presents the g-functions of a field of 8 inclined boreholes, where t/t_s is the dimensionless time with $t_s = H^2/9\alpha_s$ the borefield characteristic time and α_s the ground thermal diffusivity.

Massimo Cimmino (massimo.cimmino@polymtl.ca) is a professor of mechanical engineering at Polytechnique Montréal and Jonathan C. Cook (jack-c-cook@proton.me) is a software engineer and a private researcher.

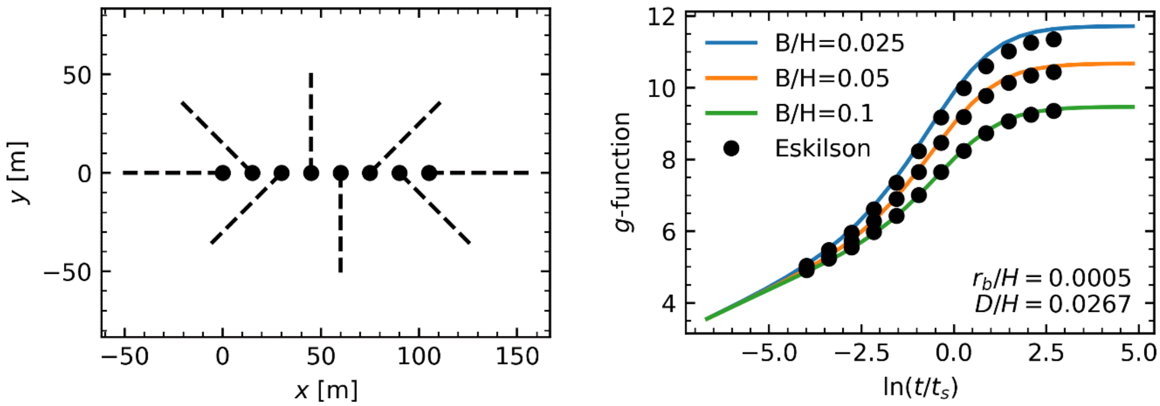

Figure 1 Top view of the bore field (left), and comparison between pygfunction and Eskilson's g-functions (right)

Following the reintroduction of the analytical finite line source solution (FLS) by Zeng et al. (2002), spatial superposition of the FLS was proposed to evaluate g-functions (Lamarche and Beauchamp, 2007; Claesson and Javed, 2011). g-Functions evaluated in this manner were shown to overestimate Eskilson's g-functions due to differences in the boundary condition at the borehole walls (Fossa, 2011). Cimmino and Bernier (2014) proposed to axially discretize each borehole into segments and superimpose the FLS solution in both space and time to reproduce Eskilson's uniform borehole wall temperature boundary condition. Their method was later extended to inclined boreholes (Lazzarotto, 2016), and to more detailed boundary conditions accounting for the heat transfer between the fluid and the borehole wall (Cimmino, 2015, 2019). Although not presented here, there is ongoing research on the development of analytical solutions and g-functions for fields of boreholes with groundwater advection and depth-dependent ground properties.

A series of simplifications and algorithmic improvements have been proposed to speed-up the evaluation of g-functions (Cimmino, 2018a; Dusseault et al., 2018; Nguyen and Pasquier, 2021; Prieto and Cimmino, 2021). While these newer methods successfully lower computational requirements for the evaluation of g-functions, they are often more complex than their predecessors, thereby hindering their adoption and extension by other researchers and practitioners. The reproducibility of results poses additional issues when considering computational efficiency, as it depends not only on the method but also on the specifications of the machine it runs on, the programming language and environment (including the version), and the often-overlooked quality of the implementation.

The open-source pygfunction package (Cimmino, 2018b) enables the evaluation of g-functions of fields of arbitrarily positioned boreholes with different boundary conditions and solvers. The package has seen active development since its introduction. It now supports the calculation of g-functions of fields of thousands of boreholes with orders of magnitude reduction in calculation time and memory usage, along with several new features. These improvements are in part due to the implementation of new methods and optimization of the codebase but also from updates to other open-source projects pygfunction depends on; namely NumPy (Harris et al., 2020) for array programming, SciPy (Virtanen et al., 2020) for scientific computing, Matplotlib (Hunter, 2007) for visualization, and CoolProp (Bell et al., 2014) for physical properties of fluids. The open-source nature of the package makes it possible for anyone to evaluate g-functions using state-of-the-art methods or for researchers to modify or extend the package for their own use. pygfunction has already been used within other open-source projects aimed at the design of borefields: GHEtool (Peere et al., 2021) and the Ground Heat Exchanger Design Toolbox (GHEDT) (Cook, 2021).

