# Gaussian Process structure for the emulation of deterministic and stochastic solvers: a simulation study

Rosa Arboretti[*] Riccardo Ceccato[†] Luca Pegoraro[‡] Luigi Salmaso[§]

**Abstract**

The increased diffusion of complex numerical solvers to emulate physical processes demands the development of fast and accurate surrogate models. Gaussian Processes (GPs) are the most widely adopted models in this context, as they proved to be sufficiently flexible to effectively mimic the behaviour of complex phenomena and they also provide a quantification of uncertainty of predictions. However, the accuracy of the model depends on both the trend component and covariance structure. In this work we conduct an extensive simulation study that investigates the performance of several GP structures considering the deterministic, homoscedastic and heteroscedastic noise settings. As a result, the findings of this work provide guidelines to practitioners dealing with both deterministic and stochastic solvers.

**Key Words:** Gaussian Process, Simulation study, Kernel, Homoscedastic, Heteroscedastic

## 1. Introduction

Recently, the development of numerical solvers of complex physical phenomena is getting increasingly more attention. The main advantage of using a computer code instead of collecting data on the physical phenomenon is that it is usually cheaper, especially in those cases in which gathering data is expensive. Furthermore, in some situations it can be impossible or impractical to collect the data needed for the purpose of the analysis. However, due to the increased complexity, the computational cost of running those numerical solvers is also rising. For this reason, there is a need to develop fast and accurate surrogate models that can reliably predict the outcome of the complex numerical solver in a fraction of the time.

Gaussian Processes (GPs) are the statistical learning algorithm mainly used for the generation of accurate surrogate models (Gramacy, 2020). Typically, GPs are trained on a set of data which includes realizations of the complex computer code at different input configurations. The selection of the input configurations for the development of the surrogate model is generally guided by a Design of Experiments (DOE) strategy. In this context we refer to the literature about “computer experiments” , in order to differentiate with the classical DOE applications on physical experiments that have different characteristics and requirements (Garud, Karimi, & Kraft, 2017).

GP models rely mainly on two assumptions: (i) the data generating mechanism produces samples coming from a multivariate Gaussian distribution, (ii) the samples are associated by a covariance function $k(\cdot,\cdot)$. Based on this, GP algorithms are able to estimate both the mean and variance for a new input vector, meaning that a quantification of the uncertainty of prediction is always included with the point estimation. This, together with the relatively high flexibility that ensures the emulation of complex phenomena, is the main advantage of this methodology with respect to other algorithms coming from the machine learning literature.

[*] Department of Civil, Environmental and Architectural Engineering, University of Padova, Padua, Italy

[†] Department of Management and Engineering, University of Padova, Vicenza, Italy

[‡] Department of Management and Engineering, University of Padova, Vicenza, Italy

[§] Department of Management and Engineering, University of Padova, Vicenza, Italy

In this article we conduct a simulation study to investigate the role of different covariance functions and their impact on the accuracy of the analysis. Furthermore, we also study whether the inclusion of a trend component in the GP model improves the final predictions. In order to cope with the arising number of stochastic solvers, common in the social, biological and management sciences, we also consider several noise settings, both homoscedastic and heteroscedastic, and multiple test functions that are computer codes emulating physical processes. The idea is to investigate several circumstances that are commonly encountered by practitioners, and to provide guidelines that can be followed for the development of accurate and reliable surrogate models.

## 2. Methodology

In this section we provide some technical details about the experimental designs and some methodological insights on the GP models. Furthermore, we briefly present a ranking methodology which has been selected for obtaining the final rank of the GP models, based on the choice of kernel functions and trend component.

### 2.1 Experimental designs

In the context of computer experiments, the key aim of DOE is to generate the sample points to fill the experimental domain (Garud et al., 2017). This translates in the generation of experimental designs that are space-filling, meaning that they tend to spread out design points as much as possible. In this article we consider two of such designs: the Random Latin Hypercube Design (LHD_rand) (McKay, Beckman, & Conover, 1979; Stein, 1987), that is one of the oldest and most adopted methodologies, and the Maximum Projection Design (MAXPRO) (Joseph, Gul, & Ba, 2015) that is one recent advancement of the LHD_rand, having several improvements (Joseph, 2016).

