Available online at www.eccomasproceedia.org
Eccomas Proceedia UNCECOMP (2023) 222-236

ECCOMAS
Proceedia

UNCECOMP 2023
5th ECCOMAS Thematic Conference on
Uncertainty Quantification in Computational Science and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou (eds.)
Athens, Greece, 12-14 June 2023

TOWARDS RELIABLE AUTOMATED DRIVING SYSTEMS: SURROGATE MODELING USING SPARSE GRIDS AND GAUSSIAN PROCESSES

Naya Baslan[1,2], Julian Schmidt[1], Alexander F. Kerschl[1], and Dirk Pflüger[2]

[1]Robert Bosch GmbH
Cross-Domain Computing Solutions
Heßbrühlstraße 21, 70565 Stuttgart-Vaihingen, Germany
{naya.baslan, Julian.Schmidt2, AlexanderFerdinand.Kerschl}@bosch.com

[2] University of Stuttgart
Scientific Computing, Institute for Parallel and Distributed Systems
Universitätsstr. 38, 70569 Stuttgart, Germany
{naya.baslan, dirk.pflueger}@ipvs.uni-stuttgart.de

Abstract. *Surrogate modeling is a field of increasing importance in the field of autonomous driving simulations. Surrogates are used to provide a time efficient model that would save long hours of simulation runs and save computational power. The field of Search Based Testing, which is the ultimate field of application of the model presented in this paper, requires varying parameters to cover the whole parameter space. Available methods tend to be very time consuming and unfortunately fail to cover the whole input space.*

The surrogate model presented in this paper is inspired by work of Schöbi et al. [1], which were the first to develop methods that improve the mean prior function of a Gaussian process to obtain an improved posterior. This paper develops a similar model, but makes use of the advantages provided by the research done in the field of Sparse Grids to improve the Gaussian Process posterior.

The obtained Gaussian posterior is tested using well known mathematical functions which are used to quantify the accuracy of the model and to provide a basis for comparing it to other methods available in the field. The application fields of the resulting model range from optimizing a possibly constrained objective function for find global optima, to performing reliability analysis over a defined parameter space.

Keywords: Surrogate Models, Sparse Grids, Gaussian Processes, Uncertainty Quantification.

ISSN:2623-3339
Peer-review under responsibility of the organizing committee of UNCECOMP 2023.
doi: 10.7712/120223.10334.20319

1 Introduction

Autonomous Driving Software is yet another field with high safety requirements. Before a self-driving car can be allowed to freely drive in the streets, it should prove to be—even in its worse case scenarios—better than a human driver. Meeting these standards would require running complex systems with expensive simulations and long hours of testing. Performing accurate tests includes trying out many driving scenarios with different input parameters. These simulation runs, until today, struggle to explore the whole parameter space and thus raise the need for developing accurate surrogate models that are inexpensive to run under different scenarios.

Great efforts have been made in developing surrogate modeling techniques by combining several methods in models that highlight the best of each method working separately. Current state-of-the-art work is presented by Schöbi et al. in [1] in and [2], where the proposed method is to combine Polynomial Chaos Expansion (PCE) techniques with Gaussian Processes in a method called PC-Kriging. This method splits the meta-modeling technique into two parts: the PCE captures the global trend of the function, while the Gaussian Process focuses on the local features.

This paper presents an alternative approach which uses Sparse Grids instead of PCEs. The new method allows for adaptivity already in the global part of the model through the use of Ritter-Novak Refinement. We call this method ”SG-Kriging”, with reference to the original ”PC-Kriging” method that was developed by Schöbi et al. in [1].

Sparse Grids [3, 4] are a hierarchical discretization technique based on a regular grid-based discretization that covers the input space. In its most basic form, each discretization level adds extra grid points half-way in between the already existing points from the previous level to provide finer computations. Sparse Grids were further studied in [5] in their spatially adaptive variants to suit high-dimensional problems and in [6] to have hierarchical B-Splines as ansatz functions, instead of the simpler piece-wise polynomial standard approach.

The SG-Kriging method explained in this paper performs a function interpolation using B-Spline-based Sparse Grids, and then uses the result as a mean prior for the Gaussian Process. The developed model aims to be: flexible—so that it can be applied to a variety of functions and testing scenarios, adaptive—so that the choice of the SG points can meet a preset requirement of exploration vs exploitation and is capable of being extended to higher-dimensional problems. The scope of this paper is however limited to only testing on a two-dimensional function, and higher dimensions will be tackled in later work.

This paper is divided into the following sections: Section 2 describes the main motivation of the paper and summarizes the current state-of-art in the field of surrogate modeling. It also explains the potential benefits of using Sparse Grids as priors for Gaussian Processes. Section 3 describes the background theory by separately explaining two commonly used meta-modeling techniques, namely Sparse Grids and Gaussian Processes. Section 4 combines these two meta-modeling techniques in a method called SG-Kriging. Moreover, Section 5 describes the experimental settings that are used for testing, while Section 6 presents first results of testing the developed method on well-known mathematical functions and provides comparisons to each of the individual methods. Finally, Section 7 summarizes the paper and suggests future work to improve the resulting SG-Kriging model.

2 Current State of the Art and Motivation

The first contribution to developing a Gaussian Process with advanced prior knowledge obtained from an external meta-modeling technique came by Schöbi et al. in [1]. In that research, Polynomial Chaos Expansions (PCEs) are used to replace the original trend of the Gaussian Process. In their core, PCEs are used to represent a random variable using orthonormal polynomials, similarly to how a Fourier Series expansion represents a function as a sum of trigonometric functions. Results by Schöbi et al. show how adding sparsity can ease the curse of dimensionality by optimally selecting the number of these polynomials and eliminating unnecessary ones. Furthermore, their definition of the least squares minimization problem shows how to optimally select the expansion coefficients of these polynomials. Their method of combining PCEs with Gaussian Processes produces results that outperform each of the methods implemented separately.

