A Novel and Fully Automated Coordinate System Transformation Scheme for Near Optimal Surrogate Construction

Johann Mynhardt Bouwer, Daniel Nico Wilke, Schalk Kok

January 26, 2024

Abstract

This work develops a novel coordinate system transformation scheme to improve the performance of common radial basis function surrogate models. This coordinate system transformation scheme is based on the fact that commonly used basis functions are isotropic.

Three main empirical findings are established in this study. Firstly, in general isotropic functions are inadequate to describe anisotropic data manifolds due to a mismatch between the functional form and the form of the data manifold resulting in poor generative performance. Counter-intuitively, utilising additional gradients during surrogate training often worsens the generative capability.

Secondly, component-wise scaling of isotropic model forms during cross-validation is inadequate to enhance the functional form of the data manifold form as anisotropic coupling in the data manifold remains coupled. Improving the match between the functional form and the data manifold form requires both rotation and scaling.

Thirdly, the coordinate system transformation scheme should predominantly be based on a collection of local curvature estimations and not on global curvature approximations. Gradients are critical to estimating the local curvature for identifying a near-optimal reference frame for surrogate construction, which then translates to additional benefits of gradients in gradient-enhanced surrogates.

Based on the above observations, this paper proposes an isotropic transformation for the data coordinate system that performs near-optimal transformations on lower dimensional data without requiring any cross-validation. The method is compared against commonly applied component-wise cross-validation data coordinate system scaling as well as the more modern Active Subspace Method on a carefully crafted decomposable test problem, which has a known optimal coordinate system, that varies between 2 and 16 dimensions.

The paper concludes after demonstrating that the developed transformation scheme, as well as the other common methods, will offer little benefit on non-decompose problems and offers some suggestions on future work to create a more general isotropic transformation.

1 Introduction

The work completed in this research is focused on the impact that a suitable coordinate system transformation pre-processing step will have on the performance of a surrogate model. In surrogate model research, specifically in the context of surrogate based optimisation (SBO), the model is used to replace a computationally expensive function, which often includes some finite element (FE) or computational fluid dynamics (CFD) simulations. The areas of SBO research can broadly be separated into four main areas, shown in Figure 1.

Most of the current research into surrogate models focuses on the training step (Step 3) of the process. This research includes various basis function cross validation strategies [1, 2, 3, 4], component scaling methods [5, 6], the regression of high fidelity and low fidelity information [7, 8], and the inclusion of gradient information into the model [2, 9]. In the case where gradient information is included directly into the model, specifically in basis function based models, often the model does not experience the performance improvement expected from including highly information dense gradient vectors [10, 2].

The work in this paper critically demonstrates that the reason the inclusion of gradient information does not offer the expected performance improvement is the model bias that the isotropic assumption introduces into the model [3, 1]. The isotropic assumption refers to the fact that the model assumes similar output variation given some input perturbation, regardless of the direction of the input perturbation. Only when this assumption is formally addressed, in the case of this work as a pre-processing step, does the inclusion of gradient information offer the expected improvement to the predictive performance of the surrogate model.

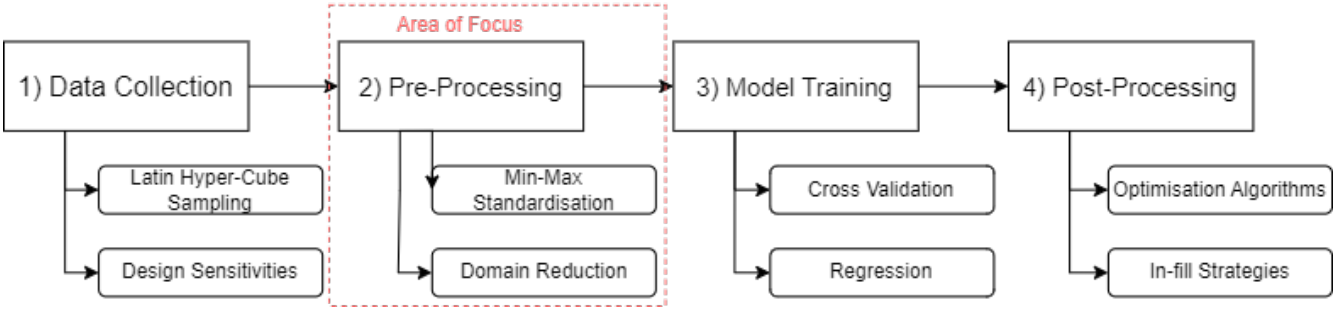

Figure 1: Flow diagram showing the main areas of research in the context of surrogate based optimisation (SBO) as well as some of the main techniques used in each step.

Common radial basis function implementations inherit the isotropic assumption and typical cross validation strategies only scale the curvature of the model uniformly in all directions [5, 7]. Therefore, some mechanism is needed to scale the curvature non-uniformly for any general function, to reduce the model bias with respect to the sampled data. One strategy to accomplish this is to perform component-based scaling of the coordinate system, or to perform component-based scaling of the basis functions of the model. This approach assumes that all the variables in the problem independently impact the outcome of the function, i.e. the variables in the problem are uncoupled. This work will demonstrate that component-based scaling is an inadequate approach to reduce model bias, in the presence of the isotropic assumption. Rather, a full transformation, i.e. some rotation and scaling of the coordinate system, is needed. Figure 2 presents this argument graphically, where it is shown that if the data has a rotated elliptical contour (not isotropic), a full transformation is needed to address the model bias that the isotropic assumption introduces.

Therefore, this paper proposes a consistent and tractable method to estimate a coordinate system in which the model bias is lessened and the isotropic assumption is reasonable. This paper then demonstrates that in this new coordinate system the inclusion of gradient information offers the expected improvement in predictive performance of the model. The proposed transformation scheme is most accurate and efficient if it makes use of sampled gradient information.

The layout of the paper is then as follows. Firstly, a discussion of related research is offered followed by a more in depth demonstration of the isotropic assumption central to the research completed in this paper. The proposed transformation scheme is then derived in Section 4 and a test problem and its important characteristics are discussed in Section 5. Results are then generated in Section 6 where the proposed transformation scheme is compared to standard and more modern approaches. Section 7 demonstrates the limitations of the proposed method in the case of non-decomposable functions. Lastly, some conclusion and recommendations for future work are offered. All the derivations of common surrogate models are offered in the appendix.

2 Related Work

In general, the unconstrained optimisation problem attempts to find some vector of design variables, $\boldsymbol{x} = [x_1, x_2, ..., x_n]^{\mathrm{T}} \in \mathcal{R}^n$, that minimises some scalar function $F(\boldsymbol{x}) : \mathcal{R}^n \rightarrow \mathcal{R}$. In many modern engineering optimisation problems, the evaluation of the function $F(\boldsymbol{x})$ often includes a computationally expensive simulation.

Although the other areas of research shown in Figure 1 are outside the scope of the research completed in this work, the standard methods, and their alternatives, that are implemented in this work are discussed in this section.

2.1 Data Collection

The most common method used to sample the design coordinate system is the Latin Hyper-Cube sampling strategy. There are many versions of this method in which additional criteria are placed on the locations of the samples in the coordinate system such as maximising the average distance between the points or minimising the correlation between the points. In this study the samples are located using defacto-standard LHS sampling without the space-filling condition enforced [11].