In this paper, for both experimental designs we generated 52 runs, as this is the size required by a Central Composite Design with the same number of dimensions considered in this study. Furthermore, this is in line with the empirical relation $N \sim 10d$ (Loeppky, Sacks, & Welch, 2009).

### 2.2 Gaussian Process models

Consider a space-filling design $\mathbf{D}$ with $d$ features and with $n$ runs, $\mathbf{D} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, where $\mathbf{x} = (x_1, ..., x_d)$ is one of the $n$ inputs and $\mathbf{y} = (y(\mathbf{x}_1), ..., y(\mathbf{x}_n))^T$ is a vector containing the outputs of the computer code. The data generating mechanism is of the GP type if $y(\mathbf{x})$ is a realization of:

$$Y(\mathbf{x}) = \mu(\mathbf{x}) + Z(\mathbf{x}) \tag{1}$$

where $\mu(\mathbf{x})$ is the trend component and $Z(\mathbf{x})$ is multivariate normal with mean 0 and covariance function $k$.

In this article, we consider two structures for the trend component $\mu(\mathbf{x})$:

- Constant, $\mu(\mathbf{x}) = \beta_0$.
- Quadratic with interactions:

$$\mu(\mathbf{x}) = \beta_0 + \sum_{i=1}^{d} \beta_i x_i + \sum_{i=1}^{d} \beta_{ii} x_i^2 + \sum_{i<j}\sum \beta_{ij} x_i x_j \tag{2}$$

The relevant coefficients are selected via backward stepwise elimination with objective the minimization of the 5-fold Cross Validation (CV) error (RMSE). In the remaining of the paper, we will refer to the "trend" case when this option is selected.

Moreover, $Z(\mathbf{x})$ is in the form:

$$Z(\mathbf{x}) \sim \mathcal{N}(0, \, \sigma^2 k(\cdot, \cdot)) \tag{3}$$

where $\sigma^2$ is the process variance (Roustant, Ginsbourger, & Deville, 2012). Furthermore, consider $x$ and $x'$ two entries of the input vectors $\mathbf{x}$ and $\mathbf{x}'$; the 1-dimensional kernel function associating $x$ and $x'$ is assumed to depend solely upon the difference: $h = |x - x'|$. This can be extended to the multi-dimensional case by taking $K(\mathbf{x}, \mathbf{x}') = \prod\limits_{i=1}^{d} k(h_i)$.

The kernel functions considered in this article are:

- Exponential: $k(h) = \texttt{exp}(-\frac{h}{\theta})$
- Gaussian: $k(h) = \texttt{exp}(-\frac{h^2}{2\theta^2})$
- Matérn $3/2$: $k(h) = (1 + \frac{\sqrt{3}h}{\theta})\texttt{exp}(-\frac{\sqrt{3}h}{\theta})$
- Matérn $5/2$: $k(h) = (1 + \frac{\sqrt{5}h}{\theta} + \frac{5h^2}{3\theta^2})\texttt{exp}(-\frac{\sqrt{5}h}{\theta})$
- Power-Exponential: $k(h) = \texttt{exp}(-(\frac{h}{\theta})^q)$, with $0 < q \leq 2$

$\theta$ is a sensitivity parameter called "lengthscale" that defines the rate of decay of the correlation among two data configurations (Gramacy, 2020; Rasmussen & Williams, 2006). The easiest implementation supposes that uniform decay in correlation exists in every direction, thus assumes $\theta$ to be a scalar. However, this assumption of radial symmetry rarely holds, and a more general representation of the lengthscale parameter is as a vector ( $\boldsymbol{\theta} = (\theta_1, ..., \theta_d)$), thus allowing the intensity of correlation to change independently on each different dimension. In this work we consider this second formulation.

Another important parameter is the nugget $t$, that is added to the diagonal elements of the covariance matrix and ensures stability of the computation, $t = 10^{-8}\texttt{var}(\mathbf{y})$.