The main motivation of of this work comes from the increasing interest and research developments in the field of Sparse Grids over the past years. This paper highlights the advantages of Sparse Grids and argues that using them to add to the prior knowledge of a Gaussian Process can produce an improved posterior. The main advantages that show promising results:

- The availability of SG^{++} [7], an open-source Sparse Grids toolbox that provides an efficient way of using Sparse Grids in various flavors ranging from the core basics to more advanced techniques.
- Incorporating adaptivity in Sparse Grids as discussed in [8] and thus driving the refinement where really required. This saves computational time in regions that are not of interest.
- The development of B-Spline-variants of Sparse Grids and resolving the previously existing issues around boundary treatment by introducing so-called hierarchical not-a-knot (nak) B-Splines as presented in [6].
- The extension of Modified nak B-Splines to provide a trade-off between spending (too many) grid points on the boundaries and eliminating boundary points. This was achieved in [9] based on a similar approach for piece-wise polynomial bases in [5] to provide an improved estimate of boundary values without the need to discretize those boundaries.

Based on these contributions, this paper presents the use of Sparse Grids to obtain the prior knowledge of a Gaussian Process. Furthermore, in their variant used here, Sparse Grids incorporate adaptivity early on in the model and thus provide a finer grid with increased prior knowledge in the regions of interest of the original function.

3 Theoretical Background

This section explains the theory behind the methods that are used in this paper, namely Sparse Grids and Gaussian Processes. Short summaries are provided for the aim of explaining how they will be combined in the next sections. Detailed information is out of the scope of the paper and can be obtained from the linked references.

3.1 Sparse Grids

Sparse Grids, as introduced by Zenger [3] for the solution of PDEs, are a discretization technique that is suited for high-dimensional problems. They are based on a hierarchical basis in

1D, which is then extended to the d-dimensional setting via a tensor product construction. Examples of these basis functions include piecewise-linear functions and basis splines (B-Splines). Piece-wise linear (or in general polynomial) functions have the disadvantage that they are non-differentiable at the grid points. This becomes problematic in applications which require the use of gradient-based algorithms. On the other hand, B-Splines are polynomials of order p and are at least $p - 1$ times continuously differentiable.

In d dimensions, the hierarchical basis functions form a sequence of nested sub-spaces, which can then be truncated; see [4, 5, 10] for more details. For sufficiently smooth functions, an optimal truncation with respect to the L^2 and maximum norms leads to so-called regular Sparse Grids, and reduces the number of grid points by orders of magnitude with only minor impact on the approximation accuracy. This mitigates the curse of dimensionality to some extent, the exponential dependency of the number of discretization points on the dimensionality. Note that an alternative approach was first introduced by Smolyak in [11] for numerical quadrature and is know as the Smolyak Rule or the Combination Technique.

The scope of this paper is to use Sparse Grids to build a surrogate model that approximates a complicated high-dimensional function by a simpler, more efficient approximation. The original function could be a black box function or a computationally expensive simulation run. In either case, unnecessary evaluations should be avoided because they increase computation time without having a significant improvement on the results. In interpolatory settings, the value of the surrogate model matches that of the original function at all grid points. In-between grid points, values are interpolated depending on the selected basis.

3.2 Generating the Regular Sparse Grid

Figure 1 shows a simple example of how a regular Sparse Grid is generated for level 2. The procedure follows the hierarchical splitting technique. In each subspace shown in Figure 1, a grid point is spent for the Cartesian product of the respective one-dimensional ones (centers of the basis functions). For sufficiently smooth functions, an optimal selection of subspaces leads to a truncation based on the level sum. Combining the grid points of all the four sub-spaces in this example results in a full-grid, while using only those of the upper triangle (shown in darker grey in the figure) result in a regular Sparse Grid of level 2. Here, this results in a total of 5 Sparse Grid points rather than the $3 \times 3 = 9$ for the corresponding full grid.

Figure 2 shows the resulting Sparse Grids for levels 1 to 5 (left to right). The top row shows grids when boundary points are omitted, while the bottom row includes them. Including boundary points challenges the efficiency of the model. A trade-off is needed between completely loosing knowledge about the boundary values and including all points. This motivated work in [5] to modify the hierarchical basis functions to extrapolate at the boundaries so as to include estimated boundary information without spending any grid points at the boundary. The idea is that adaptive refinement can then recover boundary information only where it is really necessary.

3.2.1 From the Piece-Wise Linear Basis to the B-Spline Basis

As already mentioned, piece-wise linear basis functions are problematic as they are non-smooth (non-differentiable) at the grid points. Work by Valentin in [12] introduced a hierarchy of basis splines (B-Splines) as alternative basis functions for Sparse Grids, where piece-wise polynomials are defined between the grid points and are used to interpolate the original function. These piece-wise polynomials are connected to each other by knots, which are connecting

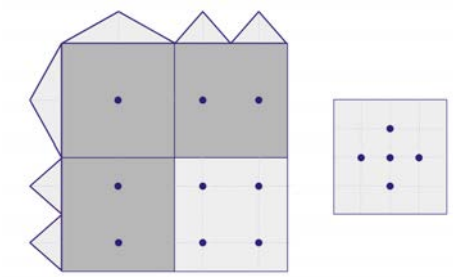

Figure 1: Hierarchical Splitting to generate a regular Sparse Grid of level 2 for a two-dimensional function. The grid on the right shows the 2d regular Sparse Grid.

