

# Near-instant g-function Construction with Artificial Neural Networks

**Bernard Dusseault**

**Philippe Pasquier**

## ABSTRACT

*A g-function is a useful tool that simplifies the calculations of heat exchanges in a ground-coupled heat pump system. In this work, we show how an artificial neural network can be trained to construct a g-function with high efficiency and reliability. First, we show how a block matrix formulation can be used to construct rapidly a g-function. This method is then used to assemble a database of 27,000 g-functions with a variety of input parameters. This database of g-functions is used to train a feed-forward neural network having three hidden layers using the back-propagation algorithm to update the weights and biases of the neurons. The network we developed in this work can estimate the long-term g-function of a ground heat exchanger made of 1 to 10 boreholes over a duration of 100 years with various ground thermal properties, borehole field configurations, length and buried depth in a few milliseconds. The contribution of this work is to lay out the methodology to allow anyone to construct a g-function with an artificial neural network.*

## INTRODUCTION

Geothermal heat exchangers (GHE) are useful constructs that allow for heating and cooling of buildings at a fraction of the operating costs of conventional systems. The most common type is made of vertical boreholes through which a heat-carrying fluid flows. Calculating the thermal response of a GHE to a heating or cooling load can be accomplished using the elegant concept of g-function (Eskilson 1987). A g-function is a smooth and monotone function that links a heat extraction rate to the average borehole wall temperature in a GHE. Such function varies according to many thermal properties and geometrical relations and is different for each GHE. In the last years, many authors have come up with new strategies to construct g-function under different assumptions (Cimmino and Bernier 2014; Lazzarotto and Björk 2016; Marcotte and Pasquier 2014; Cimmino, et al. 2013; Dusseault, et al. 2017).

The modelling of GHE is also useful for distributed thermal response tests, model-based optimisation and sizing, and can even be used in model-based control of ground-source heat pump systems (Atam and Helsen 2016a; Atam and Helsen 2016b; Atam, et al. 2016).

The ever-increasing computational capability of computers have led to the democratization of artificial neural networks (ANNs). Such networks, whose primary objective is to emulate the learning capabilities of a biological brain, can be trained to perform different tasks. In the field of geothermal energy, ANNs have been used to predict heat pump's COP (Esen, et al. 2008; Sun, et al. 2015), efficiency of a district geothermal system (Arat and Arslan 2017), model a direct expansion system (Fannou, et al. 2014), control district (Yabanova and Keeba 2013) and hybrid (Gang, et al. 2014) systems or construct the short-term g-function of a borehole (Pasquier, et al. 2018).

ANNs can also be trained to approximate smooth and monotone multivariable functions. Indeed, the universal approximation theorem (Cybenko 1989; Hornik, et al. 1989) states that a network with a single hidden layer with a finite number of neurons can approximate any continuous function on compact subsets, under mild assumptions on its activation function. In this paper, the learning abilities of ANNs are used to train a network that

can be used to construct a g-function. To assemble a training set, a block matrix formulation is used to build a database made of 27,000 g-functions. We illustrate in the paper how to train an ANN to reproduce accurately in a split second the g-function of a GHE placed on a regular grid made of up to 10 boreholes.

## CONSTRUCTION OF A G-FUNCTION WITH A BLOCK MATRIX FORMULATION

To create a database of g-functions used to train the ANN, the block matrix formulation presented by Dusseault, et al. (2017) was selected mainly for its speed and efficiency. Indeed, this approach can construct a g-function for a GHE made of 50 boreholes over 40 years in only 0.46 second. This efficiency was a necessity to obtain under a reasonable time a large database of tens of thousands of g-functions. The core concept behind the block matrix formulation is to construct a linear system of equations accounting simultaneously for the thermal interactions between all  $m$  boreholes in a GHE at all  $n$  time steps.

All thermal interactions ( $\Delta T$ ) are calculated using the finite line-source model of Claesson and Javed (2011) described by Eq 1 to 3 and computed with a unit heat load of  $\tilde{q} = 1$  W/m.

$$\Delta T(r_{j \rightarrow i}, t_k) = \frac{\tilde{q}}{4\pi k_s} \int_0^{\infty} e^{-r_{j \rightarrow i}^2/s^2} \frac{Y(Hs, Ds)}{Hs^2} ds$$

$$\frac{1}{\sqrt{(4k_s t_k)/C_s}} \quad (1)$$

$$Y(h, d) = 2 \operatorname{ierf}(h) + 2 \operatorname{ierf}(h + 2d) - \operatorname{ierf}(2h + 2d) - \operatorname{ierf}(2d) \quad (2)$$

$$\operatorname{ierf}(X) = X \operatorname{erf}(X) - \frac{1}{\sqrt{\pi}} \left(1 - e^{-X^2}\right) \quad (3)$$