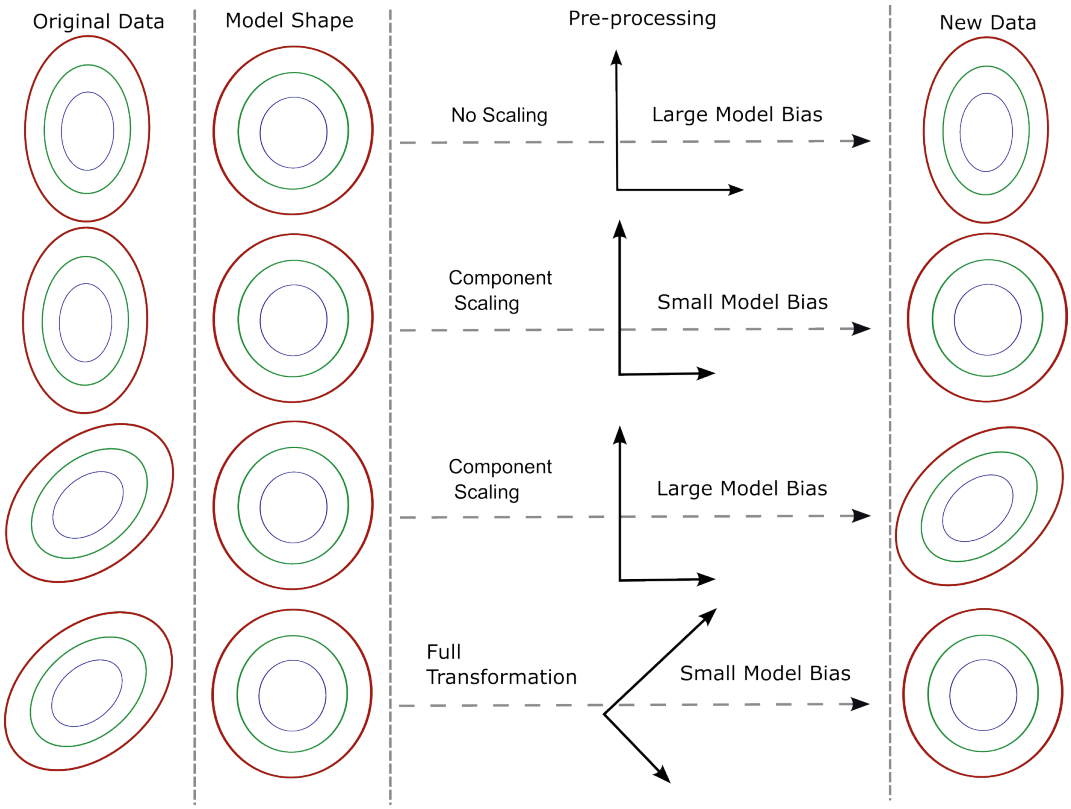

Figure 2: Visual demonstration of how different pre-processing strategies applied to different datasets that exhibit behaviour ranging from uncoupled to scaled and coupled, can lessen the model bias in basis function surrogate models.

During the sampling phase it is also possible to obtain gradient information at the sampled locations in the design coordinate system. Although many papers [12, 7, 13, 1] assume that in this scenario gradient information of the function is not available, it is often not the case. Many papers [14, 15, 16, 17] detail procedures to calculate the design sensitivities for functions that are computed using the Finite Element Method (FEM) or Computational Fluid Dynamics (CFD). Many finite element packages have adjoint sensitivities implemented, for example, Calculix [18]. This gradient information can be calculated with respect to many different design variables to perform optimisation in a wide range of problems such as shape optimisation, thermodynamics, and vibration analyses [14, 16, 19, 20, 21]. Therefore, in this work sampling scenarios with and without gradient information are considered.

2.2 Common Surrogate Models

Surrogate models can be classified into function-value based, gradient-enhanced, and gradient-only [22]. Note that surrogate models that regresses through both function value and gradient information are referred to as either gradient-enhanced (GE) models [10, 2], cooperative models (CO) [23, 24], or first order (FO) models [25, 22]. For the remainder of this research gradient-enhanced (GE) is used to describe surrogate models that regresses through both gradient and function value information. Common surrogate models include Kriging Models, Radial Basis Functions (RBF) and polynomial surrogate models [1, 12, 5, 3, 4, 2]. In this paper the function value (FV-RBF) and gradient enhanced RBF (GE-RBF) models are implemented. The derivations for these models are offered in literature or in the appendix.

2.2.1 The Isotropic Assumption

The isotropic behaviour of RBF surrogate models arises from the basis functions used in their implementation. This section demonstrates how this assumption is present in the basis functions as well as why this assumption is detrimental to the performance of common surrogate models. Some of the most common basis functions include

1. Gaussian: $\phi(\mathbf{x}, \mathbf{c}, \epsilon) = e^{-\epsilon||\mathbf{x}-\mathbf{c}||^2}$,
2. Inverse quadratic: $\phi(\mathbf{x}, \mathbf{c}, \epsilon) = \frac{1}{\sqrt{||\mathbf{x}-\mathbf{c}||+\epsilon^2}}$,
3. Multi-quadratic: $\phi(\mathbf{x}, \mathbf{c}, \epsilon) = \frac{1}{1+\epsilon||\mathbf{x}-\mathbf{c}||}$,

where the variable ϵ is referred to as the shape parameter, the point $\mathbf{c}$ is the centre of the basis function, and $\mathbf{x}$ is the point being evaluated. During the training of the surrogate model the hyper-parameter ϵ is found, as well as the amplitude of each basis function. The contour of these basis functions are shown in Figure 3.

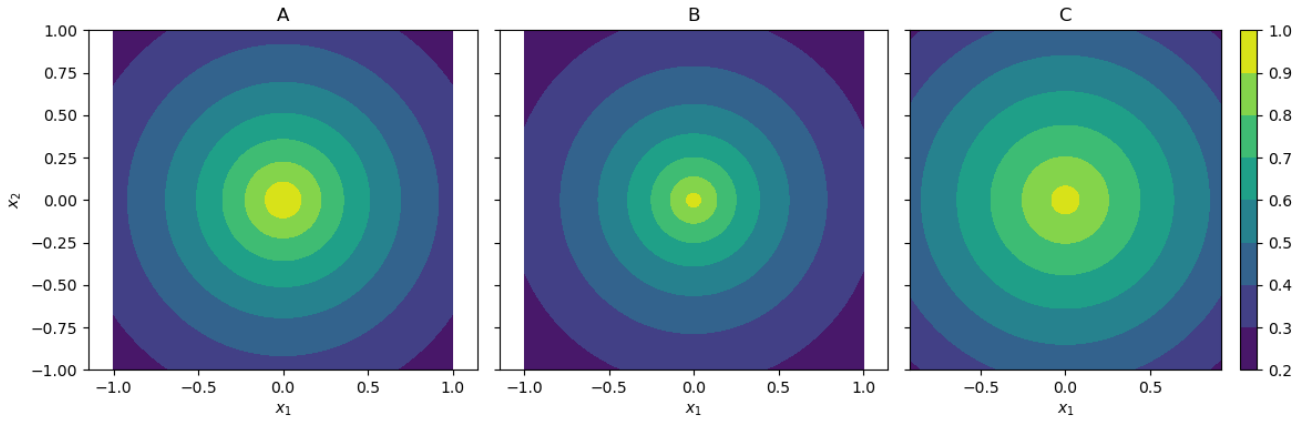

Figure 3: Contour plots of the Gaussian, Inverse Quadratic, and Multi-quadratic basis function with $\epsilon = 1$ in Subfigures A, B, and C respectively.

From these contour plots it is clear that the basis functions are symmetrical and therefore make the assumption that all the variables in the problem are uncoupled and have an equal impact on the outcome of the problem.

The only tuneable parameters of these basis functions are the shape parameter ϵ and the amplitude. The shape parameter can be determined with a k-fold cross-validation or Leave-Out-One cross-validation (LOOCV) approach [26, 12]. In this research, various shape parameters between 10^{-2} and 10^{1} are evaluated using k-fold cross-validation as a metric and the shape parameter associated with the lowest error is selected. The basis function amplitudes are computed using linear algebra (see the appendix for details).

Although this shape parameter is optimised to fit the underlying function, it can only uniformly adjust the curvature in all the directions in the design coordinate system. This uniform change in curvature can be demonstrated algebraically by deriving the second derivative of the Gaussian basis function:

$$\frac{d^2\phi(\boldsymbol{x}, \boldsymbol{c}, \epsilon)}{d\boldsymbol{x}^2} = -2\epsilon\frac{d\phi}{d\boldsymbol{x}}(\boldsymbol{x} - \boldsymbol{c})^T - 2\epsilon\boldsymbol{I}\phi(\boldsymbol{x}). \tag{1}$$

If the second derivative is evaluated at the point $\boldsymbol{x} = \boldsymbol{c}$, this results in

$$\left.\frac{d^2\phi(\boldsymbol{x}, \boldsymbol{c}, \epsilon)}{d\boldsymbol{x}^2}\right|_{\boldsymbol{x}=\boldsymbol{c}} = -2\epsilon\boldsymbol{I}, \tag{2}$$

where $\mathbf{I}$ is an identity matrix. Therefore, the act of altering or optimising the shape parameter clearly results in an equal change in the curvature of the basis function (at the centre) in all directions.

