# ADVANCED GAUSSIAN PROCESSES FOR THE SURROGATE MODELLING OF ENERGY SYSTEMS

Charles Maragna, Jérémy Rohmer, Romain Chassagne

# ADVANCED GAUSSIAN PROCESSES FOR THE SURROGATE MODELLING OF ENERGY SYSTEMS

Charles Maragna[1]*, Jérémy Rohmer[1], Romain Chassagne[1]

[1]BRGM, Orléans, France

*Corresponding Author: c.maragn@brgm.fr

## ABSTRACT

Surrogate models are efficient and accurate approximations of computationally expensive numerical models, derived from the statistical analysis of a limited number of model evaluations. Their primary purpose is to mitigate the computational burden. Surrogates are used for a wide variety of tasks such as sensitivity analysis (identifying the most influential parameters or parameter combinations), model calibration, uncertainty propagation and system optimization. Among other techniques, Gaussian Processes (GP) are especially appealing since they lie on sound theoretical foundations, are known to be flexible, rather accurate and, to some extent, they can be interpreted. This paper investigates to which extent advanced variants of GPs can improved the prediction when an exponentially increasing number of parameter combinations that must be explored. Multi-start strategy for the GP training, alternative objectives for GP training and linear embedding (i.e. considering a linear transformation of the variables) are investigated. Throughout this study, a dataset of 4589 time-dependent simulations of a geothermal absorption cooling system with 17 sizing variables is used. It turns out that none of these strategies can improve the Cross-Validation Root Mean Square Error when compared to standard GP training with Log Marginal Likelihood maximization, GP training with a single deterministic starting point, and no linear embedding. Applying standard GP algorithms and approaches seems sufficient for the surrogate modelling of energy systems, at least for the considered dataset.

## 1 INTRODUCTION

### 1.1 Surrogate modelling

Tasks such as sensitivity analysis, model calibration, uncertainty propagation or system optimization requires many evaluations of costly numerical models (Westermann and Evins, 2019). Surrogate modelling refers to a set of statistical techniques to build cheap, fast-to-run approximations of a model output $\boldsymbol{y}$ as a function of the $D$ model parameters represented by $\boldsymbol{x}$. This key output from the physical model can be, for instance, the system efficiency over its whole lifespan, as computed from a time-dependent numerical model of the system. The application of surrogates to energy in buildings has been a booming research domain since the mid-2010s (Westermann and Evins, 2019).

These parameters of the physical model, whose influence is to be investigated, serve as the input variables of the surrogate. We further consider that $N$ evaluations of the physical model are available. Unless otherwise specified, the sample $\boldsymbol{x}$ has been generated with the Latin Hyper Square (LHS), a popular method with enhanced space filling properties (McKay et al., 1979). Our goal is to build the "best" prediction of the quantity of interest $\boldsymbol{y}$ for any new design point $\boldsymbol{x}$, based on the information available at hand. In this paper, we will focus on Gaussian Processes (GP) (Rasmussen and Williams, 2006), a technique also known as *kriging* in the field of geostatistics. GP are especially appealing since

they are known to be flexible, rather accurate and, to some extent, they can be interpreted. They further provide an estimation of the uncertainty associated with any prediction.

### 1.2 Gaussian Processes: Standard approach

The function $y(\boldsymbol{x})$ to be estimated is assumed to be a realization of a centered square-integrable stochastic process Z and some Gaussian noise $\varepsilon$:

$$Y(\boldsymbol{x}) = Z(\boldsymbol{x}) + \varepsilon \tag{1}$$

The noise is assumed to follow a normal distribution $\varepsilon \sim \mathcal{N}(0, \sigma_n)$, where $\sigma_n$ is the noise standard deviation. Note that some variants of GP add a parametrized, deterministic function $\mu(\boldsymbol{x})$, however here we will focus on *simple kriging* (that is $\mu(\boldsymbol{x})=0$).

At the heart of the GP lies the covariance function or *kernel*. The kernel $k_{\boldsymbol{\theta}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = \mathrm{cov}(y_1, y_2)$ expresses the similarity between a pair of points $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$. The usual kernels are parametrized by a set of parameters $\boldsymbol{\theta}$ or *hyperparameters*. With the available dataset a covariance matrix $\mathbf{K}_{\boldsymbol{\theta}}$ is computed as:

$$\mathbf{K}_{\boldsymbol{\theta}} = \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1|\boldsymbol{\theta}) + \sigma_n & \dots & k(\boldsymbol{x}_1, \boldsymbol{x}_N|\boldsymbol{\theta}) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}_1|\boldsymbol{\theta}) & \dots & k(\boldsymbol{x}_N, \boldsymbol{x}_N|\boldsymbol{\theta}) + \sigma_n \end{bmatrix} \tag{2}$$

The prediction $y_{new}$ at a new point is given by:

$$y_{new} = \boldsymbol{k}_{new}{}^{\intercal} \cdot \boldsymbol{\alpha} \quad \text{where } \boldsymbol{\alpha} = \mathbf{K}_{\boldsymbol{\theta}}{}^{-1} \cdot \boldsymbol{y} \tag{3}$$

Where $\boldsymbol{k}_{new}{}^{\intercal} = \{k_{\boldsymbol{\theta}}(\boldsymbol{x}_{new}, \boldsymbol{x}_1) \quad \dots \quad k_{\boldsymbol{\theta}}(\boldsymbol{x}_{new}, \boldsymbol{x}_N)\}$. The variance $\mathbb{V}(\boldsymbol{x}_{new})$ associated with this prediction is as follows:

$$\mathbb{V}(\boldsymbol{x}_{new}) = \boldsymbol{k}(\boldsymbol{x}_{new}, \boldsymbol{x}_{new}) - \boldsymbol{k}_{new}{}^{\intercal} \cdot \mathbf{K}_{\boldsymbol{\theta}}{}^{-1} \cdot \boldsymbol{k}_{new} \tag{4}$$

The 95% confidence interval is merely approximately equal to $1.96 \cdot \sqrt{\mathbb{V}(\boldsymbol{x}_{new})}$.