This paper details the new features and improvements to pygfunction from its initial version 1.0 to the current version 2.2. An overview of changes affecting its computational efficiency and accuracy is first presented. Then, the implementation of support for inclined boreholes is described. Results present the evolution of computational efficiency and accuracy across versions. Finally, a development roadmap leading to a version 3.0 release is presented.

NEW FEATURES AND CHANGES TO EXISTING MODULES

pygfunction is structured into modules: *boreholes*, *gfunction*, *heat_transfer*, *load_aggregation*, *media*, *networks*, *pipes*, and *utilities*. The modules were presented in an earlier publication (Cimmino, 2018b), except for the *networks* and *media* modules introduced in versions 1.1.1 and 2.0, respectively. The *networks* module enables the configuration of piping connections between boreholes and the evaluation of g-functions for mixed parallel and series connections between boreholes (Cimmino, 2019), including fields of coaxial boreholes since version 2.1. The *media* module gives access to fluid properties using CoolProp (Bell et al., 2014). The *gfunction* module was refactored to replace the dedicated functions for different boundary conditions by a single class that can handle all calculation options. pygfunction supports 3 boundary conditions: uniform and equal heat extraction rates (Claesson and Javed, 2011), uniform and equal borehole wall temperatures (Cimmino and Bernier, 2014), and mixed inlet fluid temperatures (Cimmino, 2019). When computing g-functions, a user might use a combination of the *boreholes*, *pipes*, *networks* and *media* modules to define the borefield, depending on the chosen boundary condition, and use the *gfunction* module to evaluate the g-functions.

Since version 2.0, pygfunction is exclusively developed in Python 3, and pygfunction has seen continuous improvements in computational efficiency between each release. Part of these improvements is due to the implementation of a new solver (i.e. the *equivalent* solver) based on the equivalent borehole method of Prieto and Cimmino (2021) in version 2.1, and to the implementation of an approximation of the FLS (Cimmino, 2021) in version 2.2. The *equivalent* solver identifies groups of boreholes (usually 3 to 5) within the borefield that are expected to share similar borehole wall temperatures and heat extraction rates. Each group is represented by a single "equivalent" borehole (i.e. all boreholes of the same group share the same borehole wall temperature and heat extraction rate profiles). This massively reduces the size of the system of equations and the number of required FLS evaluations.

Significant performance improvements have been achieved by extensive use of NumPy array computations. NumPy and SciPy operations performed on NumPy arrays are executed in optimized low-level (C or Fortran) compiled code. All modules have been refactored to make better use of NumPy array computations. pygfunction v1.0 computed segment-to-segment responses in pure Python syntax with multi-threading. Cook and Spitler (2021) computed FLS responses 8.6 times faster in C++ with multi-threading. Since v2.0, pygfunction now has vectorized the FLS evaluation via the *quad_vec* function recently introduced in SciPy v1.4. This allows pygfunction to evaluate the FLS for multiple pairs of boreholes at the same time without the need for the *multiprocessing* package. The computational efficiency of *quad_vec* dwarfs multi-threaded C++ code as implemented by Cook (2021, p. 57).

An important driver for improvements has been feedback from users and researchers. Cook and Spitler (2021) highlighted issues with memory usage in pygfunction. Their results also showed that the *similarities* solver (Cimmino, 2018a) did not scale well with the number of boreholes. This motivated the refactoring of the *similarities* solver and the *gfunction* module. Several new features have been introduced following user involvement on the GitHub repository of pygfunction. This is the case for coaxial boreholes, the *media* module, and the development of an unequal discretization scheme along boreholes (described in the next section), all proposed by the second author.