First, consider a GHE composed of a single borehole. To calculate its g-function, one would first have to calculate the mean borehole wall temperature  $T_b$  at each time step of the simulation. This can be done using Eq (4) with  $m = 1$  (one borehole):

$$T_{b_i} = T_g + \sum_{j=1}^m \Delta T_{j \rightarrow i} \quad (4)$$

where  $j$  and  $i$  stand for emitting and receiving borehole, which in this case is the same borehole, and where  $T_g$  is the initial ground temperature. All values of  $\Delta T$  can be found simultaneously using a convolution product in the spectral domain (Marcotte and Pasquier 2008). Convolution in the time domain can however be accomplished by the right-end part of Eq (5):

$$\Delta T(r_{1 \rightarrow 1}, q, t) = (\mathbf{f} * \mathbf{g})(t) = \sum_{k=1}^n f(t_k) g(t - t_{k-1}) \quad (5)$$

where  $f$  represents the incremental heat load emitted by a borehole throughout the simulation. In Eq (5), the function  $g(t - t_{k-1})$  is used to indicate the solution of the FLS model (Eq 1-3) under a heat load  $\tilde{q}$  over the duration specified in the parentheses.

For convenience, Eq (5) can also be presented in a matrix form as shown by Eq (6).

$$\begin{bmatrix} \Delta T(t_1, r_{1 \rightarrow 1}) \\ \Delta T(t_2, r_{1 \rightarrow 1}) \\ \Delta T(t_3, r_{1 \rightarrow 1}) \\ \vdots \\ \Delta T(t_n, r_{1 \rightarrow 1}) \end{bmatrix} = \begin{bmatrix} g(t_1 - t_0) & 0 & 0 & \dots & 0 \\ g(t_2 - t_0) & g(t_2 - t_1) & 0 & \dots & 0 \\ g(t_3 - t_0) & g(t_3 - t_1) & g(t_3 - t_2) & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ g(t_n - t_0) & g(t_n - t_1) & \dots & g(t_n - t_{n-2}) & g(t_n - t_{n-1}) \end{bmatrix} \begin{bmatrix} f(t_1) \\ f(t_2) \\ f(t_3) \\ \vdots \\ f(t_n) \end{bmatrix} \quad (6)$$

Using a more compact format and generalizing for any emitting ( $j$ ) and receiving ( $i$ ) boreholes, the matrix formulation of Eq (6) becomes:

$$\Delta T_{i \rightarrow j} = G_{j \rightarrow i} f_j \quad (7)$$

To account for the thermal interactions between the  $m$  boreholes of a field, Eq (7) can be replicated  $m \times m$  times and presented in a compact matrix form to integrate all spatial superpositions (Eq (8)). The g-function is obtained by solving a linear system of equations under the constraint of a unit heat load expressed in matrix form by:

$$\mathbf{I} \mathbf{f} = \mathbf{\Lambda} = [1 \ 0 \ 0 \ \dots \ 0]' \quad (8)$$

Adding the previous constraints to guarantee the uniqueness of the solution, the resulting linear system is given by:

$$\begin{bmatrix} G_{1 \rightarrow 1} & G_{2 \rightarrow 1} & G_{3 \rightarrow 1} & \dots & G_{m \rightarrow 1} & \mathbf{I} \\ G_{1 \rightarrow 2} & G_{2 \rightarrow 2} & G_{3 \rightarrow 2} & \dots & G_{m \rightarrow 2} & \mathbf{I} \\ G_{1 \rightarrow 3} & G_{2 \rightarrow 3} & G_{3 \rightarrow 3} & \dots & G_{m \rightarrow 3} & \mathbf{I} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ G_{1 \rightarrow m} & G_{2 \rightarrow m} & G_{3 \rightarrow m} & \dots & G_{m \rightarrow m} & \mathbf{I} \\ \mathbf{I} & \mathbf{I} & \mathbf{I} & \dots & \mathbf{I} & \mathbf{0} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ \vdots \\ f_m \\ -(T_b - T_g) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ \Lambda \end{bmatrix} \quad (9)$$

where  $\mathbf{I}$  is the identity matrix and  $\mathbf{0}$  is either a null matrix or vector. Once Eq (9) is solved for  $T_b$  and  $T_g$ , the adimensional g-function is simply obtained with:

$$\tilde{g}(t) = 2\pi k_s (T_b - T_g) / \tilde{q} \quad (10)$$

## ARTIFICIAL NEURAL NETWORKS

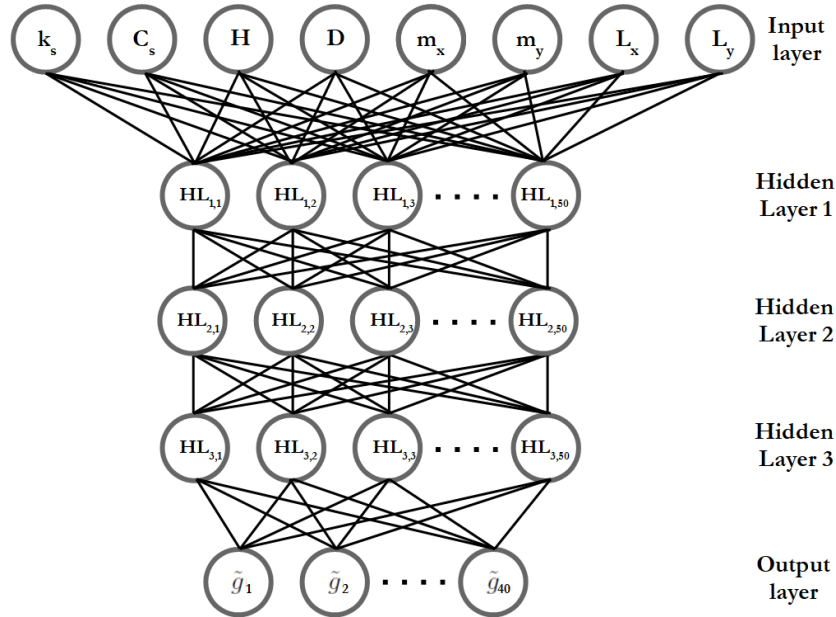


Figure 1 Artificial neural network with 3 hidden layers used to approximate a g-function.

Artificial neural networks come in many flavors, but all rely on the same building block. Indeed, artificial neurons (Rosenblatt 1958) are at the core of their learning ability and are connected with each other according to different architectures. In an ANN, each neuron gets its inputs from all the neurons of the previous layer and passes its output to all the neurons of the next layer. Every time an artificial neuron receives some signals as input, it must decide, just like its biological counterpart, which output signal to transmit to the next layer as a result. This behavior

can be summarized by Eq (11):

$$y_k = \varphi \left( \sum_{j=1}^m w_{kj} x_j + b_k \right) \quad (11)$$

where  $k$  is the neuron index,  $x_j$  is the input provided by neuron  $j$ ,  $w_k$  and  $b_k$  are the neuron' weight and bias respectively. The resulting summation is submitted to an activation function  $\varphi$  to introduce an element of non-linearity between the input and output of the neuron. There are many types of activation function in the literature that are meant to be used for specific tasks. As of today, the choice of an activation function is empirical at best and is usually driven by the experience of the user, by trial and error and by the type of phenomenon being modelled. In this work, the linear function as well as the hyperbolic tangent sigmoid function were chosen after a few tests. The linear function, linking the last layer to the output of an ANN, is a prerequisite to the universal approximation theorem. The other function, linking the input and all the hidden layers, was selected for its transitional nature after some testing.

As mentioned earlier, the universal approximation theorem only requires a single hidden layer of a finite number of neurons. However, the arbitrary number of neurons, which needs to be fine tuned by the user, can grow quite large for the modelling of complex processes such as heat transfer in a GHE. To circumvent part of the problem, we resorted to use three hidden layers as illustrated in Fig 1. By adopting such an architecture, an ANN can be trained to performed more complex tasks while having a smaller total number of neurons. This results in faster training times.