Therefore, if one shape or scale parameter is used for all directions, the model makes the implicit assumption that the underlying function is isotropic as the basis functions used in its construction are isotropic or symmetric.

However, it is unlikely that a practical engineering design or optimisation problem will utilize variables that all have equal (or at least similar) impact on the outcome of the design, and therefore, a large model bias will be present negatively impacting the performance of the model.

2.3 Anisotropic Scaling Strategies

There are many cross-validation strategies in literature that attempt to alleviate the negative consequences of the implicit isotropic assumption. These strategies include

- component-wise scaling of the coordinate system, i.e. distinct scaling factors per dimension, as an attempt to recover isotropy after scaling [5, 6],
- adapting the basis function to explicitly handle anisotropic functions by using a shape parameter for each principal direction in the design coordinate system [1, 2, 3, 4],
- or more recently, implementing the so-called Active Subspace Method (ASM) [27, 28, 29, 30].

The first two strategies, scaling the coordinate system or using multiple shape parameters, are referred to as the Kriging hyper-parameter optimisation problem. Here an n-dimensional space needs to be searched in order to find the optimum parameters. Therefore many papers apply some global optimiser to solve this problem, such as the Genetic Algorithm (GA) or Particle Swarm Optimisation (PSO) [3]. In higher dimensions, this becomes computationally expensive, so much so that it can become the bottleneck in computation time for SBO. Toal *et al.* [3] investigated four different tuning strategies on problems varying from 1D to 30D. Each of the tuning strategies sampled the model 10 000 times before a set of hyper-parameters was selected.

Other papers attempt to reduce the number of hyper-parameters in the model. Bouhel *et al.* [2, 4] used a partial-least squares (PLS) method to introduce new kernels based on the information from the PLS method. The number of hyper-parameters is then reduced to the number of principal components (PCs) the designer decides to keep based on the information gathered from the PLS method. The ideal number of PCs to be retained depends on the problem as well as the location of the sampled points. There is currently no consistent method to determine this value.

The problem with the first two strategies is that the the surrogate models become computationally intractable to construct for higher dimensional problems (typically ≥ 10) [2], and in the case where the variables are coupled (see Figure 2), the model bias is still large as no rotation of the coordinate system takes place.

For comparison purposes, in this research a simplex search algorithm, such as that used by Toal *et al.* [3], is implemented to find optimum scaling values for the Kriging hyper-parameter problem. To keep the computational costs reasonable, as well as competitive with the other methods implemented, the algorithm is limited to 100 iterations for 5 initial scaling vectors.

2.4 The Active Subspace Method

Although the Active Subspace Method (ASM) is typically a dimension reduction technique, it shares some similarities with the developed transformation scheme in this work. This method finds a lower dimensional reference frame, referred to as the active subspace, that captures the most variance in a function [27].

This method is described in detail by Constantine *et al.* [27], but a brief overview needed for implementation is offered here. The method begins by assuming that gradient information of the underlying function is available. It then constructs the following $n \times n$ matrix

$$\mathbf{C} = \mathbb{E}(\nabla f(\mathbf{x}) \nabla f(\mathbf{x})^T), \tag{3}$$

where $\mathbf{C}$ can be seen as the covariance matrix of the gradient vector. In case of SBO applications the gradient vector $\nabla f(\mathbf{x})$ is only available at discrete sampled locations. Therefore, the matrix $\mathbf{C}$ is approximated with

$$\mathbf{C} = \tilde{\mathbf{C}} = \frac{1}{p} \sum_{i}^{p} \nabla f(\mathbf{x}_i) \nabla f(\mathbf{x}_i)^T, \tag{4}$$

where p is the number of samples of the underlying function and $\nabla f(\mathbf{x}_i)$ is the gradient at these locations.

The matrix $\tilde{\mathbf{C}}$ can then be decomposed into the form

$$\tilde{\mathbf{C}} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{5}$$

where $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ are the eigenvectors and eigenvalues respectively. The eigenvalue matrix takes the form

$$\boldsymbol{\Sigma} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}, \tag{6}$$

where λ_n is the eigenvalue associated with the $n-$th eigenvector.

In the case where coordinate system reduction is implemented only m of the n eigenvectors are used to rotate the coordinate system, where the m vectors are the ones associated with the largest eigenvalues. In this work all n eigenvectors are used as only the case where a full coordinate system transformation is completed is considered. Lastly, the square root of the eigenvectors are then used to scale the coordinate system so that the variance in each direction is approximately equal. This step is occasionally omitted [31, 30], but it is included here as proposed in the original formulation [27]. The ASM method can then be summarised as

1. Compute $\tilde{\mathbf{C}}$ using Equation (4) from the sampled data set.
2. Decompose $\tilde{\boldsymbol{C}}$ into its eigenvalues and eigenvectors $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ respectively.
3. Rotate the coordinate system using the eigenvectors $\boldsymbol{V}$ and then scale each direction i with $\sqrt{\lambda_i}$.

3 Effect of Different Coordinate Systems

To demonstrate how different coordinate systems can impact the performance of a RBF model, the following uncoupled 2D function with each dimension in the domain $x_i \in [0, 1]$ is considered:

$$F(\boldsymbol{x}) = \sin(2\pi x_1) + \sin(2\pi x_2). \tag{7}$$

The effect that the scaling and rotating of the coordinate system has on the performance of the RBF surrogate model is demonstrated by defining two new coordinate systems. Firstly, a scaled coordinate system $\boldsymbol{x^*}$ is defined in which the inputs of the function are scaled using the equation

$$\boldsymbol{x^*} = \boldsymbol{S}\boldsymbol{x}, \tag{8}$$

where the matrix $\boldsymbol{S}$ is defined as

$$\boldsymbol{S} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}. \tag{9}$$

The scaled coordinate system $\boldsymbol{x^*}$ is then rotated to the coordinate system $\hat{\boldsymbol{x}}$. In this coordinate system, the function becomes coupled. The coordinate system transformation is given by

$$\hat{\boldsymbol{x}} = \boldsymbol{R}\boldsymbol{x^*} = \boldsymbol{R}\boldsymbol{S}\boldsymbol{x} \tag{10}$$

where the rotation matrix $\boldsymbol{R}$ is defined as

$$\boldsymbol{R} = \begin{bmatrix} \cos(30°) & -\sin(30°) \\ \sin(30°) & \cos(30°) \end{bmatrix}. \tag{11}$$

The functions in these three coordinate systems, namely the "original", "scaled", and "scaled and rotated" coordinate systems are shown in Figure 4.

Three RBF surrogates are then constructed using various sample numbers (varying from 10 to 25), one in the "original" coordinate system, one in the "scaled" coordinate system, and lastly one in the "scaled and rotated" coordinate system.