Non-uniform segment discretization along boreholes

The calculation of g-functions using the FLS solutions was previously done using a uniform discretization of boreholes (i.e. segments of equal lengths). Cimmino and Bernier (2014) suggested the use of n_s = 12 equal segments along the length of the boreholes after observing a 4.7 % maximum error on the steady-state value of the g-functions of fields up to 100 boreholes. Equal segments were chosen to simplify the evaluation of the FLS, since similarities can reduce the number of required evaluations in this case (Cimmino, 2018a). The purpose of the discretization is to capture the variation of heat extraction along the borehole length. The heat extraction rate varies the most at the top and bottom of the borehole (Cimmino and Bernier, 2014). Thus, a uniform segment length is sub-optimal. Eskilson (1987) proposed an expanding discretization towards the middle of the boreholes, with a factor $\gamma = \sqrt{2}$ between the length of 2 consecutive segments. The maximum number of segments considered was 12.

A non-uniform segment discretization scheme was implemented into pygfunction. The implemented scheme specifies the end-length-ratio (i.e. the ratio x_1 of the total length H represented by each of the first and last segments along the borehole), rather than the expansion factor. The expansion factor can be evaluated from the roots of the polynomial:

$$\begin{cases} 0 = (1 - 2x_1) - x_1 \cdot \gamma + 2x_1 \cdot \gamma^{n_s/2} & \text{if } n_s \text{ is even} \\ 0 = (1 - 2x_1) - x_1 \cdot \gamma + x_1 \cdot \gamma^{(n_s-1)/2} + 2x_1 \cdot \gamma^{(n_s-1)/2+1} & \text{if } n_s \text{ is odd} \end{cases} \quad (2)$$

where x_1 is the end-length-ratio, γ is the expansion factor, and n_s is the number of segments.

Values of x_1 = 0.02 and n_s = 8 provide accurate results, as will be shown in the results section. For this combination of parameters, the expansion factor is γ = 2.485. This is in line with the findings of Lamarche (2017) who used a piecewise-linear discretization of heat extraction rates along boreholes, which allowed them to represent sharp variations of heat extraction rates at the ends of the boreholes and lowered the number of segments to 8. Figure 2 compares a uniform discretization of 12 segments and a non-uniform discretization of 8 segments. It is shown that the non-uniform discretization has a finer discretization at the ends of the borehole.

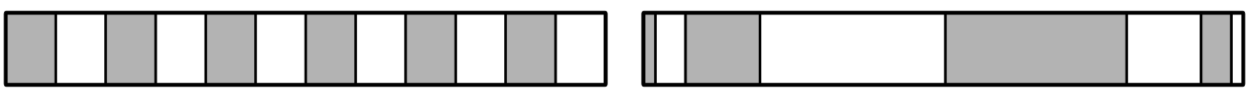

Figure 2 Uniform discretization of 12 segments (left), and non-uniform discretization of 8 segments (right)

Inclined boreholes

Calculation of g-functions for fields with inclined boreholes using discretized boreholes with uniform borehole wall temperature has so far been limited to the work of Lazzarotto (2016) and Lazzarotto and Björk (2016). Inclined boreholes were implemented into pygfunction version 2.2. The FLS is solved following the method of Lazzarotto (2016), with the remaining two integrals solved using a Gauss-Legendre quadrature over the segment lengths and an adaptative quadrature over the time variable (using the *quad_vec* function). Inclined boreholes are fully integrated with the different solvers and features of pygfunction, including the *similarities* solver to accelerate the evaluation of g-functions and an approximation of the FLS based on the method of Cimmino (2021).

Figure 1 presents the g-functions of a field of 8 boreholes with an inclination of 20°, calculated with a non-uniform discretization of 8 segments using the *similarities* solver. Eskilson's g-functions are a reproduction of the curves presented in the thesis (Eskilson, 1987). The maximum error is 2.5 % and is observed for B/H = 0.025 and $\ln(t/t_s)$ = 2.70. Possible causes for this error are a mismatch of the buried depth D used in the calculation (this information was not provided by Eskilson) and unconverged values of the g-functions due to the coarse grids used by Eskilson. Each g-function was evaluated in an average of 2.36 seconds for 30 time steps.