## METHODOLOGY

The objective of this paper is to train an ANN to approximate a g-function over a period of 100 years for GHE made of 1 to 10 boreholes placed on regular grids. There are 27 unique ways in which boreholes can be distributed between rows and columns on regular grids of a GHE made up of 1 to 10 boreholes. For example, 3 boreholes can be placed on grids in two different ways,  $1 \times 3$  and  $3 \times 1$ , while 8 boreholes can be placed in four different ways,  $1 \times 8$ ,  $2 \times 4$ ,  $4 \times 2$  and  $8 \times 1$ . In addition, the ANN must be able to approximate accurately the g-function while subjected to varying values of ground thermal conductivity ( $k_s$ ), volumetric heat capacity ( $C_s$ ), borehole length ( $H$ ), buried depth ( $D$ ), boreholes per row ( $m_x$ ), boreholes per column ( $m_y$ ), row spacing ( $L_x$ ) and column spacing ( $L_y$ ) (Fig 2) taken between the intervals presented in Table 1. Note that  $m_x$  and  $m_y$  are integers that can both vary from 1 to 10 under the constraint that  $(m_x \times m_y) \leq 10$ . The input parameters should then be selected so that the total number of boreholes in the resulting GHE does not exceed 10. Finally, all boreholes share the same radius of 0.075 m.

**Table 1. Ranges of inputs parameters of the ANN**

Parameters (units)	Units	Lower bound	Upper bound
$k_s$	W/ (m °C)	1.0	3.5
$C_s$	MJ/ (m <sup>3</sup> °C)	1.0	3.5
$H$	m	50	250
$D$	m	0	2.5
$m_x$	-	1	10
$m_y$	-	1	10
$L_x$	m	1	5
$L_y$	m	1	5

### Inputs of the ANN

The ANN developed uses a total of 8 inputs as illustrated in Fig 1. To properly train the ANN, all input parameters were centered and scaled between the interval [0 1]. This ensures, for example, that some combinations of high inputs parameters won't saturate a neuron that has otherwise been trained to respond to small values.

### Outputs of the ANN

The number of output neurons has a direct impact on the training time as well as the format in which the g-functions are produced by the ANN. To limit their number without losing information on the g-functions, the geometrical sampling scheme proposed by Marcotte and Pasquier (2008) was implemented. The progression that was selected starts at one hour and doubles every two time steps, allowing 100 years to be covered in 40 steps. The output layer then contains 40 neurons, each one corresponding to the g-function of a given time step. The whole g-function can later be reconstituted at every time step using cubic spline interpolation. Again, the training data set was normalized for the training.

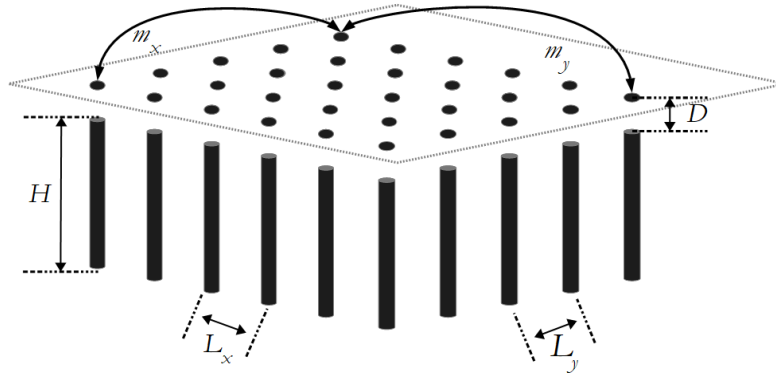


Figure 2 Geometrical parameters used as inputs.

### Architecture and Training Algorithm

After some testing, we decided to use an ANN containing three hidden layers of 50 neurons each. This was found to be a good trade-off between training time and accuracy. We also decided to train our ANN with the scaled conjugate gradient with backpropagation (Hagan, et al. 1996) using the mean squared residuals (mse) as objective function. Random initial weights and biases were also used. As a rule of thumb, about 1000 training examples are needed per parameter to train an ANN. In our case, we observed that the number of different regular grids is the force that drives mostly the training. This is because the ANN can learn easily the impact of the first four parameters ( $k_s$ ,  $C_s$ ,  $H$  and  $D$ ) no matter the GHE configuration. On the other hand, the 1 to 10 boreholes can be arranged in 27 different regular grids and their impact is hard to predict. Therefore, their contribution to the output is harder to learn to approximate for the ANN. Hence, we used the block matrix formulation presented earlier to generate a data set of 27,000 g-functions used for training.