### 2.3 Nonparametric ranking procedure

The procedure adopted for the generation of the final rank of the GP models is described in (Arboretti, Bonnini, Corain, & Salmaso, 2014). This procedure uses the general principles of permutation tests (Pesarin & Salmaso, 2010), and does not require stringent assumptions on data distribution or size. Considering $Q_i$ and $Q_j$ with $i \neq j$ two groups of data to be compared (the different structures for the GP models), the first step consists in the execution of permutation tests in order to compare the RMSE achieved by each group on an independent test set, as such obtaining $Q(Q-1)$ p-values that are the elements of a matrix $\mathbf{G}_{Q \times Q}$ that has 1 in each cell of the diagonal vector.

The subsequent steps are:

1. Create the matrix $\mathbf{S}$ where $S_{ij} = 0$ if $G_{ij} > \alpha/2$ and $S_{ij} = 1$ if $G_{ij} \leq \alpha/2$ , with $\alpha = 0.05$.
2. Generate the vector $\mathbf{r}^D$ whose elements are $r_j^D = 1 + \sum_{i=1}^{C} S_{ij}$, $j = 1, ..., Q$.
3. Generate the vector $\mathbf{r}^U$ whose elements are $r_i^U = 1 + \{\#(Q - \sum_{j=1}^{Q} S_{ij}) > (Q - \sum_{j=1}^{Q} S_{i'j}), i' = 1, ..., Q, i' \neq i\}$, $i = 1, ..., Q$.

4. Generate the vector $\mathbf{r}$ whose elements are $r_i = 1 + \{\#(r_i^D + r_i^U)/2 > (r_j^D + r_j^U)/2, j = 1, ..., Q, i \neq j\}, i = 1, ..., Q$.

The group that achieves the overall smallest RMSE occupies the first position in the final ranking. Draws are also possible, if two groups are not significantly different.

## 3. Simulation study

In this section we describe more in detail the simulation setting (Figure 1) in terms of the test functions and noise structures considered.

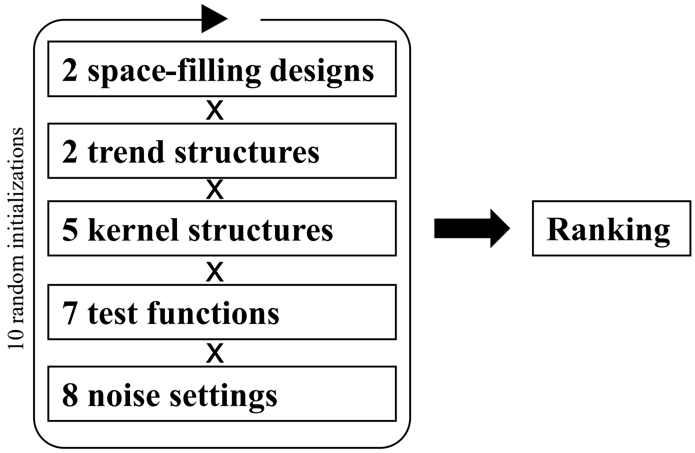


**Figure 1**: Framework of the simulation study.

### 3.1 Test functions

In order to test the performance of the different kernels and trend functions in disparate settings, 7 different test functions have been selected, namely “Borehole” , “OTL circuit”, “Piston”, “Piston Mod”, “Robot arm”, “Rosenbrock” and “Wing weight” . The functions have been retrieved from (Surjanovic & Bingham, 2021), and have been restricted to 6 active dimensions. “Piston Mod” is a modified version of the “Piston” function, with increased non-linearity effects. Furthermore, the dependent variables have been standardized and independent variables have been normalized.

### 3.2 Noise settings

We investigate both the homoscedastic and heteroscedastic noises, with different intensities:

1. Homoscedastic noise: $\epsilon \sim \mathcal{N}(0, \sigma_{hom}^2)$, with $\sigma_{hom}$ in the range $[0, 0.5\sigma_y]$ and $\sigma_y$ estimated from a large LHD design for each test function (Table 1).

2. Heteroscedastic noise: $\epsilon \sim \mathcal{N}(0, \sigma_{het}^2)$, with $\sigma_{het}$ that increases linearly with the response (Table 2).