The performance of each surrogate is measured at 1000 randomly sampled test points. The number of test points is selected so much higher than the number of construction points to ensure that the error measure is an

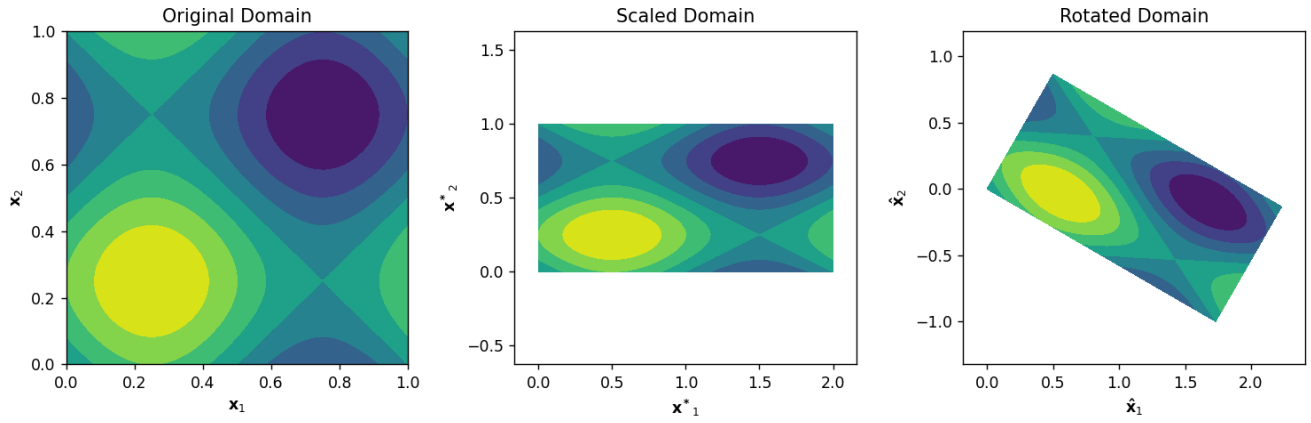

Figure 4: Contour plots of the example function illustrated in the "original" coordinate system, "scaled" coordinate system and "scaled and rotated" coordinate system.

accurate reflection of the quality of fit, and is not affected by the location of the test points. To account for the randomness present in the location of the construction points, the error calculation is repeated 50 times and the mean is recorded. To evaluate the dependency of the surrogates on the locations of the construction points, a measure of the variance of the surrogates is recorded. This is done by taking the variance of the error for each point in the test set and then recording the mean of this variance across all the points. Ideally, this result should be zero, otherwise, the surrogate greatly depends on the randomness of the sampling technique.

The performance measure used is the Root Mean Square Error (RMSE), expressed by

$$\text{RMSE} = \sqrt{\frac{\sum_i^N (V_T^i - V_P^i)^2}{N}} \tag{12}$$

where V_T^i is the target value and V_P^i is the predicted value from the surrogate. The results are shown in Figure 5 where the shaded region in the plots indicates the mean variance (averaged over 50 instances) of the surrogates.

Clearly, the coordinate system the surrogate is constructed in has a meaningful and measurable impact on the performance of a surrogate. The transformed coordinate systems, i.e. $\boldsymbol{x}^*$ and $\hat{\boldsymbol{x}}$, negatively impacted both the performance of the surrogate (increased error), as well as the consistency of the surrogate (increased variance), especially at lower sampling densities. One can also see the benefit of a complete transformation (rotation and scaling) that would transform the problem back from the rotated $\hat{\boldsymbol{x}}$ coordinate system to the original $\boldsymbol{x}$ coordinate system.

The total error of a surrogate, E_T, can then be defined as a summation of two errors. The first is the error associated with the sparsity of information, E_S, and the second is the error associated with the coordinate system the surrogate is constructed in, E_D. These errors are indicated in Figure 6.

4 Proposed Transformation Scheme

The goal of the developed transformation scheme is to recast the problem into a coordinate system where the problem is isotropic, i.e. the variables are uncoupled and have an equal impact on the outcome of the function. From the discussions presented in Sections 2.2.1 and 3 it is clear that, firstly, RBF models struggle to accommodate anisotropic functions, and secondly, a coordinate system transformation step can recast the problem into a more isotropic coordinate system. This section will critically demonstrate that curvature information is what is needed to create a general transformation scheme, and that this curvature information needs to be obtained from a collection of *local* estimations and not a single *global* estimation.

4.1 Second-Order Non-linearity

To begin this argument consider a simple quadratic function given by

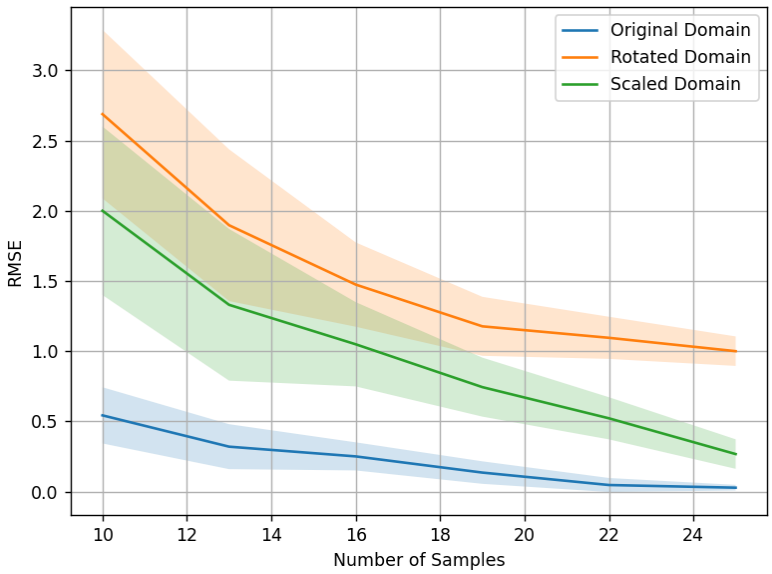

Figure 5: The mean RMSE (solid lines) and the mean variance (shaded regions) in the RMSE for the surrogates constructed in the three coordinate systems for an increasing number of samples. Means are computed across 50 instances.

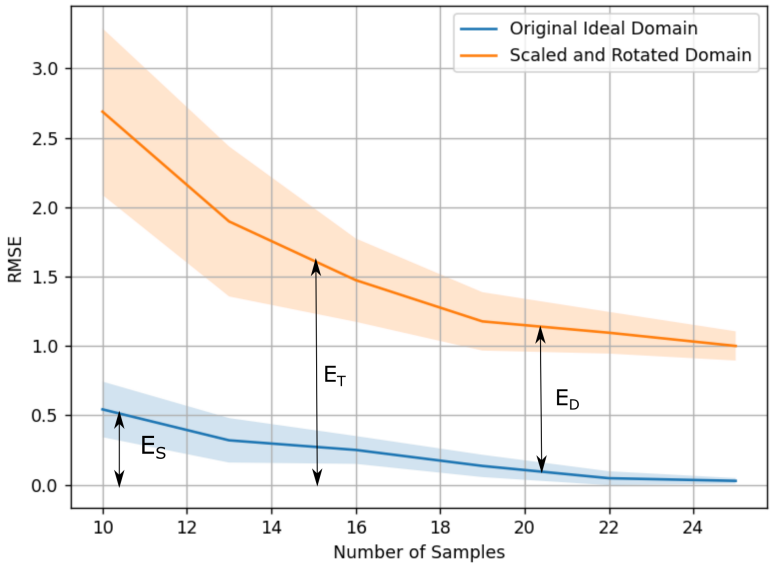

Figure 6: The sources of poor performance of a surrogate. The total error E_T consists of the sparsity of information error E_S and the construction coordinate system error E_D.

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\intercal \boldsymbol{A}\boldsymbol{x}, \tag{13}$$

where $\boldsymbol{A}$ is a 2×2 matrix that is also the Hessian of the function. Consider the three different cases

$$\boldsymbol{A}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \boldsymbol{A}_2 = \begin{bmatrix} 3 & 1 \\ 1 & 1 \end{bmatrix}, \quad \boldsymbol{A}_3 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}. \tag{14}$$

Figure 7 depicts the contour plots for these three cases. The dashed lines in these figures indicate the eigenvalues and eigenvectors of the $\boldsymbol{A}$ matrices.