RESULTS

The evolution of the accuracy and computational efficiency of pygfunction since version 1.0 is presented by evaluating the g-functions of fields of up to 400 boreholes with a uniform borehole wall temperature boundary condition. All L-shaped, U-shaped, box-shaped and rectangular array configurations of 1 × 2 to 20 × 20 boreholes are tested, as well as random configurations of up to 400 boreholes in increments of 10 boreholes generated from Poisson disk sampling using Bridson's algorithm (Bridson, 2007). This totals 1635 borefields. Boreholes have a length H = 150 m, a burial depth D = 4 m and a radius r_b = 0.075 m. The nominal spacing in the borefields is B = 7.5 m (B/H = 0.05). The ground has a thermal diffusivity $\alpha_s = 10^{-6}$ m²/s. g-Functions are evaluated from t_1 = 100 h to t_{30} = 3 000 yr at 30 geometrically expanding time steps, with $t_n = t_1 \cdot (1-\gamma^n)/(1-\gamma)$ and γ = 1.479.

All g-functions are computed on a computer with an Intel Core i9-11900K 3.5 GHz 8-core processor. The computer runs on Kubuntu 20.04.4 LTS (operating system), Python 3.7.13, NumPy 1.21.6, SciPy 1.7.3, Coolprop 6.4.1 and MatPlotLib 3.5.1. Reference results to quantify the accuracy are generated with pygfunction version 2.2 using the *detailed* solver and 32 unequal segments along the boreholes. The reference vertical grid is generated by splitting segments of the 8 unequal segment discretization into 4 segments each. The *detailed* solver is used as a reference since it does not consider any simplifications for the evaluation of the FLS. Other combinations of versions and solver options are chosen to highlight the progression of pygfunction: (1) version 1.0 using the *similarities* solver and 12 uniform segments, (2) version 2.0 using the *similarities* solver and 12 uniform segments, (3) version 2.1 using the *similarities* solver and 8 non-uniform segments, (4) version 2.1 using the *equivalent* solver and 8 non-uniform segments, and (5) version 2.2 using the *equivalent* solver, 8 non-uniform segments and the approximation of the FLS.

Accuracy

Figure 3 compares the maximum relative error, $(g - g_{ref})/g_{ref}$, for the different versions of pygfunction. Versions 1.0 and 2.0 show similar errors. The decreased error from version 2.0 to 2.1 is due to the non-uniform segment discretization. The *equivalent* solver is not as accurate as the *similarities* solver since it introduces additional simplifications in the thermal model, but version 2.1 *equivalent* with 8 unequal segments is still more accurate than version 2.0 *similarities* with a uniform discretization of 12 segments. Version 2.2 shows a negligible increase in the error due to the use of the approximation of the FLS. Overall, it is shown that pygfunction has gained accuracy compared to version 1.0.