**Table 1**: Summary of the homoscedastic noise structures.

| Noise | Description |
|---|---|
| $\sigma_{hom} = 0\sigma_y$ | 0% noise |
| $\sigma_{hom} = 0.05\sigma_y$ | 5% noise |
| $\sigma_{hom} = 0.125\sigma_y$ | 12.5% noise |
| $\sigma_{hom} = 0.2\sigma_y$ | 20% noise |
| $\sigma_{hom} = 0.5\sigma_y$ | 50% noise |

**Table 2**: Summary of the heteroscedastic noise structures.

| Noise | Description |
|---|---|
| $\sigma_{het,min} = 0.05\sigma_y$, $\sigma_{het,max} = 0.5\sigma_y$ | 5% noise at $\min y$ and 50% noise at $\max y$ (low5_high50) |
| $\sigma_{het,min} = 0.05\sigma_y$, $\sigma_{het,max} = 1\sigma_y$ | 5% noise at $\min y$ and 100% noise at $\max y$ (low5_high100) |
| $\sigma_{het,min} = 0.05\sigma_y$, $\sigma_{het,max} = 5\sigma_y$ | 5% noise at $\min y$ and 500% noise at $\max y$ (low5_high500) |

### 3.3 Results and Discussion

In this section we show and discuss the results of the application of the nonparametric ranking both for the homoscedastic (Figure 2) and heteroscedastic (Figure 3) noise settings, considering each one of the test functions. Furthermore, for providing a better synthetic visualization of the results, another application of the ranking procedure is performed, in order to obtain a final rank of the GP models with respect to each specific noise setting (Tables 3 and 4).

The results in Figures 2 and 3 provide two main indications: (i) the magnitude of noise plays a crucial role in the determination of the ranks and (ii) the performance of different kernels and trends depends on the specific test function considered. Interestingly, one of the kernels that is most (negatively) influenced by the level of noise is the Gaussian, that is very widely employed by practitioners (Gramacy, 2020). At the same time, the Exponential kernel improves its performance in presence of larger levels of noise. However, due to the large number of combinations, it is quite difficult to come to definitive conclusions only by analyzing Figures 2 and 3, thus the ranks here discussed are ranked again, in order to produce some general synthetic results that do not directly depend upon the specific test function. These are provided in Tables 3 and 4 for the homoscedastic and heteroscedastic noises respectively.

Table 3 shows that the Matérn $5/2$ kernel is the best option for the deterministic functions, together with its version including the trend component and the Gaussian kernel including the trend component. This result justifies the adoption of the Matérn $5/2$ kernel as the default in many software packages, that usually refer to the case of absence of noise (Roustant et al., 2012). As the impact of noise increases, the situation becomes less clear, in the sense that the same rank position is assigned to many groups and no practical difference exists if the trend component is included and/or a different kernel is chosen, a part from the Exponential and Gaussian kernels, that tend to rank last. When the noise increases over 20%, significant differences are detected and the Exponential kernel with the trend component ranks as first. In general, for the homoscedastic case the inclusion of a trend component appears to increase, albeit marginally, the performance of the algorithms as when a trend component is added the rank is usually the same or better than the case with only a constant trend. Two exceptions are the Matérn $5/2$ and Power-Exponential kernels, that perform worst if the trend is added.

Table 4 shows the final results for the heteroscedastic case. For all three noise structures relevant differences exist depending on the kernel and/or the inclusion of a trend component. In general, findings similar to the previous case are observed: the trend component does not play a major role, in fact for the Matérn $5/2$ and Power-Exponential kernels it reduces the accuracy of predictions. As in the previous case, the inclusion of the trend seems to be justified only for the Exponential kernel. On the other hand, the choice of the kernel appears to be more critical than in the previous case, as the Power-Exponential and Exponential + trend kernels are by far the best performers. Interestingly, for both the homoscedastic and heteroscedastic cases the Gaussian kernel is one of the worst performers.

All things considered, in the case of a stochastic solver the final guideline for practitioners is to employ a Power-Exponential kernel with constant trend, as this option seems to be preferable in the presence of noise. On the other hand, if deterministic computer codes are took into account, the Matérn $5/2$ kernel should be the preferred choice. No strong evidences in favour of the inclusion of a complex trend component are found, except for the Exponential kernel.

This work confirms the findings of previous studies (Chen, Loeppky, Sacks, Welch, et al., 2016), expanding the scope of the analysis also to noisy situations (homoscedastic and heteroscedastic).