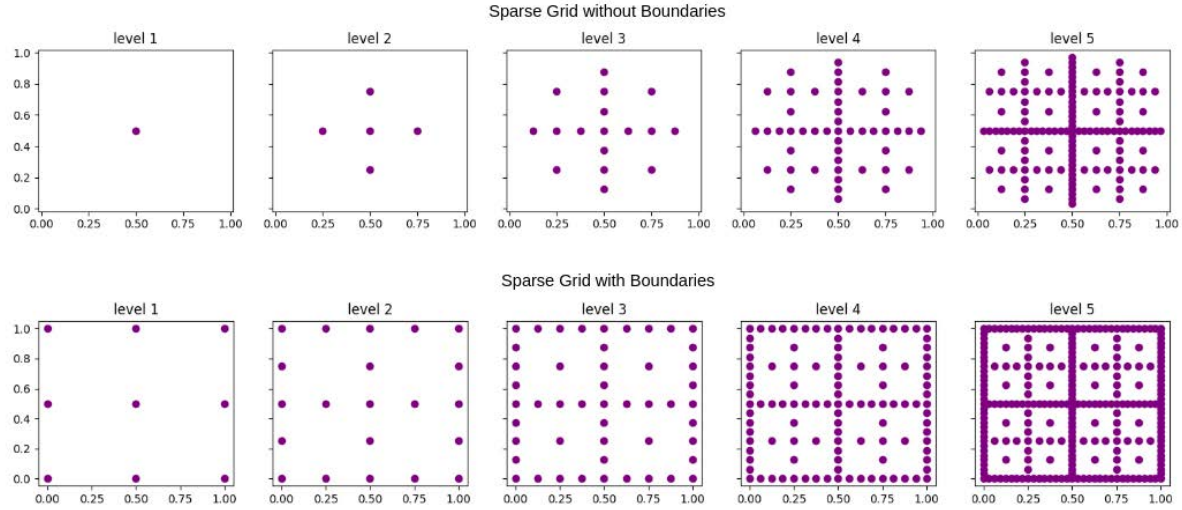

Figure 2: Two-dimensional regular Sparse Grid for levels 1, 2, 3, 4, and 5 (from left to right). The top row shows the results when omitting boundary points, while the bottom row shows the results including them. It can be noted from the bottom row, that at any level, the number of boundary points exceeds that of non-boundary points. This increases the computational complexity of the underlying surrogate model.

points between polynomials at which smoothness is ensured. Moreover, [9] discusses that including boundary conditions can be dimensionally problematic for B-Splines in Sparse Grids and therefore presents a solution using not-a-knot boundary conditions. Not-a-knot sequences are obtained by eliminating the first and last interior knots from the full grid. Furthermore, the same concept of modifying the basis to include boundary information without spending any grid points on the boundary is extended to apply to B-Splines.

3.2.2 Spatial Adaptivity in Sparse Grids

Another important contribution by [12] is the introduction of different approaches to spatial adaptivity. Spatially adaptive Sparse Grids at different levels are shown in Figure 3 for the Ritter-Novak refinement criterion that is targeted to optimzation problems. Unlike the regular grids shown in Figure 2, where the underlying grid shape only depends on the grid level, spatially adaptive Sparse Grids are sensitive to local features of the function. The grid is no longer symmetric and finer discretizations can be added in regions of interest to provide more accurate estimations.

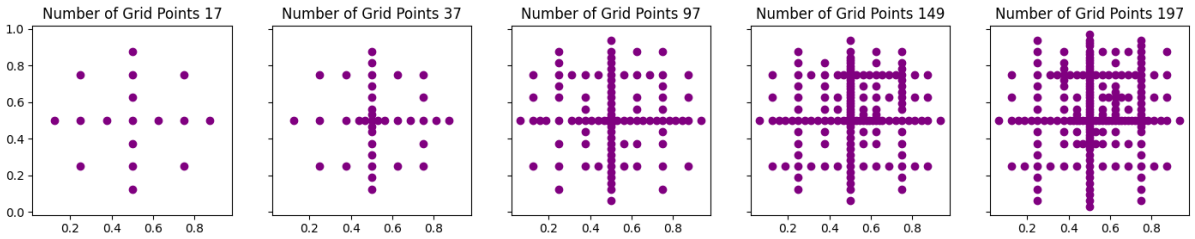

Figure 3: Intermediate snapshots of spatially adaptive refinement of a two-dimensional Sparse Grid without boundary discretization in an optimization scenario based on a Ritter-Novak refinement criterion.

3.3 Gaussian Processes

In their book about Gaussian Processes [13], Rasmussen and Williams define a Gaussian Process as a generalization of a Gaussian distribution, applied over functions. The function $f(x)$,

$$f(x) \sim \mathcal{N}(m(x),\, k(x, x'))\,, \tag{1}$$

is said to follow a Gaussian Process with a mean function $m(x)$ and a covariance function $k(x, x')$, where x and x' are two points in the input space. Covariance functions are defined using symmetric positive functions called kernels, thus the name covariance kernels. Using Gaussian Processes to create a surrogate model simply starts with prior knowledge about the original functions and then incorporates evaluation points to compute the posterior. This posterior function then becomes an efficient approximation of the original function and is thus given the name surrogate model.

The posterior of the Gaussian Process only requires second order moments of the prior, namely the mean and the covariance. For the generally used case where the mean prior is zero, defining the covariance kernel of the process is sufficient to obtain the posterior. The posterior mean and covariance are calculated as

$$m_{\text{surr}}(x) = m(x) + k_{xX}(k_{XX} + \sigma^2 I_n)^{-1}(f_X - m_X)\,, \tag{2}$$

$$k_{\text{surr}}(x, x') = k(x, x') - k_{xX}^T(k_{XX} + \sigma^2 I_n)^{-1}k_{xX},\,, \tag{3}$$

where

$x \in \mathbb{R}$ and $x' \in \mathbb{R}$ are two points in the input space,

$m(x) \in \mathbb{R}$ and $m_{\text{surr}}(x) \in \mathbb{R}$ are the mean prior and posterior functions,

$k(x, x') \in \mathbb{R}$ and $k_{\text{surr}}(x, x') \in \mathbb{R}$ are the prior and posterior covariance kernels,

$X \in \mathbb{R}^{d\times N}$ are the evaluation points and $f_X \in \mathbb{R}^N$ are original function values at these points. N is the number of training points, while d is the dimension of the input function,

$k_{xX} \in \mathbb{R}^N$ is the cross–covariance between the training points and the point x in the input space,

$k_{XX} \in \mathbb{R}^{N\times N}$ is the cross–covariance between the training points,

The subscripts surr are used because in the next sections that posterior process will define the end result of the surrogate model. Moreover, the prior mean $m(x)$ will be replaced by Sparse Grid interpolations to create a Gaussian Process with a sparse prior.