Anisotropic kernels, sometimes referred to as "Automatic Relevance Determination" (ARD) kernels, are of special interest since they can capture the heterogeneous influence of variables. Such kernels are expressed as a function of $r$:

$$r = r(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sqrt{\sum_{j=1}^{D} \left( \frac{x_1{}^{(j)} - x_2{}^{(j)}}{\ell_i} \right)^2} \tag{5}$$

Where $x_i{}^{(j)}$ accounts for the $j^{\text{th}}$ variable of data point $\boldsymbol{x}_i$, $\ell_i$ ($i$=1, … $D$) is the correlation length for the variable $i$, which indicates how far apart the input values can be for the response values to become uncorrelated along axis $i$. In other words, the smaller the value of $\ell_i$, the more influential the variable $x_i$.

Unless otherwise specified, we use the popular Matérn5/2 kernel since it is twice differentiable, making it good at capturing physical processes:

$$k_{\boldsymbol{\theta}}(\boldsymbol{x}_1, \boldsymbol{x}_2) = {\sigma_f}^2 \left(1 + \sqrt{5}r + \frac{5}{3}r^2\right) \exp(-\sqrt{5}r) \tag{6}$$

The standard procedure to determine the relevant values for the hyperparameters $\boldsymbol{\theta}$ is to numerically maximize the Log Marginal Likelihood function (LML) (Roustant et al., 2012). The LML function reads (see (Rasmussen and Williams, 2006) Eq. (5.8)):

$$\mathrm{LML}(\boldsymbol{\theta}) = -\frac{1}{2}\log(\det \mathbf{K}_{\boldsymbol{\theta}}) - \frac{1}{2}\boldsymbol{y}^{\intercal}\mathbf{K}_{\boldsymbol{\theta}}{}^{-1}\boldsymbol{y} - \frac{N}{2}\log(2\pi) \tag{7}$$

Since the physical model is assumed to be deterministic, $\sigma_n$ is set to a very small value $\sigma_n = 10^{-3} \ll 1$. For the Matérn5/2 kernel given in Eq. (6), $\boldsymbol{\theta}$ contains *D*+1 hyperparameters to be optimized: the signal deviation $\sigma_f$ and the *D* correlations lengths $\ell_i$.

A classical property of a GP is that the prediction interpolates the training dataset if $\sigma_n \ll 1$. This makes cross-validation necessary to estimate the quality of the GP. Cross-validation is a procedure where a part 1/*n* of the data is randomly selected and left apart (here *n* = 10), acting as a test sample (Hastie et al., 2009). A GP is fitted on a fraction (1-1/*n*) of the sample and the procedure is repeated *n* times by permuting the samples. As a metric for the estimation of the GP quality, we choose the Root Mean Square Error (RMSE):

$$\mathrm{RMSE} = \sqrt{\sum_{j=1}^{N} \frac{(y_{data} - y_{GP})^2}{N}} \tag{8}$$

Where $y_{data}$ and $y_{GP}$ are respectively the output from the physical model and the predictions from the cross-validated GP.

### 1.3 Objective of this paper

The objective of this paper is to investigate whether recent developments in GP variants can improve prediction accuracy (see. e.g. (Binois and Wycoff, 2022)) compared to the standard approach described in §1.2. The paper is structured as follows: in section 2, we disclosed the GP numerical implementation and a reference dataset used throughout the paper. In section 3, we investigate GP variants related to numerical strategy, training objective and linear transformation of the variables. A brief conclusion of this work is provided in section 4.

## 2 MATERIAL AND METHODS

### 2.1 Numerical implementation for GP training

The whole computations have been done in MATLAB R2024b. The Gaussian Processes are built with the *fitrgp* function of the Statistics and Machine Learning Toolbox. It natively implements the features described in §1.2. The LML (Eq. (7)) is maximized with a quasi-Newton solver. To speed up this optimization, *fitrgp* takes advantage of the analytical expression of the derivative of the LML with respect to each hyperparameter $\theta_j$ (see (Rasmussen and Williams, 2006) Eq. (5.9)):

$$\frac{\partial LML}{\partial \theta_j} = \frac{1}{2}\mathrm{tr}\left(\left(\boldsymbol{\alpha}\cdot\boldsymbol{\alpha}^{\intercal} - \mathbf{K}_{\boldsymbol{\theta}}{}^{-1}\right)\frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \theta_j}\right) \tag{9}$$

The analytical expression of $\frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \theta_j}$ for standard kernels such as Matérn5/2 can be found in (Kim, 2021).

The starting point $\boldsymbol{\theta}_0^{ref}$ (initial guess) of the LML optimization is equal to the standard deviation of $\boldsymbol{x}_i$ for the correlation length $\ell_i$ and the mean value of $\boldsymbol{y}$ for the signal standard deviation $\sigma_f$. To ensure that $\ell_i$ and $\sigma_f$ are positive, the optimization uses an unconstrained variable $\boldsymbol{\varphi} = \log(\boldsymbol{\theta})$.

Unless otherwise specified, the "standard approach" for GPs has to be understood as the theoretical approach described in §1.2 along with the numerical implementation of GP training described in this paragraph.

All computations are run on a computer equipped with a 12th Gen Intel® Core(TM) i5-12500H and 32 Go RAM.