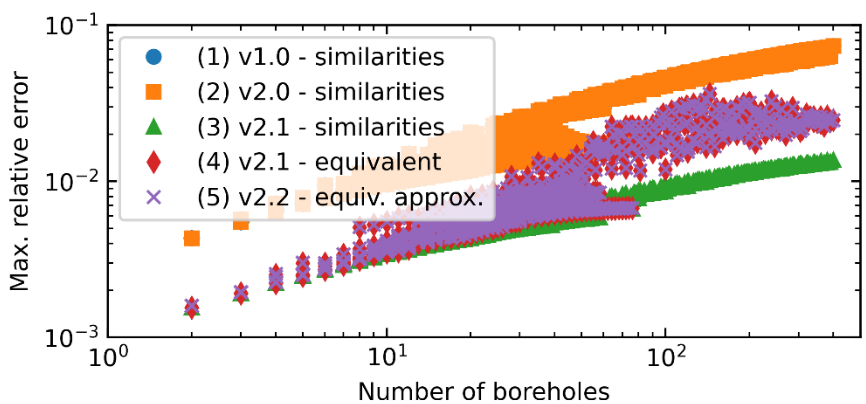

Figure 3 Maximum relative error on the g-functions

Computational efficiency

Figure 4 compares the calculation time and memory usage for the different versions of pygfunction. The decreased calculation time from version 1.0 to 2.0 is mainly due to a refactoring of the *similarities* solver to make better use of NumPy capabilities and to the use of the newly introduced *quad_vec* function of SciPy. The decreased calculation time from version 2.0 to 2.1 is due to the decrease in the number of segments enabled by an unequal segment discretization. The *equivalent* solver is significantly faster than the *similarities* solver due to the reduced number of evaluations of the FLS. The efficiency gain increases with the number of boreholes: the calculation times for a field of 400 randomly positioned boreholes are 26.1 and 0.775 seconds for the *similarities* and *equivalent* solvers, respectively. The decreased calculation time from version 2.1 to 2.2 is due to the approximation of the FLS and further optimization of the code, lowering the calculation time to 0.245 seconds for the same field. The efficiency gains introduced by the *equivalent* solver removes the requirement of a g-function database for use in pygfunction integrated design tools (Cook, 2021). In terms of memory usage, the decrease from version 1.0 to 2.0 are due to code refactoring following the observations of Cook and Spitler (2021). The memory usage for the *similarities* method is mainly a function of the total number of segments, which explains the decrease from version 2.0 to 2.1. The *equivalent* solver in versions 2.1 and 2.2 significantly decreases the memory usage when compared to the *similarities* solver, due to the reduced number of considered equivalent boreholes in the calculations. The reduction is such that memory usage ceases to be an issue: the required memory for a field of 400 randomly positioned boreholes is 5.26 Gb and 43.7 Mb for the *similarities* and *equivalent* solvers, respectively. The memory usage of the *equivalent* solver shows plateaus at approximately 0.8 Mb and 10 Mb for versions 2.1 and 2.2, respectively. The cause of these plateaus is unknown.

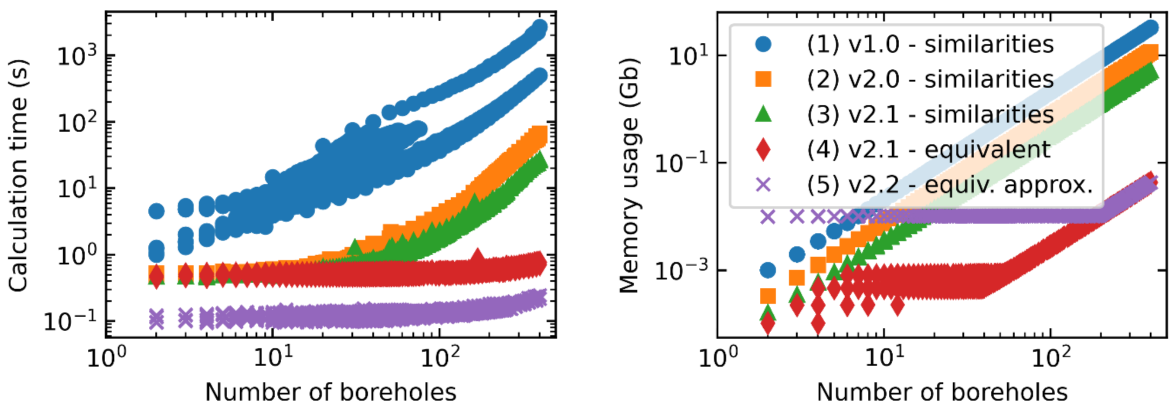

Figure 4 Calculation time (left), and peak memory usage (right)

The reductions in calculation time and memory usage make the calculation of g-functions of very large fields possible. The largest borefield known to the authors is located at Daxing International Airport (China) and is reported to be comprised of 10 497 boreholes. g-Functions of such large borefields have never been reported due to the previously enormous requirements in calculation time and memory usage (Cook and Spitler, 2021). Figure 5 presents the g-functions of a field of 110 × 100 borehole in a rectangular configuration. The calculation time was 78.0 seconds on average, with 68.0 seconds devoted to the identification of equivalent boreholes and 8.18 seconds devoted to the calculation of the g-function over 30 time steps. Note that the identification of equivalent boreholes is independent of the number of time steps.