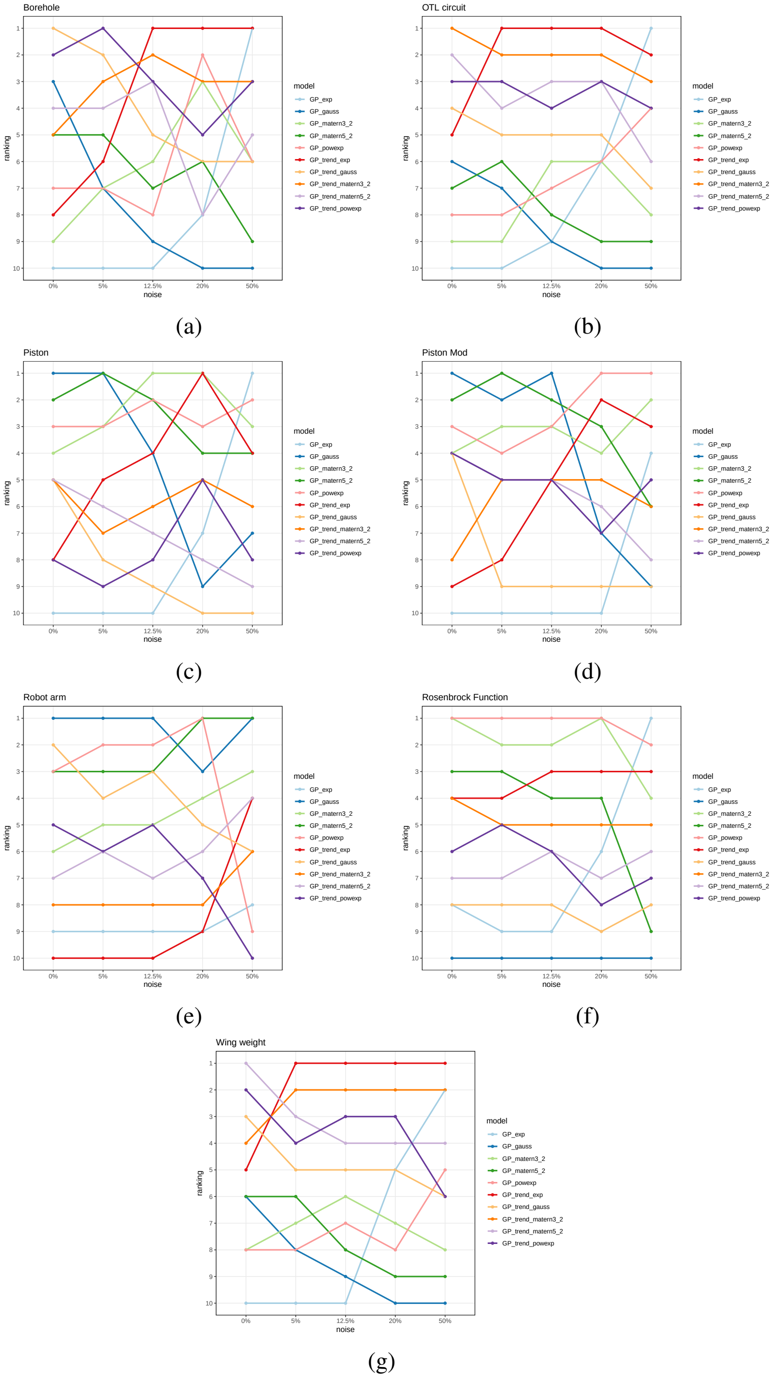
## 4. Conclusions

In this paper we conducted a simulation study that tests the performance of different kernel functions on the accuracy of a GP model. Furthermore, we also investigated the role of the trend component, and we considered several noise settings, both homoscedastic and heteroscedastic. This implies that the results of the study are applicable for both deterministic and stochastic computer solvers. Additionally, since 7 test functions with various degrees of non-linearity are used, we believe that the simulation setting covers many practical situations encountered by practitioners.