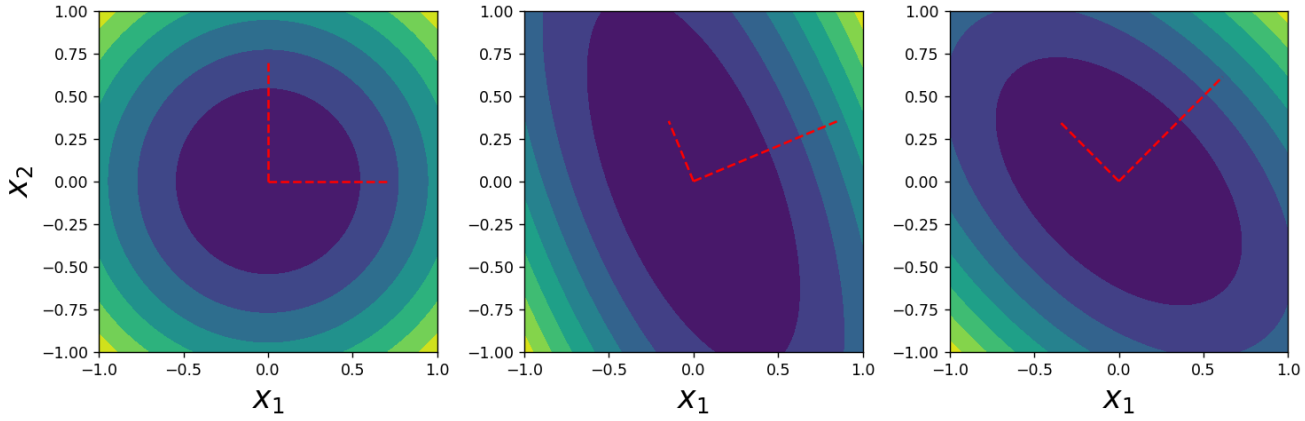

Figure 7: Contour plots of quadratic functions with Hessians given by $\boldsymbol{A}_1$, $\boldsymbol{A}_2$ and $\boldsymbol{A}_3$. The dashed lines indicate the eigenvectors and their lengths are chosen proportional to the eigenvalues.

The shape of the function in the case of $\boldsymbol{A}_1$, when the function is isotropic, closely resembles the shape of the basis functions. Therefore, the goal of the transformation scheme should be to create a coordinate system where for any $\boldsymbol{A}$, the function evaluated in the transformed coordinate system should resemble the case where $\boldsymbol{A} = \boldsymbol{A}_1$.

This can be achieved by utilising the eigenvalues and eigenvectors of the Hessian matrix. The coordinate system is then rotated using the eigenvectors of the Hessian and scaled by the square root of the eigenvalues for each direction. Figure 8 shows the contours using this transformation scheme, for the 3 different $\boldsymbol{A}$ matrices. Clearly,

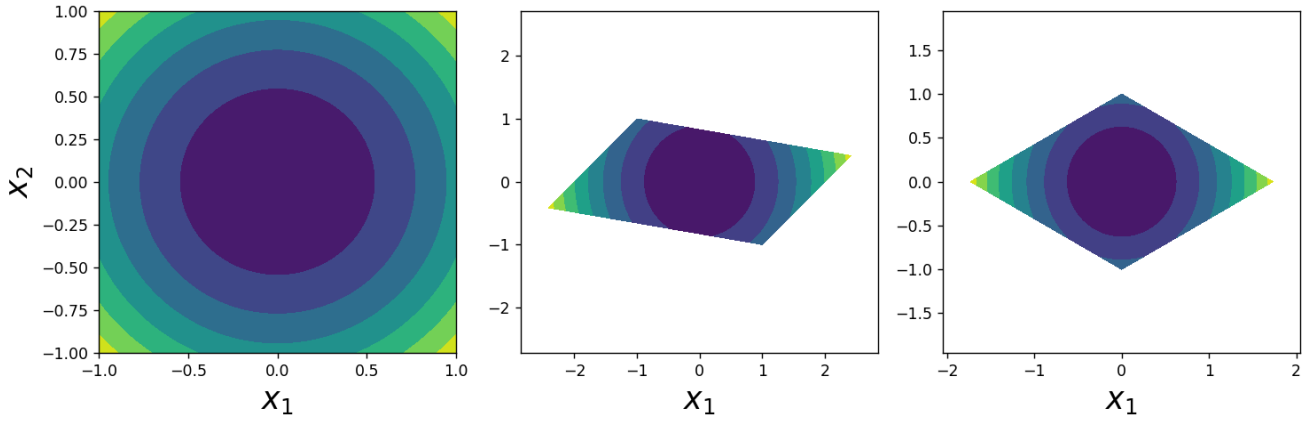

Figure 8: Contour plots of quadratic functions with Hessians given by $\boldsymbol{A}_1$, $\boldsymbol{A}_2$ and $\boldsymbol{A}_3$, after applying the proposed transformation scheme to the original coordinate system.

by taking into account the curvature in all directions the problem can be recast into a coordinate system where the function is isotropic.

4.2 General Higher-Order Non-linearity

The challenge is to generalise this procedure such that it can be implemented on any non-linear function (i.e. any problem with an unknown Hessian). Initially, it seems reasonable to take some global curvature measure, since the surrogate is fit on the entire coordinate system. This is however not the case. The theoretical motivation behind the transformation scheme developed in this section is as follows: RBF surrogate models are constructed as a summation

of isotropic basis functions placed throughout the design domain. Therefore, if *locally* the underlying function is anisotropic (meaning anisotropic at the location of the basis function), the basis function will not offer a reliable estimation of the local behaviour. As the model is a summation of these now unreliable basis functions, the overall predictive ability of the model suffers. As such, by making use of local estimations of curvature it becomes possible to predict the suitability of using isotropic basis functions to predict the underlying function's behaviour. The eigenvectors and eigenvalues of these local estimations of curvature then also inform what the optimum coordinate system is for each local basis function. Therefore, it is possible to approximate a single *global* coordinate system as an average of all the optimum *local* coordinate systems.

This theoretical framework will be motivated with two example numerical problems. Consider the 1D function

$$F(x) = \sin(f\pi x) + 15(x - 0.5)^2, \tag{15}$$

where f is 6 and 12 for the two example problems depicted in Figure 9. This can be thought of the behaviour of a high dimensional function in two eigen directions. These functions are each sampled 50 times. Since the optimum local length scale in the two directions differ, this functions is clearly anisotropic. If we scale the functions such that the global curvature estimates become similar, no scaling will be required as the global estimates both return similar curvature estimations. However, if we scale the functions such that the local curvature estimates become similar, then the function becomes more isotropic and should be approximated better using the summation of isotropic basis functions.

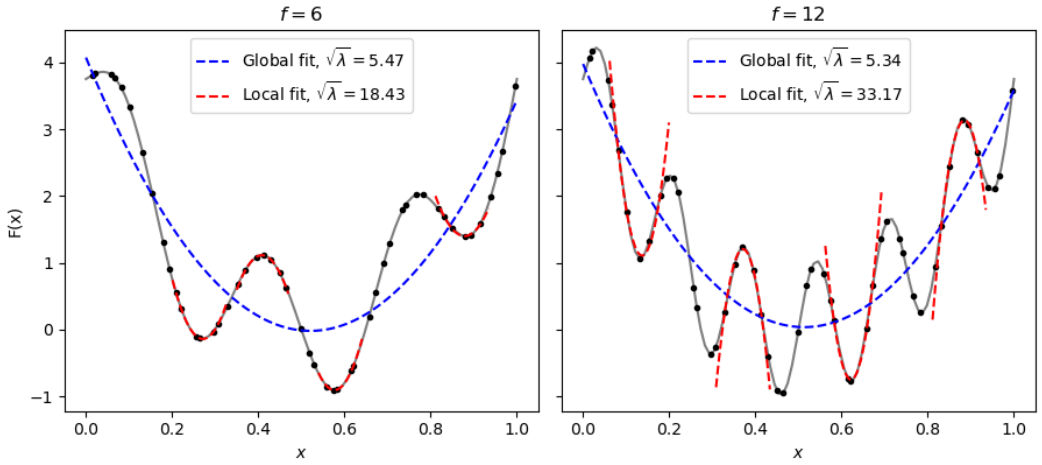

Figure 9: Two 1D example problems illustrating the difference between a single global fit and many local fits. The average optimum scaling factor proposed by both estimations, $\sqrt{\lambda}$, for the local and global fits are also denoted on the figure.