### 2.2 Reference dataset

Throughout the paper, we use a dataset generated from a time-dependent model of a geothermal absorption chiller in a tropical climate run over one year of operation (Maragna et al., 2024). This system uses a geothermal stream (typically in temperature range 80–110 °C) to power a single-effect absorption chiller that produces space cooling. The fluid leaving the generator of the absorption system is further cooled through a heat exchanger to prepare Domestic Hot Water (DHW) at 40 °C. If the sorption chiller is not able to fully meet the cooling load, an Air-Source Heat Pump (ASHP) is turned on. We focus on two outputs from the model:

- $W_{el}^{*}$ [-]: the normalized electric consumption of the system, i.e. the amount of electricity consumed by the whole system (for the pumps, ASHP, etc.) [J] divided by the amount of delivered cooling and DHW [J],
- $Q_{sorp}^{ev\,\sim}$ [-]: the normalized amount of cooling produced by the absorption (subscripted *sorp*) at its evaporator (superscripted *ev*) chiller [J] relative to the total cooling requirement [J]. Note that the missing fraction of cooling is covered by the ASHP.

The influence of $D = 17$ model parameters upon $W_{el}^{*}$ and $Q_{sorp}^{ev\,\sim}$ has been investigated (see (Maragna et al., 2024), Table 7 for further information). These variables $\boldsymbol{x}$ are related to the characteristics of the thermal demand and distribution, geothermal resource, equipment and control strategy. For convenience, each variables $x_i$ has been normalized in the range [0;1]. A total of $N = 4589$ model has been evaluated.

As we want $Q_{sorp}^{ev\,\sim}$ to strictly remain in the range [0; 1], we apply a well-known trick where the GP is trained on $\Phi^{-1}\left(Q_{sorp}^{ev\,\sim}\right)$ rather than $Q_{sorp}^{ev\,\sim}$, with $\Phi^{-1}$ the probit function (i.e. the inverse cumulative distribution function of a standard normal random variable) (Swiler et al., 2020). Similarly, to ensure that the predictions of $W_{el}^{*}$ are positive, its GP is trained on $\sqrt{W_{el}^{*}}$ rather than $W_{el}^{*}$. Note that, before evaluating any prediction using Eq. (3), the reciprocal transformations unwrap $Q_{sorp}^{ev\,\sim}$ and $W_{el}^{*}$.

### 2.3 Application of standard GP to the dataset

We first apply the standard GP reminded in §1.2 with the numerical implementation described in §2.1. For both datasets, the 10-fold CV-RMSE as a function of the dataset size is depicted in Figure 1, considering from 300 to 4589 training points. After a sharp improvement of the surrogate quality, the RMSE barely decreases with increasing dataset size. This is associated with a stabilization of correlation lengths $\ell_i$ (see Figure 2). In Figure 2, we plot the normalized value of the inverse of the squared correlation lengths ${\ell_i}^{-2} / \sum_{i=1}^{D} {\ell_i}^{-2}$ to emphasize small values $\ell_i$ associated with influential variables. For both GP, the most influential variables are the geothermal temperature, flow rate and the nominal capacity of the absorption chiller (see (Maragna et al., 2024) for further discussion). For the full dataset, the RMSE is equal to 0.0111 for $W_{el}^{*}$ and to 0.0338 for $Q_{sorp}^{ev\,\sim}$.

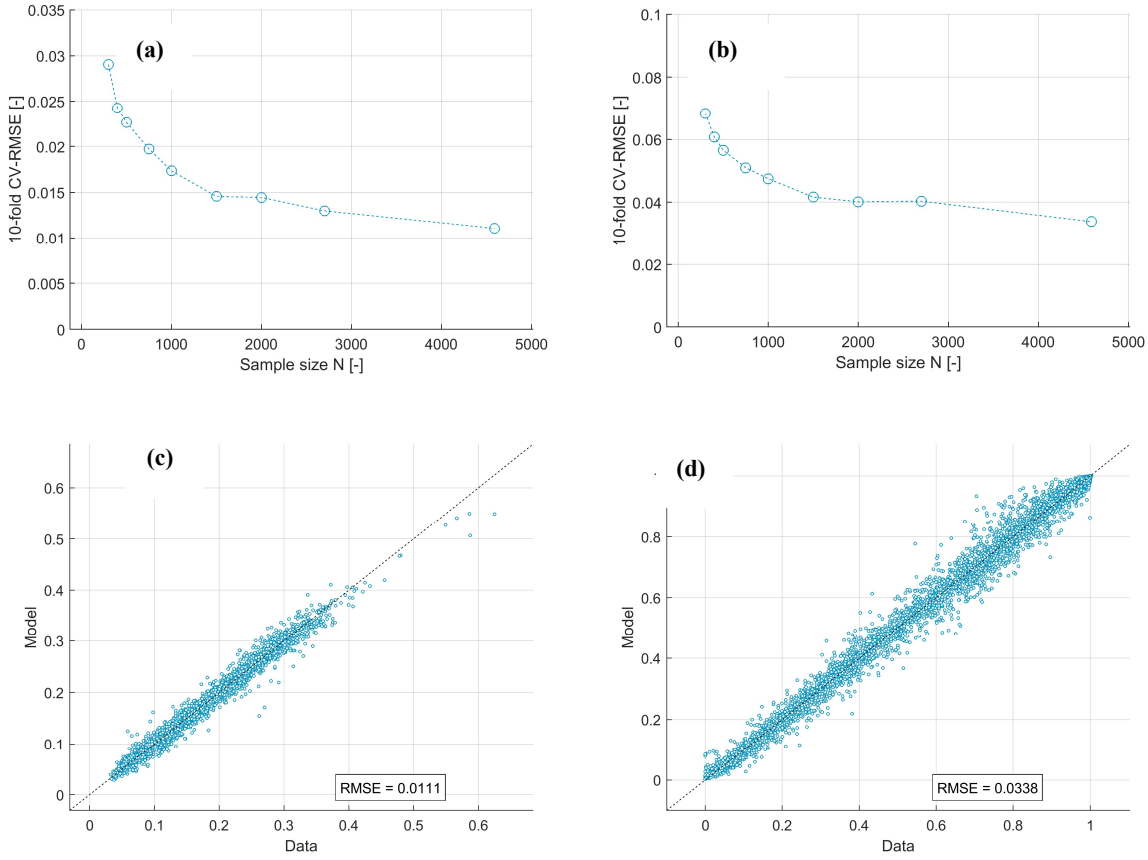


**Figure 1:** 10-fold Cross-Validation Root Mean Square Error (CV RMSE) as function of the dataset size for **(a)** $W_{el}^{*}$ and **(b)** $Q_{sorp}^{ev\ \sim}$. Data vs. model prediction from 10-fold CV for **(c)** $W_{el}^{*}$ and **(d)** $Q_{sorp}^{ev\ \sim}$, considering the full dataset ($N$=4589 points)