The main findings can be summarised as follows: (i) there is no strong indication that the inclusion of a trend component generally improves the predictions, (ii) the choice of an appropriate kernel function is crucial, (iii) the Matérn $5/2$ and Power-Exponential kernels are the most promising in deterministic and noisy situations respectively, (iv) the Gaussian kernel, that is one of the most widely employed, turns out to be one of the worst performers in the simulation setting considered.

**Table 3**: Final rank of the GP structures for the homoscedastic noise cases. The ranks are provided for each noise level, while the bars show the rowwise sum, thus giving an indication of the overall performance of each model structure (the smaller the better).

| **Model** | **0%** | **5%** | **12.5%** | **20%** | **50%** | |
|---|---|---|---|---|---|---|
| GP_exp | 10 | 10 | 10 | 8 | 2 | 40 |
| GP_gauss | 4 | 1 | 9 | 8 | 9 | 31 |
| GP_matern3_2 | 4 | 1 | 1 | 2 | 5 | 13 |
| GP_matern5_2 | 1 | 1 | 1 | 5 | 6 | 14 |
| GP_powexp | 4 | 1 | 1 | 2 | 3 | 11 |
| GP_trend_exp | 9 | 1 | 1 | 1 | 1 | 13 |
| GP_trend_gauss | 1 | 1 | 1 | 8 | 10 | 21 |
| GP_trend_matern3_2 | 4 | 1 | 1 | 2 | 4 | 12 |
| GP_trend_matern5_2 | 1 | 1 | 1 | 7 | 6 | 16 |
| GP_trend_powexp | 4 | 1 | 1 | 5 | 6 | 17 |



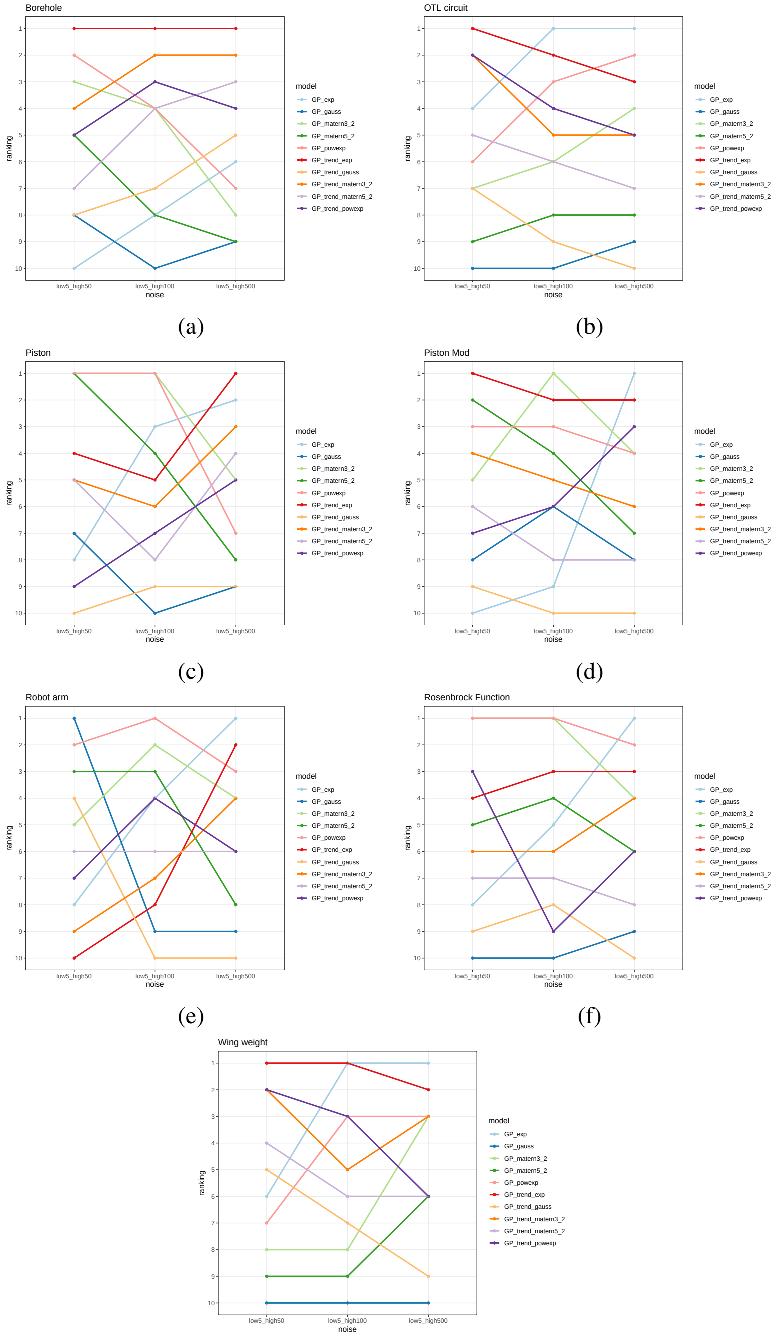
(a) (b)