4 Combining Sparse Grids and Gaussian Processes

The method proposed in this section consists of combining two surrogate modeling techniques to create a more powerful method. In other words, the resulting method consists of a two step process, where the output of Sparse Grids interpolation becomes the input of the Gaussian Process and the final result is the sought for surrogate. As already mentioned in the motivation section, the Sparse Grid toolbox incorporates B-Spline bases of different orders and allows for adaptivity. Work by Schöbi et al. in [1] use Polynomial Chaos Expansions instead of Sparse Grids for this purpose. In their method, PCEs focus on the global behaviour of the original function, while the Gaussian Process analyzes its local properties. Having Sparse Grids as local priors, local properties can already be incorporated into the surrogate model at an earlier stage and thus result in a both globally as well as locally accurate surrogate model.

A second challenge for PCEs which is alleviated for Sparse Grid techniques, is posed by non-parametric distributions. Non-parametric distributions come in handy when dealing with the initial stages of building a statistical model, such as a kernel density estimator. Before enough knowledge is obtained to identify which parametric distribution the model belongs to, non-parametric distributions can be used as they are generalized to suit different classes of models. PCEs are challenged in this field because their main building block, the orthonormal polynomial basis, does not exist for non-parametric distributions.

4.1 Notation and Setting Used

To account for the results of the two different meta-modeling methods that are used in this section, different subscripts are used to describe the different stages. The subscript sg refers to the first output of the meta-modeling technique, namely after Sparse Grid discretization is performed. The subscript surr refers to the final result of the whole model, that is the posterior of the Gaussian Process whose original prior was the Sparse Grid interpolation. Moreover, $\tilde{f}(x)$ refers to the surrogate function approximation, while $f_{\mathrm{orig}}(x)$ refers to the original function. With these notations in mind, $\tilde{f}_{\mathrm{sg}}(x)$ refers to the first stage surrogate function and $\tilde{f}_{\mathrm{surr}}(x)$ refers to the final surrogate, which is the end result of this paper. The grid points at which the function evaluations are performed are denoted as X, where X_{sg} refers to the Sparse Grid points and X_{gp} refer to the random points that are originally used to train the Gaussian Process. Since each evaluation of the original function is very computationally expensive and must be avoided wherever possible, maximum possible knowledge exploitation must be made of these evaluations once they are performed. For example, once the Sparse Grid meta-modeling technique is done, X_{sg} and $f_{X_{\mathrm{sg}}}$ evaluations are obtained. These points can be further included in the training set of the Gaussian Process and thus expanding the total number of points to $N_{\mathrm{tr}} = (N_{\mathrm{sg}} + N_{\mathrm{gp}})$. This extended dataset is now defined as $X_{\mathrm{tr}} = [X_{\mathrm{sg}}, X_{\mathrm{gp}}] \in \mathbb{R}^{d \times N_{\mathrm{tr}}}$, where d is the dimension over which the original function is defined.

4.2 Resulting Surrogate Model: Gaussian Process with a Sparse B-Spline Prior

Figure 4 shows how the implementation of the end result surrogate model is performed. First, the original function is interpolated using a Sparse Grid with a B-Spline basis. Different options exist for this step. B-Splines of first order are simple piece-wise linear functions, while they converge towards Gaussians for increasing order. Depending on the required accuracy of the first stage surrogate, $\tilde{f}_{\mathrm{sg}}(x)$, the number of sampled grid points and the order of the B-splines have to be adapted. Note that the subscript X_{sg} refers to the evaluation of the respective function at those grid points, unlike the general function definition $f_{\mathrm{orig}}(x)$ which can take any numerical

value based on the input x.

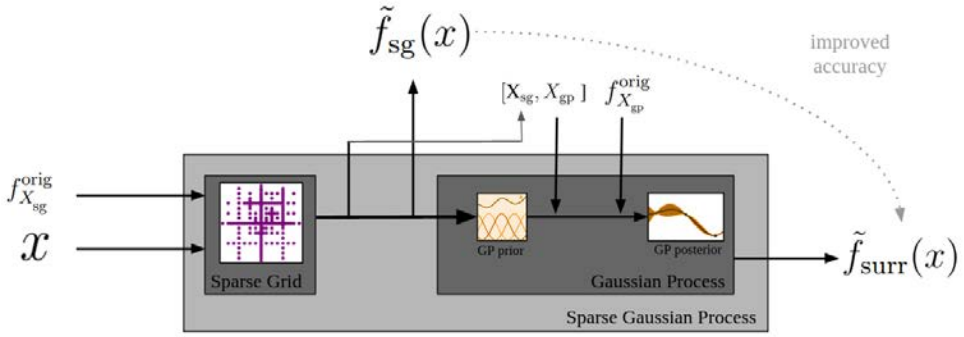

Figure 4: Gaussian Process with a sparse B-Spline prior (SG-Kriging).

The first stage surrogate function $\tilde{f}_{\text{sg}}(x)$ is then used to define the Gaussian Process and acts as the mean function of the prior. To exploit the currently available knowledge about the original function, the Sparse Grid points X_{sg} and the function evaluations at these points $f^{\text{orig}}_{X_{\text{sg}}}$ are then used to expand the training set of the Gaussian Process. Furthermore, Gaussian Process modeling is carried out in its standard way as explained in Section 3.3. The end result is the second stage surrogate $\tilde{f}_{\text{surr}}(x)$, which is a more accurate approximation of the original function and is used to achieve the aim of this paper.