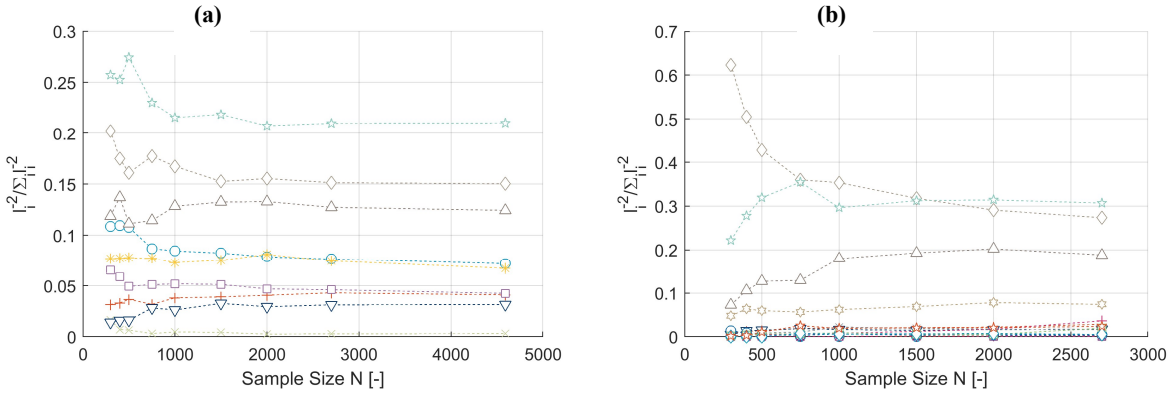


**Figure 2:** Normalized squared inverse of correlation lengths $\ell_i$ as a function of the dataset size for **(a)** $W_{el}^{*}$ and **(b)** $Q_{sorp}^{ev\ \sim}$. Each line is for one predictor $x_i$ ($i$=1, …, 17).

## 2.4 Metrics for the comparison

To compare the results, we use two metrics:

- The duration of computation $t$ [s], including data preparation, surrogate training and cross-validation. Note that the fairness of comparison, the folds are prepared in advance so that for a given dataset size, all configurations see the same folds,

- The 10-fold Cross-Validation Root Mean Square Error, as computed by Eq. (8).

# 3 RESULTS AND DISCUSSIONS

## 3.1 Numerical strategy

### 3.1.1 Multiple starting points

The LML (Eq. (7)) may exhibit multiple local maxima. According to Rasmussen et Jaeger, "*there is no guarantee that the marginal likelihood does not suffer from multiple local maxima. Practical experience with simple covariance functions seem to indicate that local maxima are not a devastating problem, but certainly they do exist. In fact, every local maximum corresponds to a particular interpretation of the data. (...) Care should be taken that one doesn't end up in a bad local optimum.*"

We investigate to which extent the LML maximum is sensitive to the initial guess of hyperparameters $\boldsymbol{\theta}_0$, and if there are some LML maximum that cannot be reached from the initial guess $\boldsymbol{\theta}_0^{ref}$. To do so, we generate 10 starting points $\boldsymbol{\varphi}_0 = \log(\boldsymbol{\theta}_0^{ref}) + \boldsymbol{a}$, where $\boldsymbol{a}$ is vector whose elements are randomly generated from a uniform distribution in the range [-3, 3]. The LML is optimized for each starting point, and the optimization with the highest LML is retained. We do this experiment for sample size $N$ = 300, 400, 500, 750 and 1000.

Figure 3 clearly shows that the multi-start strategy provides no improvement for the 10-fold CV-RMSE. Indeed, among the 10 starting points $\boldsymbol{\varphi}_0$, only 1 to 3 points (depending on the sample size) have converged to the same optima as the reference starting point $\boldsymbol{\varphi}_0^{ref}$. The other points are stuck in local maxima characterized by LML values much lower than the optimal one. For instance, for $N$ = 1000 points, only 2 points converge to the same value of $\boldsymbol{\varphi}_0^{ref}$ (LML = +2287.9), while the other ones are stuck at LML = -598.06. Therefore, for the dataset considered in this paper there is no local optima; the empirical rule to use $\boldsymbol{\theta}_0^{ref}$ seems well grounded.

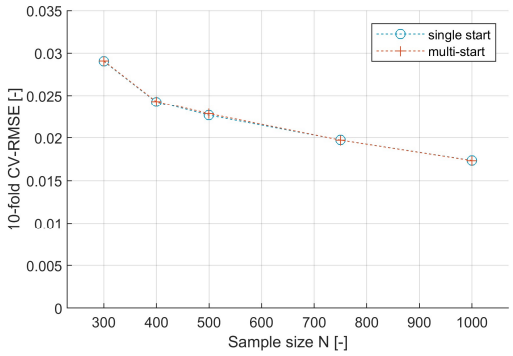


**Figure 3:** Comparison of the 10-fold CV-RMSE for a single start and multi-start strategy.

### 3.1.2 Stochastic optimization

The training of a GP requires computing $\mathbf{K}_{\boldsymbol{\theta}}{}^{-1}$ whose computational cost is typically proportional to the cube of the dataset size ($\approx \mathcal{O}(N^3)$). Therefore, standard GP scale badly with large datasets, which is the main drawback of GP. For the full dataset used in this paper (i.e. $N$ = 4589 points), one evaluation of the LML (Eq. (7)) is typically in the range 5–10 s. During the optimization process, the LML has to be evaluated many times, typically in the order of a few hundred to several thousand times before an optimal value is found. Training and cross-validating the GP of $W_{el}{}^*$ requires approximatively 1.5 h when using the full dataset (see Figure 4).