(c) (d)

(e) (f)

(g)

**Figure 2**: Homoscedastic noise: rank of the GP models for each test function.



(a) (b)

(c) (d)

(e) (f)

(g)

**Figure 3**: Heteroscedastic noise: rank of the GP models for each test function.

**Table 4**: Final rank of the GP structures for the heteroscedastic noise cases. The ranks are provided for each noise level, while the bars show the rowwise sum, thus giving an indication of the overall performance of each model structure (the smaller the better).

| **Model** | **low5_high50** | **low5_high100** | **low5_high500** | |
|---|---|---|---|---|
| GP_exp | 10 | 4 | 2 | 16 |
| GP_gauss | 8 | 9 | 10 | 27 |
| GP_matern3_2 | 4 | 2 | 5 | 11 |
| GP_matern5_2 | 5 | 5 | 8 | 18 |
| GP_powexp | 1 | 1 | 3 | 5 |
| GP_trend_exp | 2 | 2 | 1 | 5 |
| GP_trend_gauss | 9 | 9 | 9 | 27 |
| GP_trend_matern3_2 | 3 | 5 | 3 | 11 |
| GP_trend_matern5_2 | 7 | 8 | 7 | 22 |
| GP_trend_powexp | 5 | 5 | 5 | 15 |

## References

Arboretti, R., Bonnini, S., Corain, L., & Salmaso, L. (2014). A permutation approach for ranking of multivariate populations. *Journal of Multivariate Analysis*, *132*, 39–57.

Chen, H., Loeppky, J. L., Sacks, J., Welch, W. J., et al. (2016). Analysis methods for computer experiments: How to assess and what counts? *Statistical science*, *31*(1), 40–60.

Garud, S. S., Karimi, I. A., & Kraft, M. (2017). Design of computer experiments: A review. *Computers & Chemical Engineering*, *106*, 71-95. (ESCAPE-26) doi: https://doi.org/10.1016/j.compchemeng.2017.05.010

Gramacy, R. B. (2020). *Surrogates: Gaussian process modeling, design and optimization for the applied sciences*. Boca Raton, Florida: Chapman Hall/CRC. (http://bobby.gramacy.com/surrogates/)

Joseph, V. R. (2016). Space-filling designs for computer experiments: A review. *Quality Engineering*, *28*(1), 28-35. doi: 10.1080/08982112.2015.1100447

Joseph, V. R., Gul, E., & Ba, S. (2015). Maximum projection designs for computer experiments. *Biometrika*, *102*(2), 371–380.

Loeppky, J. L., Sacks, J., & Welch, W. J. (2009). Choosing the sample size of a computer experiment: A practical guide. *Technometrics*, *51*(4), 366–376.

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*(2), 239–245. Retrieved from http://www.jstor.org/stable/1268522

Pesarin, F., & Salmaso, L. (2010). *Permutation tests for complex data: theory, applications and software*. John Wiley & Sons.

Rasmussen, C., & Williams, C. (2006). Gaussian processes for machine learning.,(mit press: Cambridge, ma).

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), 1–55. Retrieved from http://www.jstatsoft.org/v51/i01/

Stein, M. (1987). Large sample properties of simulations using latin hypercube sampling. *Technometrics*, *29*(2), 143–151.

Surjanovic, S., & Bingham, D. (2021). *Virtual library of simulation experiments: Test functions and datasets.* Retrieved 2021-09-01, from http://www.sfu.ca/ ssurjano