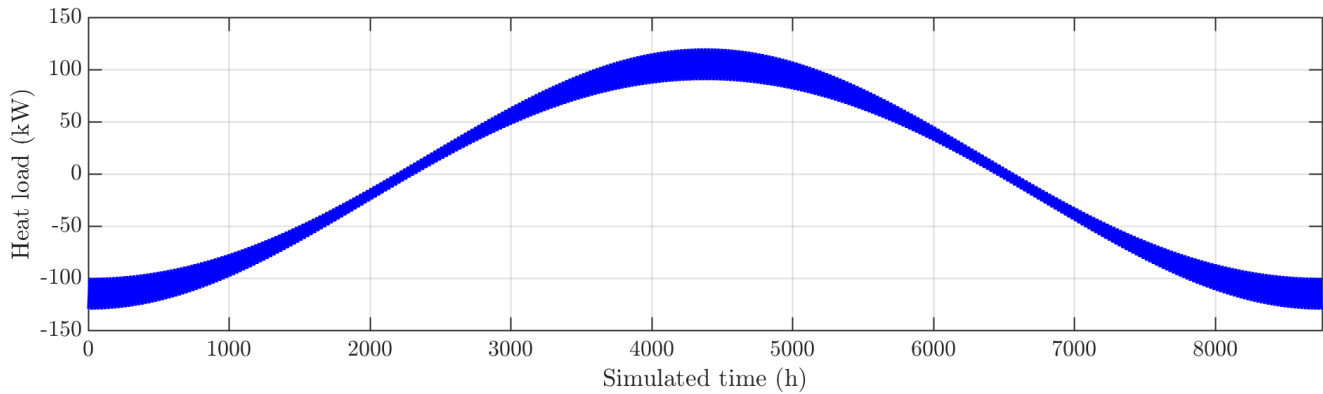


Figure 3 Hourly heat load signal used to verify the accuracy of the ANN.

It is important to note that the accuracy of an ANN is not verified using the same training data, as overfitting can sometimes exaggerate the accuracy of the network and lead to poor generalizations capabilities. Therefore, the trained ANN is tested using 27 independent g-functions generated after the training phase, one for each different grid arrangement. After comparison, both the approximated and reference g-functions are convolved with the same heat load signal (Fig 3) in the spectral domain to calculate fluid temperatures throughout the simulation duration (100 years).

## RESULTS AND DISCUSSION

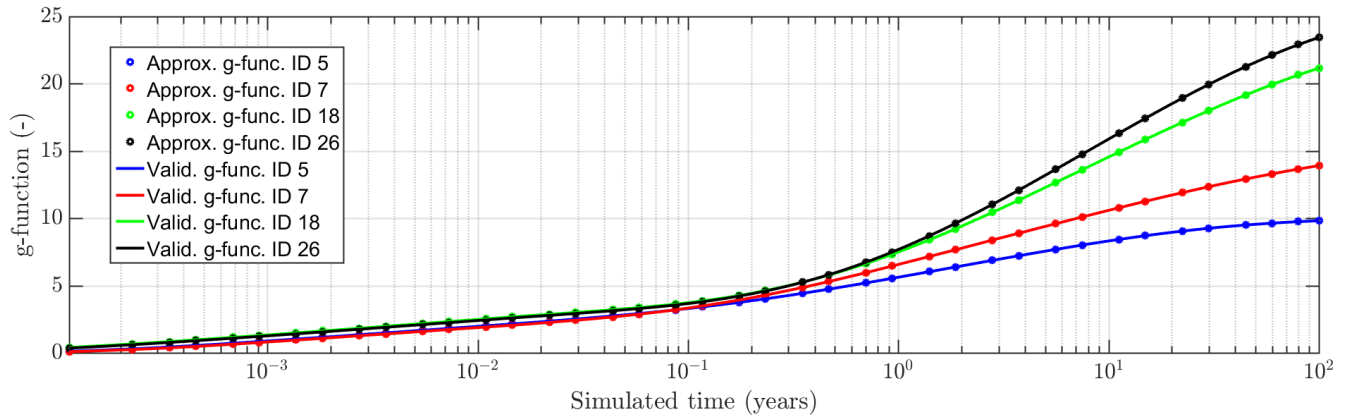


Figure 4 Comparison between the reference and ANN g-functions of ID 5, 7, 18 and 26.

Figure 4 compares a few g-functions obtained with the block matrix formulation and those produced by the ANN after its training for four sets of input parameters chosen randomly. The four g-functions shown are coming from a set of 27 g-functions (1 per geometrical layout) and are the cases having the worst fits. Nevertheless, one can see that the g-functions constructed by the ANN are visually practically superimposed.