*Stochastic mini-batch optimization* is emerging in the GP community to tackle this challenge (see e.g. (Chen et al., 2022)). At each iteration, these optimization techniques select a random subsample of $n$ points (also called a *mini-batch*) to evaluate the training objective and its gradient and then update the optimization variables (here the GP hyperparameters) based on that gradient. Stochastic optimization is a standard practice to train Neural Networks. Here we implement Adam (Kingma and Ba, 2017), which is known to be straightforward to implement, computationally efficient, has little memory requirements, and is well suited for problems that are large in terms of data and/or parameters. We use 10,000 iterations with a mini-batch size $n$ = 32 (similar results were obtained with $n$ = 64 and $n$ = 128), and a canonical "learning rate" set to $10^{-2}$ (the learning rate is the proportionality to be applied between the change in the training objective and the change in the variables). One expect that, by dividing the sample size by a factor $\approx 10^2$, one decreases the time needed for LML computation by a factor $10^6$. Note that setting 10,000 iterations is sufficient to ensure a LML plateau has been reached. Once the LML has been optimized, the matrix $\mathbf{K}_{\boldsymbol{\theta}}$ used for the predictions (Eq. (3)) is computed with the full dataset to avoid deteriorating the prediction.

The Adam optimizer yields slightly higher RMSE compared to the standard quasi-Newton optimizer (for ${W_{el}}^*$, RMSE = 0.120 vs. 0.111 for the full dataset, see Figure 4a) but requires only 2–3 minutes to be executed, to be compared with 1.5 h for the standard quasi-Newton optimizer (see Figure 4b). Indeed, most of the time is tracked back to the final inversion of $\mathbf{K}_{\boldsymbol{\theta}}$ (Eq. (3)) done with the full dataset for accurate predictions. Therefore, stochastic optimization can be useful to quickly estimate to which extent additional evaluations of the physical model can further improve the predictions from the surrogate.

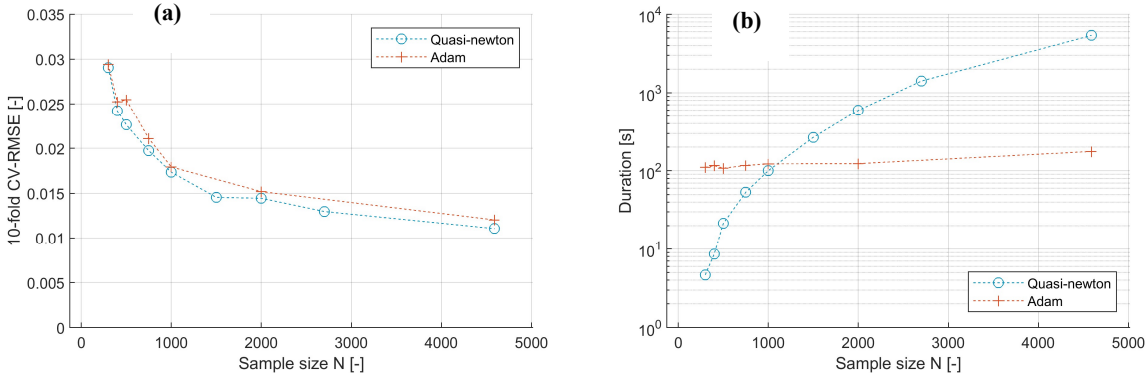


**Figure 4: (a)** 10-fold CV-RMSE of ${W_{el}}^*$ GP and **(b)** Computation time (including training and 10-fold Cross Validation) as a function of the dataset size. The standard numerical implementation (§2.1) and the stochastic optimization with ADAM solver are considered (Kingma and Ba, 2017).

### 3.2 Training objective: Minimization of Root Mean Square Error

It has been suggested to maximize an estimation of the Leave-one-out Cross-Validation RMSE (here denoted LooRMSE) as an alternative to the LML Maximization (Bachoc, 2013). The training objective then reads:

$$\mathrm{LooRMSE}(\boldsymbol{\theta}) = \frac{\mathbf{1}}{N} \boldsymbol{y}^{\mathrm{T}} \mathbf{K}_{\boldsymbol{\theta}}{}^{-1} \left( \mathrm{diag}\left( \mathbf{K}_{\boldsymbol{\theta}}{}^{-1} \right) \right)^{-2} \mathbf{K}_{\boldsymbol{\theta}}{}^{-1} \boldsymbol{y} \tag{10}$$

The minimization of Eq. (10) has been implemented in MATLAB in a similar way as LML maximization (see §2.1). The expression of the gradient $\frac{\partial \mathrm{LooRMSE}}{\partial \boldsymbol{\theta}}$ has been provided to the solver (see the supplementary material of (Bachoc, 2013)), after the correctness of its implementation has been checked against a finite-difference scheme provided by MATLAB built-in function *checkgradients*. It turns out that, in terms of 10-fold CV-RMSE, the LML maximization performs slightly better than the RMSE minimization for small datasets (see Figure 5a), but the difference is not significant for larger

ones. This is qualitatively in line with the results of (Bachoc, 2013). One can observe some slight differences in the inferred $\ell_i$ for small datasets (see Figure 5b).