4.3 Hyperparameters of the Surrogate Model

From the definition of the resulting surrogate model as a Gaussian Process with a Sparse B-Spline prior, several important hyperparameters come into play. The Gaussian process that is used in this paper has a squared exponential covariance kernel, defined as

$$k_{\text{surr}}(x, x') = \sigma^2 \exp - \frac{(x - x')^\top (x - x')}{2l^2} \,. \tag{4}$$

The lengthscale parameter, l controls the smoothness of the function, while the signal variance term, σ controls the deviations from the mean. A simple hyperparameter optimization algorithm is implemented in this paper. It is based on Algorithm 2.1 in [13] where the log marginal likelihood is maximized to obtain optimal parameter values. The current implementation is not very stable in all cases, and better methods do exist (such as cross validation), but it does create an improvement for the results obtained here.

4.4 Validation of the Surrogate Model

The end result is a surrogate function, $\tilde{f}_{\text{surr}}(x)$, which approximates the original function, $f_{\text{orig}}(x)$. Model validation is performed by first generating a mesh that spans over the possible range of values in the input space. For example, for interpolating a two-dimensional function, a two-dimensional grid is created to cover the range of possible input values. The grid size is variable and can be increased to create a finer grid and thus perform the validation over more evaluation points. Both the original and the surrogate function are evaluated at these points of the grid and then error measures are calculated to obtain a numerical value resembling the accuracy of the model against the original function. The grid can be created by random subdivisions sizes of the input space or equally sized ones. The formula used to calculate the error

measures is

$$\mathrm{err}_{\mathrm{gen}} = \frac{\sum_{i=1}^{N_{val}} (f_{\mathrm{orig}}(x_i) - \tilde{f}_{\mathrm{surr}}(x_i))^2}{\sum_{i=1}^{N_{val}} (\tilde{f}_{\mathrm{surr}}(x_i) - \tilde{\mu}_{\mathrm{orig}})^2}\,, \tag{5}$$

where N_{val} is the size of the validation set and $\tilde{\mu}_{\mathrm{orig}}$ is a constant and it indicates the mean value of the original function. The formula is the same as was used in [1] and gives a measure of the relative generalization error of the surogate model.

5 Experiments

This section presents the results obtained from testing the Gaussian Process with a Sparse Grid B-Spline prior on analytical performance test functions, such as the Rastrigin function. These functions are chosen because they are well-known in the field of optimization and provide common ground for benchmarking comparisons. The Rastrigin funciton is a particularly interesting function because of its multiple local minima in each dimension. The error is computed at the grid points of a regular grid of size 150^2; see Section 4.4.

5.1 Scope of Application

In the current work, the method presented here is used to create a surrogate model to approximate the original function over the entire search space. Therefore, throughout all the experiments conducted here, the adaptivity criterion is either completely deactivated or set to a very low value. Adaptivity in the context of Sparse Grids relates the grid refinement to the value of the original function at the evaluated grid points. For an optimization task, such as finding the global minimum, grid points can be created only in the regions with the lowest known function values. This aims to obtain a higher degree of accuracy around the global minumum. Search-based testing, which is the future scope of this project, will benefit from this adaptive refinement criterion, as it would speed up the search process by guiding it towards obtaining good solutions (or possibly bad if fail cases are being searched for).

5.2 Rastrigin Function

The Rastrigin function, originally developed by Leonard A. Rastrigin [14], is a non-convex function that is frequently used to test global optimization algorithms. We selected the Rastrigin function as it is a very challenging function that represents characteristics of our target scenario: We have plenty of local minima, and the function is in general smooth. One restriction, however still remains, namely that there often exists in automated driving scenarios, cases with switches which can no longer be modeled by a smooth function. Originally it was defined for two dimensions with an input search space $x_i \in [-5.12, 5.12]$ for $i = 1, 2$. It can, however, be generalized for higher dimensions. Moreover, the input search spaces in this paper are unit hyper-cubes of the form $x_i \in [0, 1]$ for $i = 1, 2, ..., n$. Therefore, the Rastrigin function is re-scaled so that its multiple local minima lie in this reduced search space between 0 and 1. We use the Rastrigin function in the form

$$f(x) = 20 + \sum_{i=1}^{2} ((10x_i - 4)^2 - 5\cos 2\pi(10x_i - 4)) \quad \text{for } x_i \in [0, 1]\,, \tag{6}$$

see Figure 5 for an illustration. The factor 10 scales the function to create multiple local minima in the reduced search space, while the shifting by 4 shifts the global minimum so that it is no longer located at the center of the Sparse Grid.

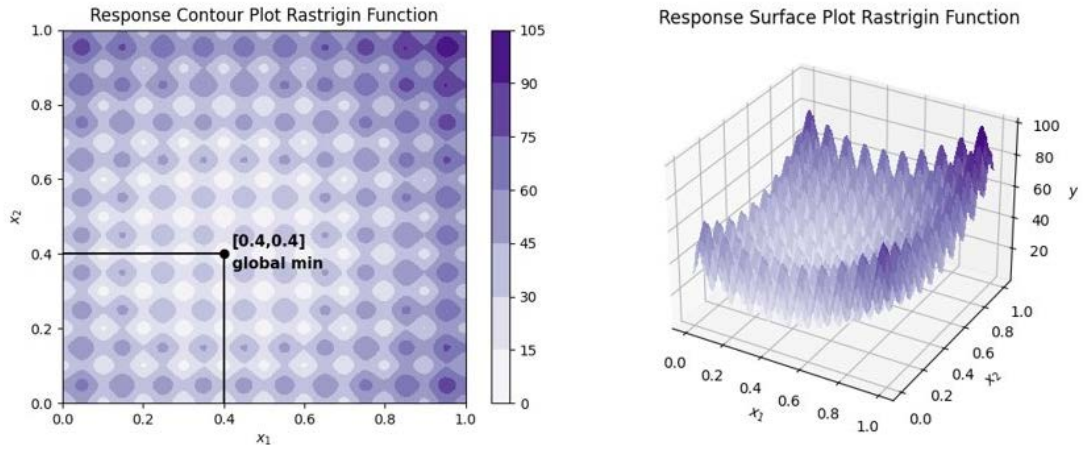

Figure 5: The scaled and translated Rastrigin function as used in the experiments. It is smooth with many local minima and a single global minimum.