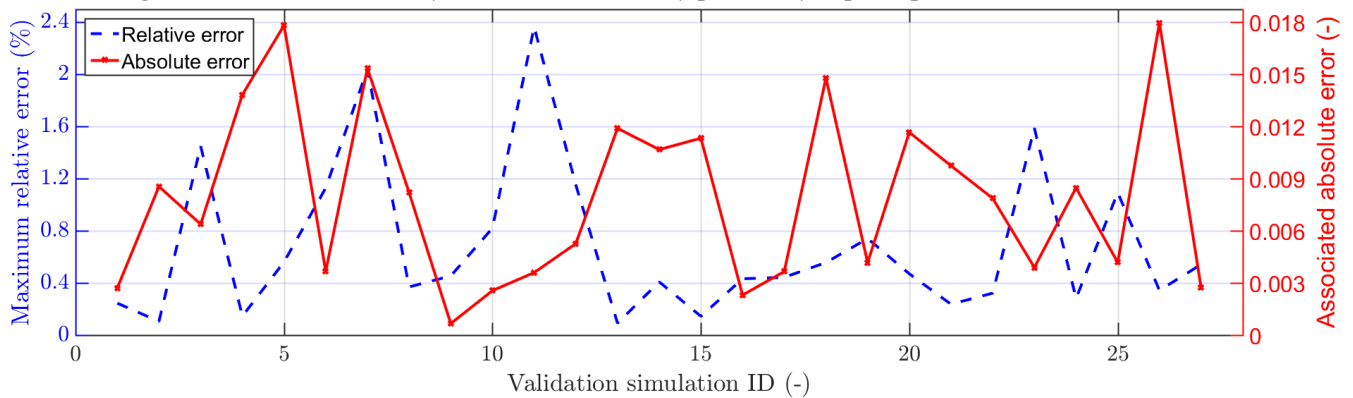


Figure 5 Maximum relative error and corresponding absolute errors committed by the ANN.

Figure 5 illustrates the error made by the ANN when approximating the g-functions for each of the 27 possible layouts with input parameters chosen randomly. From Fig 5 it is clear that no correlation is observed between

the error produced by the ANN and the layout of the ground heat exchanger. Also, we can see that even in the worst cases (ID 5, 7, 18 and 26), the absolute error is small with maximum relative errors that barely exceed two percent. Although it is not intrinsically bad, the universal approximation theorem suggested that a better accuracy could be achieved with the ANN we implemented. To find out why, we observed that the maximum error occurred mostly (10 times out of 27) at the first time step. Moreover, in most cases the errors are decreasing from the second neuron onwards as illustrated in Figure 6. This was a proof that 27,000 simulations are probably not enough to perfectly train our ANN and that a larger training set could have better its response on the first neuron.

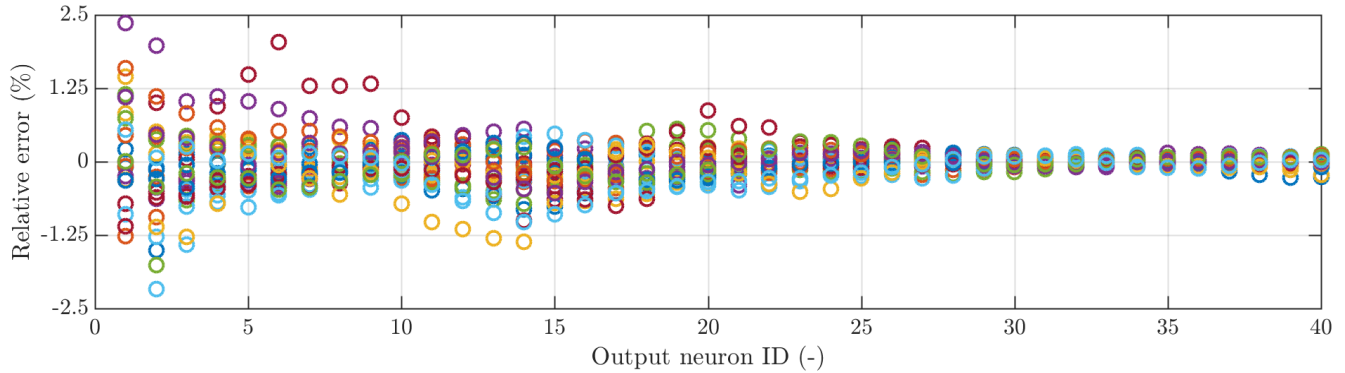


Figure 6 Relative error per output neuron for all 27 approximated g-functions.

Figure 7 compares the resulting fluid temperatures that were obtained after convolving in the spectral domain the approximated g-functions and the reference g-functions to the same heat load signal. Fig 7 clearly shows that the relative errors on fluid temperatures are up to an order of magnitude less important than the ones on g-functions. Indeed, most of the simulations are within an error of 0.1 °C and never exceed 0.3 °C, which is more than enough for sizing and simulation purposes. The sinusoidal heat load signal and the convolution process probably damp the error made at the first time step of the g-function, which results in a lower error.