The shortcomings of a single global approximation method is also demonstrated on the 2D problem from Equation (7), in the scaled and rotated coordinate system using 25 samples. The function is approximated using a full $n-$dimensional quadratic fit of the form

$$\boldsymbol{f} = \sum_{i}^{n}\sum_{j}^{n} w_{ij}x_i x_j + \sum_{k}^{n} w_k x_k + w_c, \tag{16}$$

where the weights w_{ij}, w_k, and w_c are associated with the quadratic and coupling terms, the linear terms, and the constant term in the equation respectively. For this example problem the case where $n = 2$ is used. The w_{ij} weights solved from this fitted function can then be re-arranged into the Hessian of the quadratic fit

$$\boldsymbol{H} = \begin{bmatrix} 2w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & 2w_{22} & \dots & w_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & \dots & 2w_{nn} \end{bmatrix}, \tag{17}$$

where $w_{12} = w_{21}$ as the matrix is symmetric. In the implementation of this paper, an interpolating fit is constructed. This requires as many function values as there are unknown coefficients in the fit. These points are selected as the closest points surrounding the point at which the Hessian is approximated, resulting in a *local* approximation of the Hessian.

Figure 10 depicts the contour plots of the scaled and rotated function, a global full quadratic fit and four local full quadratic fits. In this example local fits at four random sampled points are constructed. The five nearest neighbours are used to construct the fit.

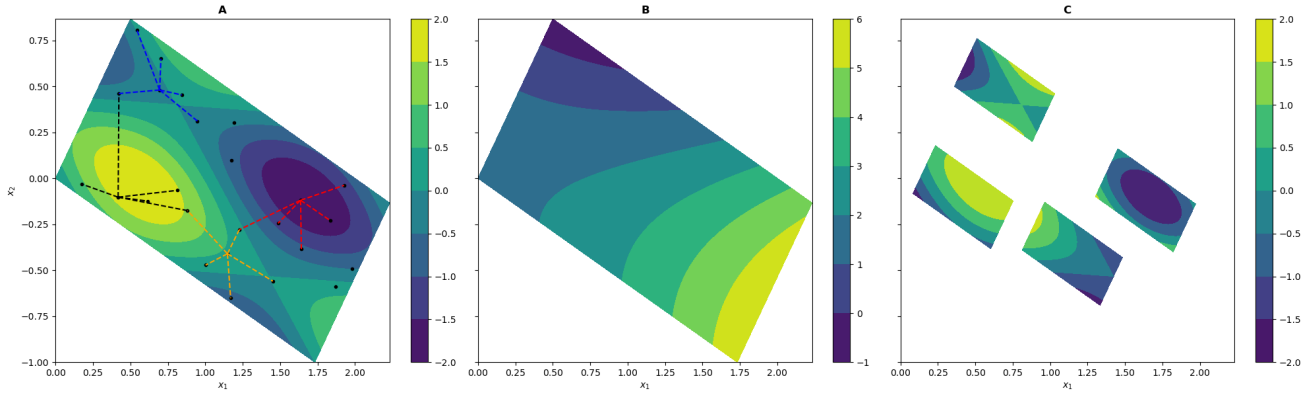

Figure 10: Contour plots of the original function, the global quadratic fit and local fits in subplots $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ respectively. The local cluster of points used for each local fit are shown by the coloured dashed lines.

Note that the global quadratic fit offers very little resemblance to the curvature of the underlying function. This occurs as the quadratic assumption cannot capture the full non-linearity of the underlying function across the entire coordinate system. The regressed quadratic fit instead offers a poor representation of the underlying curvature as it completes a global least squares fit using function information. Although the regressed quadratic fit has a low function value error, as this is the information it is constructed with, it offers a poor representation of the curvature of the underlying function.

The local fits on the other hand clearly offer a better representation of the curvature present in the problem. To remove variance in the local information some average measure of the local estimations must be found. Obtaining an average orthogonal matrix from all the local eigenvectors is not a trivial computation and averages of orthogonal matrices are not themselves orthogonal [32]. Therefore one average *global* Hessian is created from the many *local* Hessians. This is done by using the decomposition used in the Saddle-Free Newton method [33]

$$\boldsymbol{H} = \boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{V}^T, \tag{18}$$

where $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ are the eigenvectors and a diagonal matrix containing the eigenvalues along the diagonal, respectively. Each local Hessian is then recreated by taking the absolute value of the eigenvalue matrix,

$$\boldsymbol{H}_{\text{rec}} = \boldsymbol{V}|\boldsymbol{\Sigma}|\boldsymbol{V}^T. \tag{19}$$

The average *global* Hessian matrix is then calculated by taking the average of these reconstructed *local* Hessians:

$$\boldsymbol{H}_{\text{avg}} = \frac{1}{N}\sum_{i}^{N} \boldsymbol{H}_{\text{rec}}. \tag{20}$$

The Hessian is reconstructed with the absolute values of the eigenvalues as the local Hessians can be either concave or convex, as can be seen in Figure 9. The average of a collection of concave and convex Hessians can be a zero matrix as the positive and negative curvatures may be equal. Therefore, all the local Hessians are reconstructed to be convex (positive eigenvalues), keeping only the *magnitude* and *direction* of the local curvature.

Next the eigenvalues and eigenvectors of this average global Hessian are computed. The coordinate system is rotated using the eigenvectors as columns in an orthogonal matrix, and each direction is scaled with the square root of the eigenvalues.

4.3 Hessian Estimation

For the research completed in this paper two methods are selected to estimate Hessian information depending on the information available. When only function information is available the quadratic fits used in Section 4.2 are implemented. In the case where gradient information is available, the Symmetric Rank 1 (SR1) Hessian update method [22] is used:

$$\boldsymbol{H}_{k+1} = \boldsymbol{H}_k + \frac{(\boldsymbol{y}_k - \boldsymbol{H}_k \Delta \boldsymbol{x}_k)(\boldsymbol{y}_k - \boldsymbol{H}_k \Delta \boldsymbol{x}_k)^T}{(\boldsymbol{y}_k - \boldsymbol{H}_k \Delta \boldsymbol{x}_k)^T \Delta \boldsymbol{x}_k}. \tag{21}$$

The initial Hessian estimate $\boldsymbol{H}_0$ is an identity matrix and the term $\boldsymbol{y}_k$ is defined as

$$\boldsymbol{y}_k = \nabla F(\boldsymbol{x}_k + \Delta \boldsymbol{x}_k) - \nabla F(\boldsymbol{x}_k). \tag{22}$$

To ensure that the *local* Hessian approximation is rank sufficient, n SR1 updates are performed at the n closest points surrounding the point where the Hessian is estimated. This of course requires the gradient vector at each of these n points. The two methods are referred to as gradient enhanced local Hessian method (GE-LHM) and function value local Hessian method (FV-LHM).

A key difference between these two Hessian estimation methods is the minimum number of points each method requires in order to provide an estimation of the local Hessian. The SR1 method requires $n+1$ points (the centre point and the closest n points) while the quadratic fit requires a local cluster containing $n(n-1)/2+n+1$ points in n-dimensional space. This implies that when gradient information is available, the proposed transformation scheme scales favourably with problem dimension (linear scaling), while the function value-based Hessian approximation method becomes prohibitively expensive (quadratic scaling).

4.4 Numerical Transformation Example

To demonstrate the proposed method Figure 11 shows contour plots of Equation (7) in the scaled and rotated coordinate system, in the GE-LHM transformed coordinate system, the FV-LHM transformed coordinate system, and ASM transformed coordinate system for increasing sample numbers.

Ideally, as the sample number increases the methods should converge towards the original coordinate system (depicted in Figure 4). Figure 11 demonstrates that for this example problem the GE-LHM quickly converges to the original ideal coordinate system, while the FV-LHM and ASM need additional samples.

4.5 Effect of Transformation on the Gradient Vector

When the coordinate system is transformed, the gradients are indirectly also transformed. Therefore the gradients need to be transformed into the new coordinate system before they are used in the construction of the surrogate model. This is done by first expressing the underlying function as a function of the transformed coordinate system

$$\boldsymbol{F}(\hat{\boldsymbol{x}}) = \boldsymbol{F}(\hat{\boldsymbol{x}}(\boldsymbol{x})), \tag{23}$$

where now the original coordinate system $\boldsymbol{x}$ is assumed to be coupled and anisotropic, and the transformed coordinate system $\hat{\boldsymbol{x}}$ to be the ideal uncoupled and isotropic coordinate system.