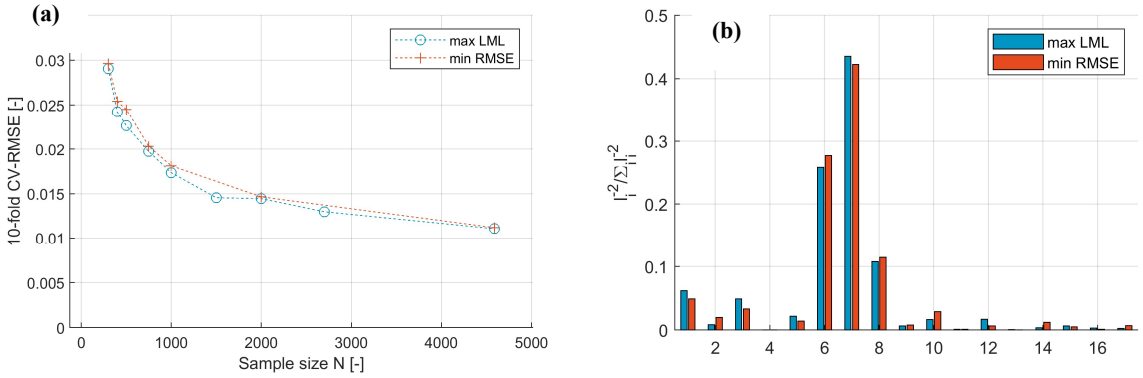


**Figure 5:** Comparison the training through LML maximization (Eq. (7)) and minimization the estimated leave-one-out cross-validation RMSE (Eq. (10)): **(a)** 10-fold CV-RMSE, **(b)** Normalized squared inverse of correlation lengths $\ell_i$ for $N$=750 points.

### 3.3 Linear embedding

To tackle the curse of dimensionality, one appealing approach is to apply a linear transformation (or *linear embedding*) to transform the variable $\boldsymbol{x}$ into $\boldsymbol{x}'$:

$$\boldsymbol{x}' = \mathbf{A} \cdot \boldsymbol{x} \tag{11}$$

Where $\mathbf{A}$ is an orthonormal $d \times D$ matrix (with $d \leq D$). The GP will be trained on $\boldsymbol{x}'$ instead of $\boldsymbol{x}$. Note that if $d = D$, then $\mathbf{A}^{\intercal}$=$\mathbf{A}^{-1}$, i.e. $\mathbf{A}$ is a rotation matrix. To find an appropriate matrix $\mathbf{A}$, we use the Active Subspace methodology (Constantine et al., 2014). A matrix $\mathbf{C}$ is first introduced such that:

$$\mathbf{C} = \frac{1}{N'} \sum_{i=1}^{N'} (\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_i)) . (\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_i))^T \tag{12}$$

In Eq. (12), $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_i) = \left[ \frac{\partial f(x_i^{(1)})}{\partial x^{(1)}} \quad \dots \quad \frac{\partial f(x_i^{(j)})}{\partial x^{(j)}} \quad \dots \quad \frac{\partial f(x_i^{(D)})}{\partial x^{(D)}} \right]$ denotes the gradient of the function of interest $f$ with respect to the variables for data point $\boldsymbol{x}_i$. $\mathbf{C}$ can be evaluated on any point lying in the unit hypercube and is expressed as $\mathbf{C} = \mathbf{W}\,\mathbf{\Lambda}\,\mathbf{W}$, where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \dots, \lambda_D)$ is its eigenmatrix (the eigenvalues $\lambda_j$ have been sorted in descending order). One may select only the most significant eigenvalues $\lambda_j$ $(j = 1, \dots, d)$ to build the GP, leaving apart the smallest values (for instance, find $d$ such that $\sum_{j=1}^{d} \lambda_i < c \sum_{j=1}^{D} \lambda_j$ with $c$ <1 is a constant left to the judgement of the expert). The matrix $\mathbf{A}$ is then merely $\mathbf{A} = \mathbf{W}^T$. Here, to estimate $\mathbf{C}$, $N' = 3000$ points are randomly generated in the unit hypercube and for every point $\boldsymbol{x}_i$ the gradient $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_i)$ is evaluated with a finite difference scheme. The process is repeated 4 times, training the GP on $\boldsymbol{x}'$ in the subsequent iterations.

To illustrate the benefit of Active Subspace for functions with variations along preferential axis, we use the following toy function with $D$=6 and $N$=100 points:

$$f(\boldsymbol{x}) = \sin\left( 5 \sum_{j=1}^{D} x^{(j)} \right) \tag{13}$$

It is clear that a one-dimensional embedding is sufficient (i.e. the active subspace is of dimension $d$ =1), which its associated projection matrix $\mathbf{A} = \pm \left[ \frac{1}{\sqrt{D}} \quad \dots \quad \frac{1}{\sqrt{D}} \right]$ and $\lambda_1 \gg \lambda_j$ $(\forall j > 1)$. Indeed, while the standard GP fails to accurately predict the response (see Figure 6a), the Active Subspace performs much

better (see Figure 6b). As expected, the first eigenvalue is much larger than the second one (here by about six orders of magnitude), while every element in the first line of **A** is approximatively $0.408 \approx \frac{1}{\sqrt{6}}$ (the following lines of **A**, as associated with tiny eigenvalues, are not representative).

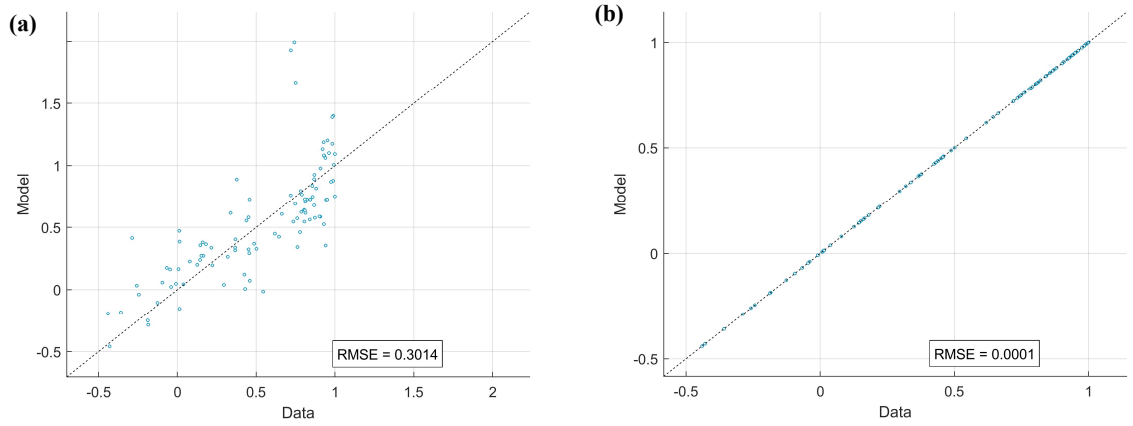


**Figure 6:** Comparison of the 10-fold CV-RMSE for the toy function (Eq. (13)) with **(a)** Standard GP, **(b)** GP with linear embedding.