DEVELOPMENT ROADMAP

While pygfunction version 1.0 contributed the first open-source tool for the calculation of g-functions, versions 2.0, 2.1 and 2.2 focused on increasing its computational efficiency. pygfunction is now able to evaluate g-functions for fields containing vertical or inclined boreholes with relative ease. Going forward, efforts will be put into increasing its capabilities and its ease of use. Two major changes are planned for an eventual version 3.0: (1) a restructuration of the documentation, including the introduction of a "Getting started" tutorial, and (2) a simulation module to enable the simulation of complete ground-source heat pump systems, including the borefield and the heat pump. For subsequent releases, the scope of g-function calculations will be increased to consider additional physical processes, e.g. groundwater advection, layered ground conditions, and short-term effects. This poses significant challenges: while there are known analytical solutions that can be used for the calculation of g-functions, they require extensive changes to the *similarities* and *equivalent* solvers to enable efficient calculations.

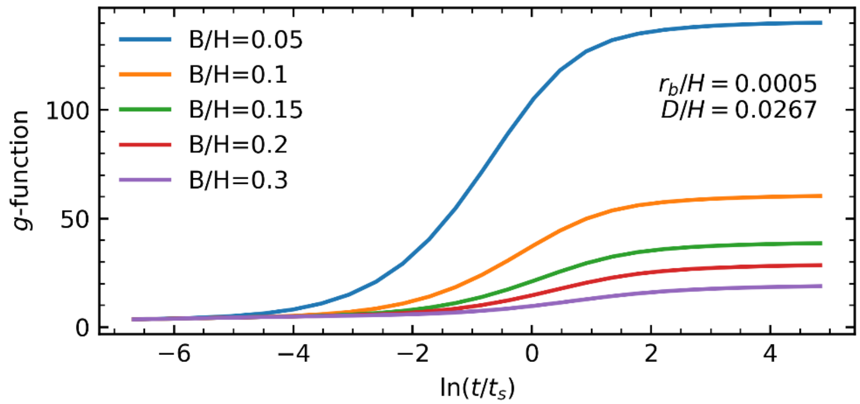

Figure 5 g-Functions of a field of 11 000 boreholes in a 110 × 100 configuration

CONCLUSION

The recent improvements to pygfunction were presented. New features since its original release include the implementation of inclined boreholes, the *media* and *network* modules, and the *equivalent* solver. Factoring in optimizations to the codebase and updates to other open-source projects, pygfunction version 2.2 achieves higher accuracy than version 1.0 at a fraction of the cost in calculation time and memory usage. Future development will increase the scope of pygfunction to include additional physical processes such as groundwater advection, layered ground conditions, and short-term effects.

ACKNOWLEDGMENTS

Contributions to open-source projects like pygfunction are not limited to code development. Various indirect contributions in the form of questions, bug reports and comments on efficiency made on the GitHub platform have led to improvements in the computational efficiency of pygfunction, in the clarity of its documentation and its ease of use. We acknowledge all who initiated or participated on "issues". The second author would like to acknowledge the DOE and Dr. Jeffrey D. Spitler for indirectly aiding contributions to pygfunction.

NOMENCLATURE

α_s = Ground thermal diffusivity (m^2/s)
γ = Segment discretization expansion factor (-)
B = Borehole spacing (m)
D = Borehole buried depth (m)
g = g-function (-)
H = Borehole length (m)
k_s = Ground thermal conductivity (W/m-K)
D = Borehole buried depth (m)
n_s = Number of segments (-)
$\bar{Q}'$ = Average heat extraction rate per unit borehole length (W/m)
r_b = Borehole radius (m)

T_b^*	=	Effective borehole wall temperature (°C)	t_s	=	Borefield characterisitic time (s)
T_g	=	Undisturbed ground temperature (°C)	x_1	=	End-length-ratio (-)
t	=	Time (s)			

REFERENCES

Bell, I. H., J. Wronski, S. Quoilin and V. Lemort. 2014. *Pure and pseudo-pure fluid thermophysical property evaluation and the open-source thermophysical property library CoolProp*. Industrial & Engineering Chemistry Research 53(6): 2498-2508.

Bridson, R. 2007. *Fast Poisson disk sampling in arbitrary dimensions.* SIGGRAPH sketches 10(1): 1.

Cimmino, M. and M. Bernier. 2014. *A semi-analytical method to generate g-functions for geothermal bore fields*. International Journal of Heat and Mass Transfer 70: 641-650.