Using the chain rule and Equation (23), the function gradient can be expressed as

$$\frac{d\boldsymbol{F}}{d\boldsymbol{x}} = \frac{d\boldsymbol{F}}{d\hat{\boldsymbol{x}}} \frac{d\hat{\boldsymbol{x}}}{d\boldsymbol{x}}, \tag{24}$$

where $\frac{d\boldsymbol{F}}{d\boldsymbol{x}}$ is the gradients that were found when the underlying function was sampled and $\frac{d\boldsymbol{F}}{d\hat{\boldsymbol{x}}}$ is the gradients in the new transformed coordinate system. Therefore the new gradient vector can be found by solving

$$\frac{d\boldsymbol{F}}{d\hat{\boldsymbol{x}}} = \frac{d\boldsymbol{F}}{d\boldsymbol{x}} \left(\frac{d\hat{\boldsymbol{x}}}{d\boldsymbol{x}} \right)^{-1}, \tag{25}$$

The required term $(\frac{d\hat{\boldsymbol{x}}}{d\boldsymbol{x}})^{-1}$ follows from

$$\hat{\boldsymbol{x}} = \boldsymbol{R}\boldsymbol{S}\boldsymbol{x}. \tag{26}$$

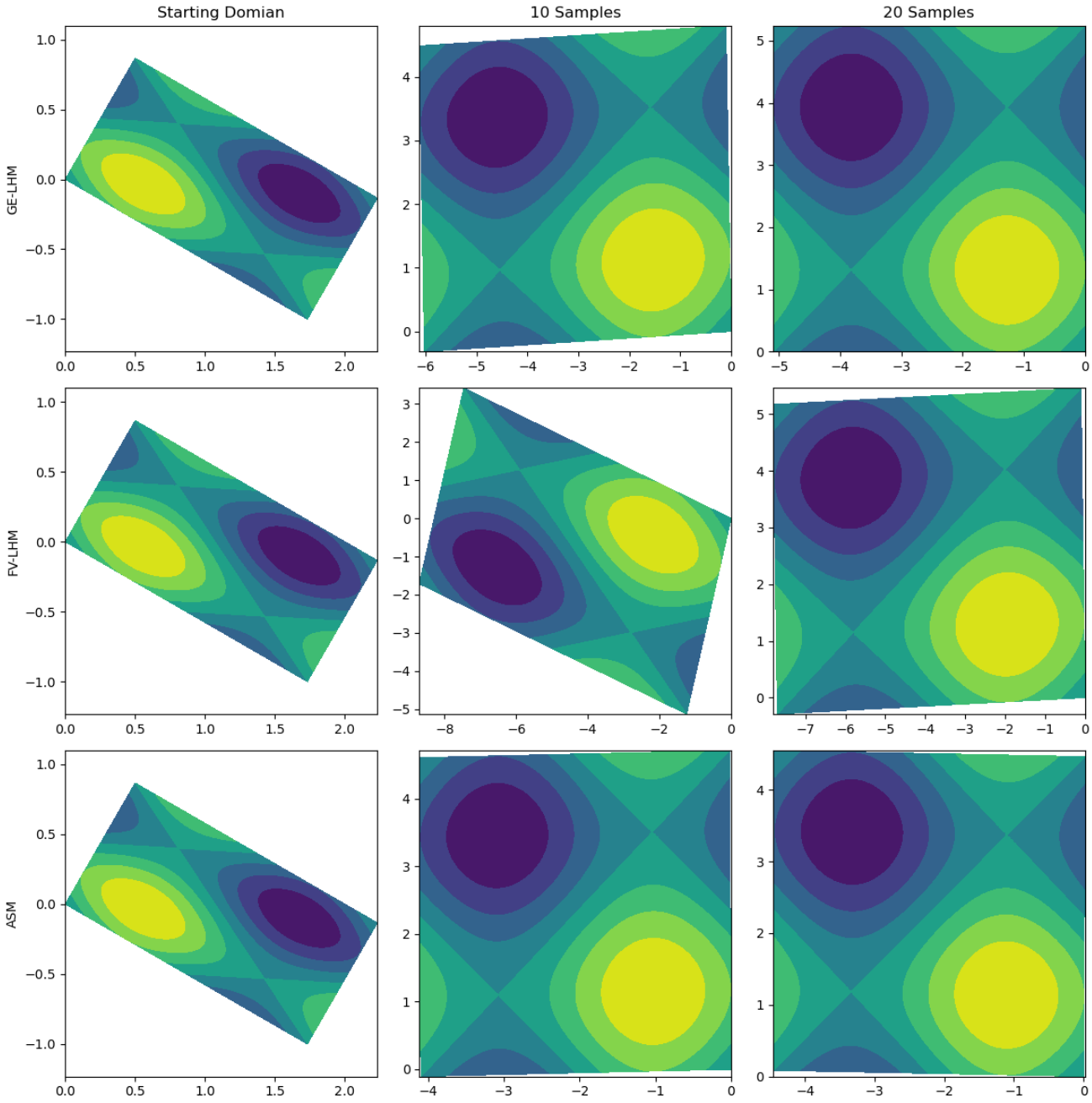

Figure 11: Contour plots of Equation (7) in the scaled and rotated coordinate systems, a transformed coordinate system using 10 samples, and a transformed coordinate system using 20 samples.

Taking the gradient of Equation (26) yields

$$\frac{d\hat{\boldsymbol{x}}}{d\boldsymbol{x}} = \boldsymbol{R}\boldsymbol{S}. \tag{27}$$

Since $\boldsymbol{R}$ is a orthogonal matrix, $\boldsymbol{R}^{-1} = \boldsymbol{R}^{\intercal}$. Therefore,

$$\left(\frac{d\hat{\boldsymbol{x}}}{d\boldsymbol{x}}\right)^{-1} = (\boldsymbol{R}\boldsymbol{S})^{-1} = \boldsymbol{S}^{-1}\boldsymbol{R}^{-1} = \boldsymbol{S}^{-1}\boldsymbol{R}^{\intercal}. \tag{28}$$

Since the scaling matrix $\boldsymbol{S}$ is a diagonal matrix, its inverse is simply the inverse of each diagonal entry placed in

the same location on the diagonal. The final transformed gradient from Equation (25) then becomes

$$\frac{d\boldsymbol{F}}{d\hat{\boldsymbol{x}}} = \frac{d\boldsymbol{F}}{d\boldsymbol{x}}\boldsymbol{S}^{-1}\boldsymbol{R}^{\intercal}. \tag{29}$$

4.6 Summary of Proposed Transformation Procedure

The implementation of the proposed transformation procedure can be separated into three procedures. The first procedure, procedure 1, iterates through all the sampled points and calculates an average Hessian estimation. Procedures 2 and 3 compute local Hessian estimations from some subset of points in the sample set.

Procedure 1: Transformation Procedure

Input : Sampled Information of the Underlying Function.
Output: The transformed coordinate system $\hat{\boldsymbol{x}}$ and the gradients $\frac{d\boldsymbol{F}}{d\hat{\boldsymbol{x}}}$

for *All sampled points* **do**
  if *Gradient Information is available* **then**
    Use Procedure 2
  else
    Use Procedure 3
  end
end
for *All Hessian Estimations* **do**
  Compute $\boldsymbol{H}_{\text{rec}} = \boldsymbol{V}|\boldsymbol{\Sigma}|\boldsymbol{V}^T$
end
Compute $\boldsymbol{H}_{\text{avg}} = \frac{1}{N}\sum_i^N \boldsymbol{H}_{\text{rec}}$;
Find the eigenvalues and eigenvectors of the average Hessian;
if *Gradient Information is available* **then**
  Compute $\frac{d\boldsymbol{F}}{d\hat{\boldsymbol{x}}} = \frac{d\boldsymbol{F}}{d\boldsymbol{x}}\boldsymbol{S}^{-1}\boldsymbol{R}^{\intercal}$ using the eigenvalues and eigenvectors;
end
Compute $\hat{\boldsymbol{x}} = \boldsymbol{RSx}$ using the eigenvalues and eigenvectors;
Return $\hat{\boldsymbol{x}}$, and $\frac{d\boldsymbol{F}}{d\hat{\boldsymbol{x}}}$

Procedure 2: Gradient Information Based Hessian Estimation

Input : A sampled point
Output: A Local Hessian Estimation

Find the $n+1$ closest points;
Initialise $\boldsymbol{H_0}$ as an Identity matrix ;
Arrange from furthest to closest;
for *Closest Points Subset* **do**
  Compute $\boldsymbol{H}_{k+1} = \boldsymbol{H}_k + \frac{(\boldsymbol{y}_k - \boldsymbol{H}_k\Delta\boldsymbol{x}_k)(\boldsymbol{y}_k - \boldsymbol{H}_k\Delta\boldsymbol{x}_k)^T}{(\boldsymbol{y}_k - \boldsymbol{H}_k\Delta\boldsymbol{x}_k)^T\Delta\boldsymbol{x}_k}$
end
Return The local Hessian Estimation.