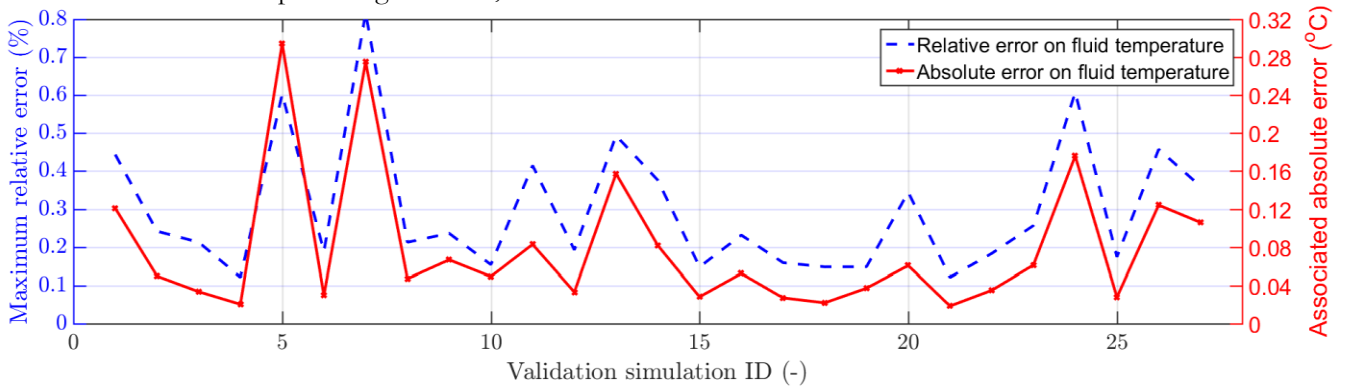


Figure 7 Maximum relative and absolute errors committed on the fluid temperatures after convolution with the heat load signal of Figure 3.

Finally, we emphasized the computational gain associated with the use of the ANN. Indeed, only 14 milliseconds are required to construct 27 g-functions over a period of 100 years (1 per configuration). By comparison,

construction of the same 27 g-functions by the block matrix formulation is achieved in 11.74 seconds, a ratio of 1 to 838. Such efficiency could be useful when optimizing ground-coupled heat pump systems or assessing the uncertainty of a design, which could require several thousands of g-function evaluations.

## CONCLUSION

In this article, we demonstrated the feasibility of using artificial neural networks to approximate g-functions of ground heat exchangers varying from 1 to 10 boreholes placed on regular grids. We covered the architecture of the network we developed, how we trained it and its efficiency and accuracy when compared to a benchmark of g-functions calculated with an independent method. Our results show that a g-function of a 100 years duration can be calculated almost instantaneously with good accuracy. Moreover, we showed that the maximum imprecision of the ANN g-functions is very localised at the first time step, which could eventually be improved by additional works. The methodology developed in this article can easily be adapted to construct and train artificial neural networks for any number of boreholes.

## BLOCK MATRIX NOMENCLATURE

$\Lambda$	=	Constraint vector (-)	$k$	=	Thermal conductivity (W/m°C)
$C$	=	Volumetric heat capacity (MJ/m <sup>3</sup> °C)	$m$	=	Number of boreholes (-)
$D$	=	Boreholes' buried-depth (m)	$n$	=	Number of time steps (-)
$f, \mathbf{f}$	=	Incremental heat-flux function (W/m)	$\tilde{q}$	=	Unit heat load (1 W/m)
$g$	=	Finite-line source solved under $\tilde{q}$ (m°C/W)	$t$	=	Simulation time (s)
$\mathbf{G}$	=	Lower-triangular matrix of $g$ (°C)	$\mathbf{T}$	=	Temperature (°C)
$\tilde{g}$	=	g-function (-)	$\Delta T, \Delta \mathbf{T}$	=	Thermal perturbation (°C)
$H$	=	Borehole length (m)	$r$	=	Distance (m)
$\mathbf{I}$	=	Identity matrix (-)			

## ANN NOMENCLATURE

$\Sigma$	=	Combination function (-)	$x$	=	Neuron's input (-)
$\varphi$	=	Activation function (-)	$y$	=	Neuron's output (-)
$b$	=	Neuron's bias (-)	$X, Y$	=	Boreholes' cartesian coordinates (m, m)
$w$	=	Neuron's weight (-)	$\theta, r$	=	Boreholes' polar coordinates (rad, m)