6 Results and Comparison

This section uses the Rastrigin function to compare between the different surrogate model types. The aim of this part is to view the performance of the Gaussian Process and the Sparse Grids as separate surrogate modeling techniques and compare it to the combination between the two individual methods. First, the pure Gaussian Process is tested, i.e., one with zero prior knowledge. Second, the Sparse Grid method is tested. And third, the two methods are combined, where the Sparse Grid surrogate is used as a prior for the Gaussian Process. Violin plots are used to display the results, as they depict both the statistics of the results as well as their density. The width of the violin plot indicates how often a certain value occurs in the data. A wider section indicates a higher probability of occurrence, while a narrower section indicate lower probability. The purple marker indicated in the figures shows the mean value, while the orange marker indicates the median.

6.1 Pure Gaussian Process with Zero Mean Prior

The first set of results in Figures 6 and 7 shows the performance of interpolating the Rastrigin function defined by Equation (6) using a pure Gaussian Process. In other words, this is a standard Gaussian Process that uses a zero mean prior function. The covariance kernel that was used throughout this paper is the squared exponential kernel. There are two main hyperparameters that are defined for this kernel, i.e., the length scale and the overall variance of the function. The general trend obeserved in the figures is,that the accuracy improves as the number of samples in increased. The violin plot is used to show the results for 10 experiment runs per sample size; allowing for statistical as well as distribution inference. Moreover, a simple hyperparameter optimization algorithm based on maximizing the log mariginal likelihood, as defined in Algorithm 2.1 in [13] is implemented and its results are show in Figure 7. It can be noticed that the hyperparameter optimization significantly improves the results.

6.2 Pure Sparse Grids Surrogate

Figure 8 shows the results of creating the surrogate model using only the Sparse Grid method. For this purpose, modified not-a-Knot B-Splines of degree 3 are used as basis functions. The

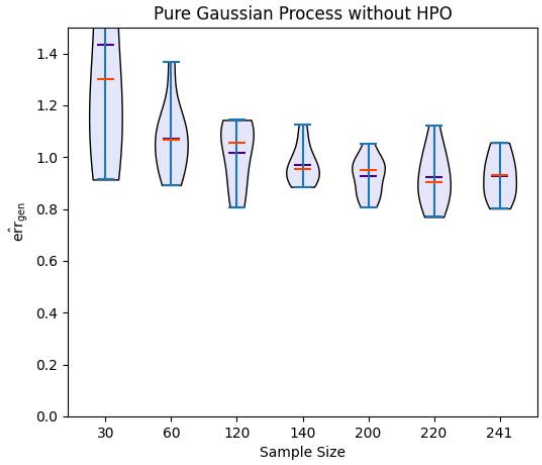

Figure 6: Pure Gaussian Process with zero mean prior without hyper-parameter optimization.

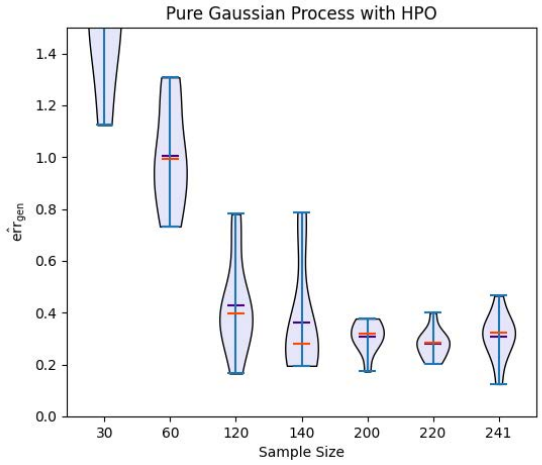

Figure 7: Pure Gaussian Process with zero mean with hyper-parameter optimization.

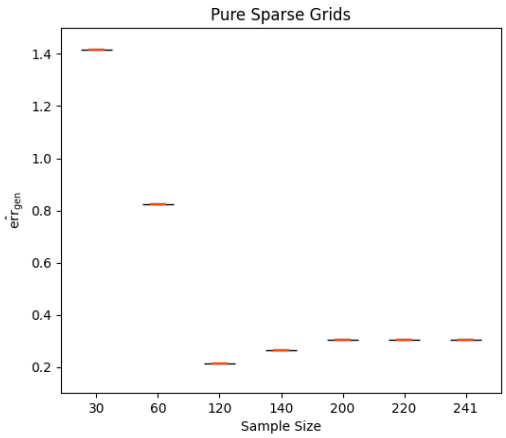

Figure 8: Pure Sparse Grids interpolation.

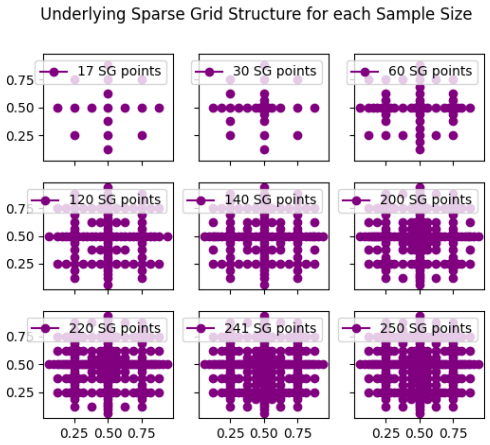

Figure 9: Underlying grid structure for each number of evaluation points.

plot shows that increasing the number of sampling points does not necessarily improve the accuracy of the model. The reason behind this is a general problem that exists for interpolating a function over a discretized grid. Based on the number of grid points that are evaluated and on their location in the input search space, there is always a risk that the overall function values are over- or under-estimated based on where the underlying function is evaluated. Figure 9 shows the underlying Sparse Grid structure of each of the four discretizations. Note that we choose a grid refinement independent of the target function in this first proof of concept—in particular, we employ no hyperparameter optimization. For the use as a simple prior for Gaussian Process Interpolation in a subsequent step, this is, however, already sufficient to achieve good results.