The application of the AS to the $W_{el}{}^{*}$ dataset does not improve the prediction, see Figure 7a, the RMSE computed with the AS is even slightly higher than with standard GP. However, the Active Subspace provides useful insights into the structure of $W_{el}{}^{*}$.

Figure 7b shows that $d = 8$ eigenvalues $\lambda_i$ capture 99 % of $\sum_{j=1}^{D} \lambda_j$. However, it is necessary to take into account almost all components of **A** to get an accurate prediction, as shows slower the decrease in the RMSE with the number of eigenvalues (in Figure 7b, the RMSE is computed on a test sample equal to 10% of the full dataset, for the GP hyperparameters, including **A**, determined for the full dataset). We have checked that the matrix **A** is almost the same throughout the 10 cross-validated folds: in each fold, each eigenvector is linear with its counterpart obtained considering from the full dataset (the mean value for the angle $\alpha$ between two eigenvectors is typically $\cos\alpha \approx 0.99$). Indeed, the main advantage of AS is its ability to produce insight into the significant associations of parameters. Table 1 shows for instance that the first eigenvector is mostly affected by the available geothermal flow rate per maximum cooling power $\dot{m}_{GTH}/\dot{Q}_{cooling}^{max}$, while the second one depicts the complex interplay between the geothermal flow rate $\dot{m}_{GTH}$, temperature $T_{GTH}$ and the capacity of the absorption chiller. For the absorption chiller to produce a significant amount of cooling that substitutes to the electricity-intensive cooling from the air-source heat pump, a significant amount of hot geothermal flow must be available, and quite a large absorption chiller must be installed.

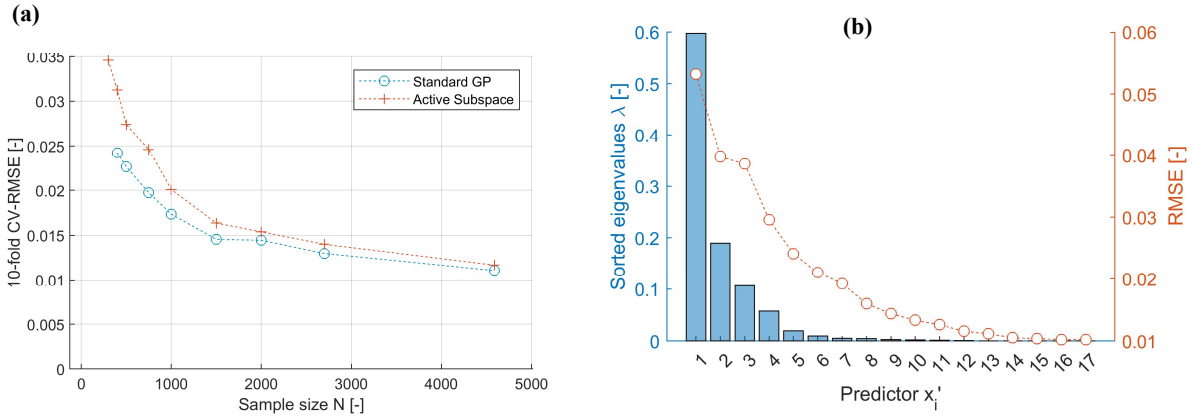


**Figure 7: (a)** Comparison of the 10-fold CV-RMSE for $W_{el}{}^{*}$ with the Standard GP and GP with linear embedding (active subpace), (b) Eigenvalues of **A** and RMSE for as a function of the number of dimensions *d* of the active subspace **A**.

**Table 1:** Four first components of the three most significant eigenvectors for $W_{el}{}^{*}$ ($\dot{m}_{GTH}/\dot{Q}_{cooling}^{max}$: ratio of geothermal flow rate $\dot{m}_{GTH}$ to maximum cooling power required by the building $\dot{Q}_{cooling}^{max}$ [kg.s$^{-1}$.W$^{-1}$], $T_{GTH}$: geothermal temperature [°C], $\dot{Q}_{cooling}^{max}$: maximum cooling power [W], $q_{\ell}{}^{-1}$: Pipe length per estimated delivered energy [m.W$^{-1}$], $\dot{Q}_{sorp}^{ev}{}^{*}$: Normalized nominal cooling power of the absorption chiller [W.W$^{-1}$])

| 1st eigenvector ($j$=1) ($\lambda_1$ = 0.597) | | 2nd eigenvector ($\lambda_2$ = 0.190) | | 3rd eigenvector ($\lambda_3$ = 0.107) | |
|---|---|---|---|---|---|
| **Predictor** | $a_{kj}{}^2$ | **Predictor** | $a_{kj}{}^2$ | **Predictor** | $a_{kj}{}^2$ |
| $\dot{m}_{GTH}/\dot{Q}_{cooling}^{max}$ | 0.5879 | $T_{GTH}$ | 0.3742 | $\dot{Q}_{sorp}^{ev}{}^{*}$ | 0.4019 |
| $T_{GTH}$ | 0.2026 | $\dot{Q}_{sorp}^{ev}{}^{*}$ | 0.3202 | $\dot{Q}_{cooling}^{max}$ | 0.2194 |
| $\dot{Q}_{cooling}^{max}$ | 0.0887 | $\dot{m}_{GTH}/\dot{Q}_{cooling}^{max}$ | 0.2246 | $\dot{m}_{GTH}/\dot{Q}_{cooling}^{max}$ | 0.1837 |
| $q_{\ell}{}^{-1}$ | 0.0729 | $T_{sp\ cooling}$ | 0.0277 | $q_{\ell}{}^{-1}$ | 0.1236 |