## Subscripts

$b$	=	Borehole	$j$	=	Emitting borehole
$g, s$	=	Surrounding ground, soil	$j \rightarrow i$	=	Influence of borehole $j$ on borehole $i$
$i$	=	Receiving borehole	$m$	=	Number of input in a neuron

## REFERENCES

- Arat, H. and O. Arslan. 2017. *Optimization of district heating system aided by geothermal heat pump: A novel multistage with multilevel ANN modelling*. Applied Thermal Engineering 111: 608-623.
- Atam, E. and L. Helsen. 2016. *Ground-coupled heat pumps: Part 1- Literature review and research challenges in modeling and optimal control*. Renewable and Sustainable Energy Reviews 54: 1653-1667.



- Atam, E. and L. Helsen. 2016. *Ground-coupled heat pumps: Part 2- Literature review and research challenges in optimal design*. Renewable and Sustainable Energy Reviews 54: 1668-1684.
- Atam, E., D. Patteeuw, S. Antonov and L. Helsen. 2016. *Optimal control approaches for analysis of energy use minimization of hybrid ground-coupled heat pump systems*. IEEE Transactions on Control Systems Technology 24(2): 525-540.
- Claesson, J. and S. Javed. 2011. *An Analytical Method to Calculate Borehole Fluid Temperatures for Time-scales from Minutes to Decades*. ASHRAE Transactions 117(2):279-288.
- Cimmino, M. and M. Bernier. 2014. *A semi-analytical method to generate g-functions for geothermal borefields*. International Journal of Heat and Mass Transfer. 70: 641-650.
- Cimmino, M., M. Bernier and F. Adams. 2013. *A contribution towards the determination of g-functions using the finite line source*. Applied Thermal Engineering. 51(1-2): 401-412.
- Cybenko, G. 1989. *Approximation by superpositions of sigmoidal functions*. Budapest, Hungary: Eötvös Loránd University.
- Dusseault, B., P. Pasquier and D. Marcotte. 2017. *A block matrix formulation for efficient g-function construction*. Renewable Energy In Press, Accepted Manuscript.
- Esen, H., M. Inalli, A. Sengur and M. Esen. 2008. *Performance prediction of a ground-coupled heat pump system using artificial neural networks*. Expert Systems with Applications 35 (4): 1940-1948.
- Eskilson, P. 1987. *Thermal Analysis of heat extraction borehole*. Lund, Sweden: University of Lund.
- Fannou, J.-L. C., C. Rousseau, L. Lamarche and S. Kajl. 2014. *Modeling of a direct expansion geothermal heat pump using artificial neural networks*. Energy and Buildings 81: 381-390.
- Gang, W., J. Wang and S. Wang. 2014. *Performance analysis of hybrid ground source heat pump systems based on ANN predictive control*. Applied Energy 136: 1138-1144.
- Hagan, M. T., H. B. Demuth., M. H. Beale and O. De Jesus. 2014. Neural network design 2<sup>nd</sup> edition. Stillwater, Oklahoma. Oklahoma State University.
- Hornik, K., M. Stinchcombe and H. White. 1989. *Multilayer feedforward networks are universal approximators*. Neural Networks 2 (5): 359-366.
- 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.
- Marcotte, D. and P. Pasquier. 2008. *Fast fluid and ground temperature computation for geothermal ground-loop heat exchanger systems*. Geothermics. 37 (6): 651-665.
- Marcotte, D. and P. Pasquier. 2014. *Unit-response function for ground heat exchanger with parallel, series or mixed borehole arrangement*. Renewable Energy. 68: 14-18.
- Pasquier, P., A. Zarrella and R. Labib. 2018. Near-instant construction of short-term g-functions by artificial neural network. TO BE SUBMITTED.
- Rosenblatt, F. 1958. *The Perceptron: A Probabilistic Model for Information Storage and Organization in The Brain*. Psychological Review 65 (6): 386-407.
- Sun, W., P. Hu, F. Lei, N. Zhu and Z. Jiang. 2015. *Case study of performance evaluation of ground source heat pump system based on ANN and ANFIS models*. Applied Thermal Engineering 87: 586-594.
- Yabanova, I. and A. Keeba. 2013. *Development of ANN model for geothermal district heating system and a novel PID-based control strategy*. Applied Thermal Engineering 51 (1-2): 908-916.