6.3 Combining the Two Methods: Gaussian Process with Sparse Grid B-Spline Prior

This section presents different experiments that show the performance of the SG-Kriging method under varying settings. It also aims to show that improving a Gaussian Process prior using a Sparse Grid Surrogate enhances the end performance. Contour plots from the first and

second stage surrogates as well as violin plots of the generalized error measures of the end model are used to visualize the performance of SG-Kriging. Finally, comparisons are made between different levels of priors to show that the developed method is sensitive to the percentage of Sparse Grid points used.

6.3.1 Preliminary Results of SG-Kriging

A simple experiment, shown in 10 is first performed using only 17 Sparse Grid points for the prior. The total number of evaluation points used for both the prior and the Gaussian Proces are shown on the horizontal axis. These preliminary results already show that using this improved prior reduces the generalized error of the resulting model. Comparing these results to each of the individual runs in Figures 7 and 8 shows that SG-Kriging behaves better, even at low number of sampling points.

Moreover, the contour plots in Figure 11 show different levels of Sparse Grid priors being used. Each of the columns in the figure refer to a different number of Sparse Grid points. The top row shows all the priors, while the bottom row shows the corresponding posteriors. For these five experiment runs, the total number of evaluation points for Sparse Grids plus Gaussian process are fixed to 241. That is, for example, when 47 points are used for the Sparse Grids prior, the rest adding up to 241 are used by the Gaussian Process to obtain the posterior. These results show that the prior, with increasing numbers of evaluation points is able to capture the properties of the Rastrigin function despite its many local minima. Each posterior is shown to be an improvement over its corresponding prior. It can also be noticed that the Gaussian Process at all levels improves the performance around the boundaries of the function. This shows that the extrapolation done by Sparse Grids at the boundaries can be made more accurate using a Gaussian Process. One question that still remains open is how to choose the optimal number of Sparse Grid points at different sampling sizes. The next experiment run addresses this issue by studying the effect of changing the percentage of Sparse Grid points on the end results.

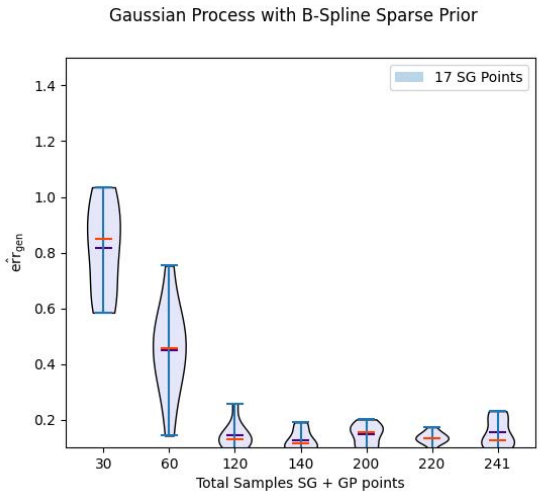

Figure 10:

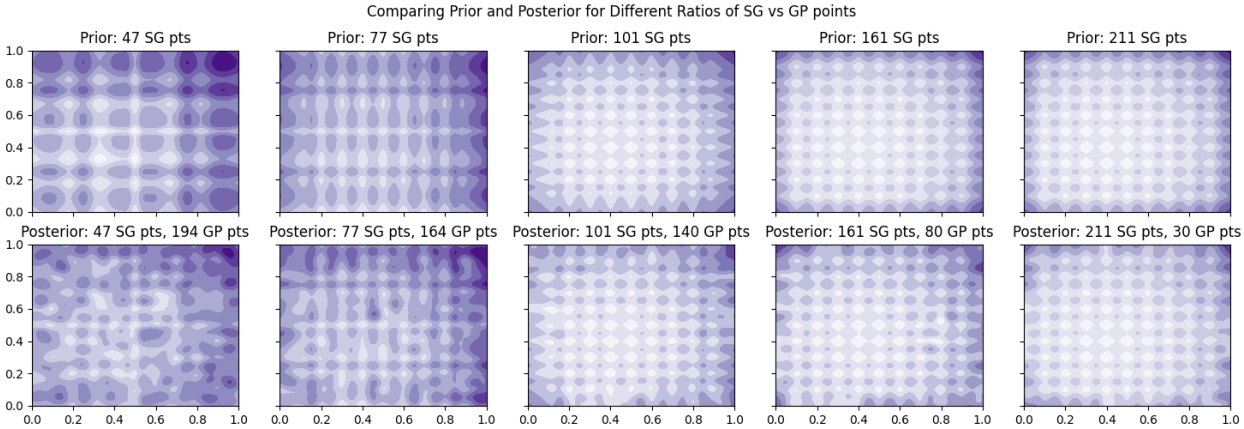

Figure 11: For a total of 241 evaluation points spent per run, the figure shows how the posterior is an improvement of the prior. Each column represents one run, with the top figure showing the prior and the bottom one shows the posterior. Results show that having a Gaussian Process on top of the Sparse Grid surrogate model does indeed improve the results.

6.3.2 Sensitivity of SG-Kriging to the Percentage of Sparse Grid Points in the Prior

In the third set of experiments, shown in Figure 12 we examine how the ratio between Sparse Grid points and random training for the Gaussian Process influences the accuracy of the resulting surrogate model. The twelve plots show the results obtained from incrementally increasing the ratio of the number of Sparse Grid points used in SG-Kriging for fixed numbers of total evaluation points. The two plots on the extremes, that is the top left one and bottom right one show the two separate methods. The top left is for a pure Gaussian Process with zero mean prior, while the bottom right one is for pure Sparse Grids. In the plots, the horizontal axis shows the total number of evaluation points used, while the legend indicates the percentage of Sparse Grid points (up to rounding and refinement-based deviations). We observe high errors for both the individual methods working separately. Moreover, adding Sparse Grid points does improve the results, indicating that a good prior is crucial. The results tend to worsen again at percentages of Sparse Grids points higher than 70%, which is indicating that the Sparse Grids method is becoming more dominant. The current implementation of Sparse Grids is still not improving the accuracy with increasing number of evaluation points, as was seen in Figure 8. This limitation is now affecting the performance of the combination method and will be addressed in future work. These results show that SG-Kriging is sensitive to the relative and absolute amount of Sparse Grid points spent, and that some sort of automatic hyperparameter optimization will be required in the future. However, for up to 80% Sparse Grid points inclusion, the resulting surrogate model behaves better than any of the two separate methods, showing that the proposed method is promising and raising the interest for its future development.