## 4 CONCLUSIONS

Several variants of Gaussian Processes have been investigated to assess whether they can improve prediction performance by reducing the 10-fold Cross-Validation RMSE, compared to a standard GP implementation. For the considered dataset made of $N$ = 4589 model evaluations in $D$ = 17 dimensions, it turns out that:

- A multi-start strategy for the GP training (optimization of the hyperparameters) does not improve the prediction, and can even deteriorate the GP quality if the optimizer is stuck in some local minima of the training criteria.
- A stochastic optimization method can dramatically reduce GP training time, from the order of hours to minutes, though the determined hyperparameters are slightly different from the hyperparameters discovered by a standard quasi-Newton optimization and the RMSE is slightly higher.
- Considering the leave-one-out cross validation-based Root Mean Square Error minimization for the GP training instead of the Log Marginal Likelihood maximization yields no significant difference.
- Linear embedding can dramatically improve the prediction for a toy function whose variations are along a preferential axis. Though it does not improve the RMSE for the dataset considered in this study, it gives insight into the significant combinations of variables at stake.

## NOMENCLATURE

**Latin letters**

| | |
|---|---|
| $\mathbf{A}$ | Orthonormal matrix for linear embedding |
| $D$ | Number of predictors (i.e. variables of the GP) |
| $k(.,.)$ | Covariance function (kernel) |
| $\mathbf{K}$ | Covariance matrix |
| $\ell_i$ | Correlation associated with predictor $i$ |
| $N$ | Number of model evaluations available |
| $r$ | Mahalanobis distance between a pair of points |
| $\boldsymbol{x}$ | Parameter of the physical model: variable (aka predictor) of the surrogate |
| $\boldsymbol{y}$ | Output of the physical model |

**Greek letters**

| | |
|---|---|
| $\varepsilon$ | Gaussian noise |
| $\boldsymbol{\varphi}$ | Transformed hyperparameters $\boldsymbol{\varphi} = \log(\boldsymbol{\theta})$ |
| $\boldsymbol{\lambda}$ | Eigenvalue |
| $\boldsymbol{\theta}$ | Hyperparameters (parameter of the Gaussian Process) |

**Subscripts**

| | |
|---|---|
| *0* | initial |

**Superscripts**

| | |
|---|---|
| *ref* | reference |

**Abbreviations**

| | |
|---|---|
| LML | Log Marginal Likelihood |
| RMSE | Root Mean Square Error |
| CV | Cross Validation |
| GP | Gaussian Process |

## REFERENCES

Bachoc, F., 2013. Cross Validation and Maximum Likelihood estimations of hyper-parameters of Gaussian processes with model misspecification. Computational Statistics & Data Analysis 66, 55–69. https://doi.org/10.1016/j.csda.2013.03.016

Binois, M., Wycoff, N., 2022. A Survey on High-dimensional Gaussian Process Modeling with Application to Bayesian Optimization. ACM Trans. Evol. Learn. Optim. 2, 1–26. https://doi.org/10.1145/3545611

Chen, H., Zheng, L., Kontar, R.A., Raskutti, G., 2022. Gaussian Process Parameter Estimation Using Mini-batch Stochastic Gradient Descent: Convergence Guarantees and Empirical Benefits. Journal of Machine Learning Research 23, 1–59.

Constantine, P.G., Dow, E., Wang, Q., 2014. Active subspace methods in theory and practice: applications to kriging surfaces. SIAM J. Sci. Comput. 36, A1500–A1524. https://doi.org/10.1137/130916138

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning, Springer Series in Statistics. Springer New York, New York, NY. https://doi.org/10.1007/978-0-387-84858-7

Kim, J., 2021. Kernels and Derivatives of Kernels.

Kingma, D.P., Ba, J., 2017. Adam: A Method for Stochastic Optimization. https://doi.org/10.48550/arXiv.1412.6980

Maragna, C., Altamirano, A., Tréméac, B., Fabre, F., Rouzic, L., Barcellini, P., 2024. Design and optimization of a geothermal absorption cooling system in a tropical climate. Applied Energy 364, 123102. https://doi.org/10.1016/j.apenergy.2024.123102

McKay, M.D., Beckman, R.J., Conover, W.J., 1979. A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. Technometrics 21, 239–245. https://doi.org/10.2307/1268522

Rasmussen, C.E., Williams, C.K.I., 2006. Gaussian processes for machine learning, Adaptive computation and machine learning. MIT Press, Cambridge, Mass.

Roustant, O., Ginsbourger, D., Deville, Y., 2012. DiceKriging, DiceOptim: Two R Packages for the Analysis of Computer Experiments by Kriging-Based Metamodeling and Optimization. Journal of Statistical Software 51, 1–55. https://doi.org/10.18637/jss.v051.i01

Swiler, L., Gulian, M., Frankel, A., Safta, C., Jakeman, J., 2020. A Survey of Constrained Gaussian Process Regression: Approaches and Implementation Challenges. J Mach Learn Model Comput 1, 119–156. https://doi.org/10.1615/JMachLearnModelComput.2020035155

Westermann, P., Evins, R., 2019. Surrogate modelling for sustainable building design – A review. Energy and Buildings 198, 170–186. https://doi.org/10.1016/j.enbuild.2019.05.057

## ACKNOWLEDGEMENT

The authors would like to thank BRGM for its support under internal research project COMETEUS. This paper would not have been possible without this support.