Cimmino, M., 2015. *The effects of borehole thermal resistances and fluid flow rate on the g-functions of geothermal bore fields*. International Journal of Heat and Mass Transfer 91: 1119-1127.

Cimmino, M. 2018a. *Fast calculation of the g-functions of geothermal borehole fields using similarities in the evaluation of the finite line source solution*. Journal of Building Performance Simulation 11(6): 655-668.

Cimmino, M. 2018b. *pygfunction: an open-source toolbox for the evaluation of thermal response factors for geothermal borehole fields*. eSim 2018 – the 10th conference of IBPSA-Canada, Montréal, Canada: 492-501.

Cimmino, M., 2019. *Semi-Analytical Method for g-Function Calculation of bore fields with series-and parallel-connected boreholes*. Science and Technology for the Built Environment 25(8): 1007-1022.

Cimmino, M. 2021. *An approximation of the finite line source solution to model thermal interactions between geothermal boreholes*. International Communications in Heat and Mass Transfer 127: 105496.

Claesson, J. and S. Javed. 2011. *An analytical method to calculate borehole fluid temperatures for time-scales from minutes to decades*. In ASHRAE Transactions 117(2): 279-288.

Cook, J. C. 2021. *Development of computer programs for fast computation of g-functions and automated ground heat exchanger design*. M.Sc. Thesis. Oklahoma State University, Stillwater, OK, USA.

Cook, J. C. and J. D. Spitler. 2021. *Faster computation of g-functions used for modeling of ground heat exchangers with reduced memory consumption*. International Building Simulation Conference 2021. Bruges, Belgium.

Dusseault, B., P. Pasquier and D. Marcotte. 2018. *A block matrix formulation for efficient g-function construction*. Renewable Energy 121: 249-260.

Eskilson, P. 1987. *Thermal analysis of heat extraction boreholes*. Ph.D. Thesis. Lund university, Lund, Sweden.

Fossa, M. 2011. *The temperature penalty approach to the design of borehole heat exchangers for heat pump applications*. Energy and Buildings 43(6): 1473-1479.

Harris, C. R., K. J. Millman, S. J. van der Walt et al. 2020. *Array programming with NumPy*. Nature 585(7825): 357-362.

Hunter, J. D. 2007. *Matplotlib: A 2D Graphics Environment*. Computing in Science & Engineering 9(3): 90-95.

Lamarche, L. and B. Beauchamp. 2007. *A new contribution to the finite line-source model for geothermal boreholes*. Energy and Buildings 39(2): 188-198.

Lamarche, L. 2017. *G-function generation using a piecewise-linear profile applied to ground heat exchangers*. International Journal of Heat and Mass Transfer 115: 354-360.

Lazzarotto, A. 2016. *A methodology for the calculation of response functions for geothermal fields with arbitrarily oriented boreholes – Part 1*. Renewable Energy 86: 1380-1393.

Lazzarotto, A. and F. Björk. 2016. *A methodology for the calculation of response functions for geothermal fields with arbitrarily oriented boreholes – Part 2*. Renewable energy 86: 1353-1361.

Nguyen, A. and P. Pasquier. 2021. *A successive flux estimation method for rapid g-function construction of small to large-scale ground heat exchanger*. Renewable Energy 165: 359-368.

Peere, W., D. Picard, I. Cupeiro Figueroa, W. Boydens and L. Helsen. 2021. *Validated combined first and last year borefield sizing methodology*. International Building Simulation Conference 2021. Bruges, Belgium.

Prieto, C. and M. Cimmino. 2021. *Thermal interactions in large irregular fields of geothermal boreholes: the method of equivalent boreholes*. Journal of Building Performance Simulation 14(4): 446-460.

Virtanen, P., R. Gommers, T. E. Oliphant et al. 2020. *SciPy 1.0: Fundamental algorithms for scientific computing in Python*. Nature Methods 17(3): 261-272.

Zeng, H. Y., N. R. Diao and Z. H. Fang. 2002. *A finite line-source model for boreholes in geothermal heat exchangers*. Heat Transfer – Asian Research 31(7): 558-567.