7 Summary and Future Work

This paper presents first results of an alternative technique to combine two meta-modeling methods to create an accurate and computationally efficient surrogate model that will be used instead of the original expensive function. The resulting SG-Kriging method is obtained by incorporating a Sparse Grid interpolation of a function as the mean prior term of a Gaussian Process. The resulting Gaussian posterior is what is regarded as the surrogate model that will

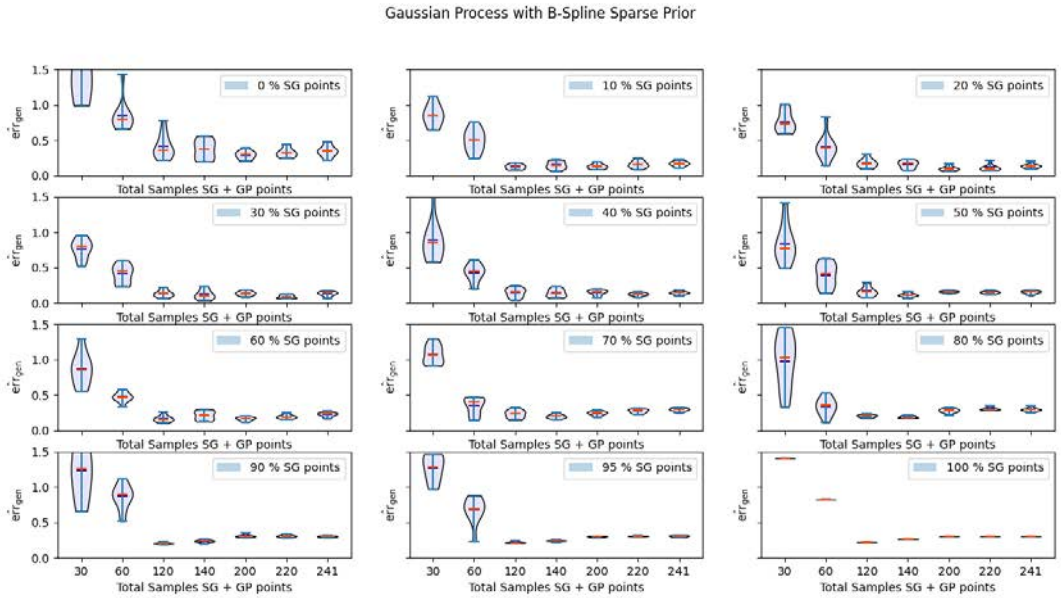

Figure 12: SG-Kriging using different percentage combinations of Sparse Grid Points. The best results are achieved for intermediate percentages of Sparse Grid points, depending on the total number of grid points spent. In most cases, the range of optimal values is $30\% - 50\%$ for Sparse Grid points versus Gaussian Process points.

replace the original function and is better for a given total number of evaluation points than each of the two methods individually. Our results furthermore show for a difficult analytic function, that the relative amount of effort (function evaluations) spent to improve the prior is a new important parameter of our proposed SG-Kriging method. This work contains only a first proof of concept and has many promising extensions. Future research in this topic includes implementing hyper-parameter optimization algorithms that will optimally calibrate the kernel and the percentage of the Sparse Grid points. Furthermore, the results presented here have been obtained for an uninformed adaptive refinement, which calls for further research regarding the adaptive refinement criterion of the Sparse Grid.

REFERENCES

[1] Roland Schöbi, Bruno Sudret, and Joe Wiart. Polynomial-chaos-based kriging. *International Journal for Uncertainty Quantification*, 5(2):171–193, 2015.

[2] Roland Schöbi. *Surrogate models for uncertainty quantification in the context of imprecise probability modelling*. PhD thesis, ETH Zürich, Zürich, Switzerland, 2017.

[3] Christoph Zenger. Sparse Grids. In Wolfgang Hackbusch, editor, *Parallel Algorithms for Partial Differential Equations*, volume 31 of *Notes on Numerical Fluid Mechanics*, pages 241–251. Vieweg, 1991.

[4] Hans-Joachim Bungartz and Michael Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.

[5] Dirk Pflüger. *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. Verlag Dr. Hut, München, August 2010.

[6] Julian Valentin. Hierarchische Optimierung mit Gradientenverfahren auf Dünngitterfunktionen. Master's thesis, University of Stuttgart, Department of Computer Science, IPVS,, 2014.

[7] Dirk Pflüger et al. SG++ – software for spatially adaptive sparse grids, 2012.

[8] Dirk Pflüger. Adaptive sparse grids and the curse of dimensionality, 2013. *Seminar Multiscale Modeling and Simulation, Cluster of Excellence Engineering of Advanced Materials*.

[9] Julian Valentin and Dirk Pflüger. Hierarchical gradient-based optimization with B-splines on sparse grids. In *Sparse Grids and Applications - Stuttgart 2014*, pages 315–336, Cham, 2016. Springer International Publishing.

[10] Jochen Garcke. Sparse grids in a nutshell. In Jochen Garcke and Michael Griebel, editors, *Sparse Grids and Applications*, pages 57–80, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[11] Sergey A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions, 1963. pp. 10421043. English Russian Translation: Soviet Mathematics Doklady, Vol. 4, 1963, pp. 240-243.

[12] Julian Valentin. *B-Splines for Sparse Grids: Algorithms and Application to Higher-Dimensional Optimization*. PhD thesis, University of Stuttgart, Germany, 2019.

[13] Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[14] Leonard Andreevič Rastrigin. Systems of extremal control. *Nauka*, 1974.