5 Test Problem

In order to further evaluate i) the benefit of adequate coordinate system transformation, and ii) the proposed transformation scheme, an n-dimensional test problem is constructed. This test problem is created by adapting the numerical problem used in Section 3 to a more general form where the problem dimension can be altered.

Procedure 3: Function Information Based Hessian Estimation

Input : A sampled point

Output: A Local Hessian Estimation

1 Find the $n(n-1)/2+n+1$ closest points;

2 Fit local Quadratic function using $\boldsymbol{f} = \sum_i^n \sum_j^n w_{ij} x_i x_j + \sum_k^n w_k x_k + w_c$;

3 Rearrange weight vector into the Hessian ;

4 **Return** The local Hessian Estimation.

The test problem will then be used to investigate the benefit of appropriate coordinate system transformation as a function of problem dimension. If we select the test function as a decomposable function

$$f(\boldsymbol{x}) = f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n), \tag{30}$$

then the resulting Hessian will be a diagonal matrix. Then independent scaling along each coordinate axis might create an isotropic or near-isotropic function. Therefore we deliberately select our test function as a decomposable function, ensuring that we know the optimal reference frame in which to express the function. The remaining feature that we deliberately embed into the test function, is varying length scales in different coordinate directions. This results in a test function for which we can easily alter certain characteristics, such as problem dimension and complexity. The fact that key characteristics of the function can be easily altered allows for an independent study of desired characteristics without the need to create a new test function entirely. The test function is chosen to have the form

$$f(\boldsymbol{x}) = \frac{1}{n} \sum_{i=1}^{N} A_i \sin(F_i x_i), \tag{31}$$

where n is the problem dimension and F_i and A_i are the frequency and amplitude in the i^{th} coordinate direction. The amplitudes and frequencies are found from

$$A_i = -2 \exp \frac{-(2i-N)^2}{N} + 3, \tag{32}$$

$$F_i = \frac{3\pi}{2 + 2\exp \frac{-20i+N}{2}} + \frac{\pi}{2}. \tag{33}$$

These frequency and amplitude equations attempt to keep the complexity of the function relatively constant as the problem dimension increases. The frequency is bound between $[0.5\pi; 2\pi]$ and the amplitude between $[1, 3]$.

Another feature that is easily added to the test function, is to rotate the problem into an arbitrary reference frame. As the original test function exhibits a diagonal Hessian, a rotation of the design space is added to create a problem where the variables are coupled. This version of the test function will then assess how well the rotation aspect of the proposed transformation scheme works, i.e. if an uncoupled reference frame exists then the transformation scheme must be able to find it. The original coordinate system is rotated using a random rotation matrix $\boldsymbol{R}$ created from

$$\boldsymbol{R} = \text{expm}(\pi(\boldsymbol{A} - \boldsymbol{A}^\intercal)), \tag{34}$$

where $\boldsymbol{A}$ is a random matrix with elements sampled between $[-0.5, 0.5]$ and expm is the exponential map. The exponential map of a skew matrix $(\boldsymbol{A} - \boldsymbol{A}^\intercal)$ results in an orthogonal matrix [22].

In this research, the case where gradient information is available is also discussed. Therefore, the gradients of the n-dimensional test function are needed. The gradient of Equation (31) is simply

$$\frac{\partial F_i}{\partial x_i} = \frac{1}{n} A_i F_i \cos(F_i x_i), \tag{35}$$

where in the case of coordinate system rotation, the process detailed in Section 4.5 is followed.

6 RMSE Results

The numerical investigation in this section follow a two-step process

1. a coordinate system transformation,
2. followed by surrogate construction.

The results, therefore, attempt to separate the contribution of these two steps to the performance of a surrogate. Specifically, the information used to perform coordinate system transformation is deliberately separated from the information used to construct the surrogate.

This is done by constructing the FV-RBF and GE-RBF surrogate models, in six different coordinate systems. These six coordinate system transformation strategies are

- The gradient informed local Hessian estimation method (GE-LHM): Coordinate system transformation (rotation and scaling) performed by estimating the Hessian using gradient information,
- The function informed local Hessian estimation method (FV-LHM): Coordinate system transformation (rotation and scaling) performed by estimating the Hessian using function information,
- The Kriging hyper-parameter optimisation method: only coordinate system scaling (no rotation) is used, as discussed in Section 2.3,
- The Min-Max scaled method: The coordinate system is scaled (no rotation) to $[0; 1]$ in all dimensions,
- The Active Subspace Method: the implemented version selects all the eigenvectors, and
- The ideal transformation method: This transformation is only possible since we have an analytical expression for the underlying function, where the optimal rotation matrix $\boldsymbol{R}$ and scaling factors are known.

By using two different models in six different coordinate systems, the results will demonstrate if the construction *coordinate system* consistently impacts the performance of the surrogate model, regardless of the information used in the construction of the model. The two surrogate models, function value and gradient enhanced, are chosen to have the same model flexibility, i.e. the same number of centres, to further isolate the effect the construction coordinate system has on the performance of the surrogate model. By fixing the flexibility of the surrogate model it will be shown that the ill-suitably of the coordinate system the model is constructed in, and not a lack of construction information, is the main source of the approximation error.

The RMSE of the surrogates is found by sampling the error at 10^5 test points. Such a large number of test points is selected to ensure that an accurate RMSE is computed even for the high-dimensional versions of the test problem. This process is then repeated 50 times to be able to compute the average RMSE error, as well as the variance in the RMSE. Figure 12 presents the results for the 2-dimensional test problem. The average RMSE (solid lines) and the variance in RMSE (shaded areas) are shown for the function value and gradient enhanced RBF models in all six construction coordinate systems. The RMSE results are presented in the log coordinate system so that the performance of the models can be compared across a wide range of accuracy levels.

This 2D example shows that there is a benefit in constructing the surrogate in the transformed coordinate system instead of the $[0; 1]$ scaled coordinate system. For example, if the goal accuracy of the problem was 10^{-1} the GE-LHM coordinate system would require on average almost 50% less samples, from 20 to 11 samples, than the standard Min-Max coordinate system.

It is also noticeable that only scaling the coordinate system, i.e. the Kriging scaled results, is not nearly as beneficial as complete coordinate system transformation (scaling and rotation) that is achieved by ASM and gradient informed LHM at lower sample densities. Once sufficient samples are used, the function informed LHM begins to rapidly approach the performance of the gradient based methods.

The anisotropic nature of the problem is quickly overcome by sampling the coordinate system densely enough, but, as will be shown, overcoming the coupled and anisotropic nature of the function with dense enough sampling becomes far more difficult in higher dimensional problems. Figures 13 and 14 present the results for the 4 and 8-dimensional problems respectively.

This increase in problem dimension highlights both the importance of a complete transformation scheme as well as the benefit of gradient information. Firstly, for the 4-dimensional problem, there is some benefit of the Kriging-based scheme over the proposed function transformation and the simple Min-Max scaling. But, as the problem dimension increases to 8, this benefit diminishes to almost zero in the case of FV-RBF models. The second observation to note is the clear performance gain when a suitable completely transformed construction coordinate system is used. This gain is once again evident in the ideal transformation, gradient informed LHM, or

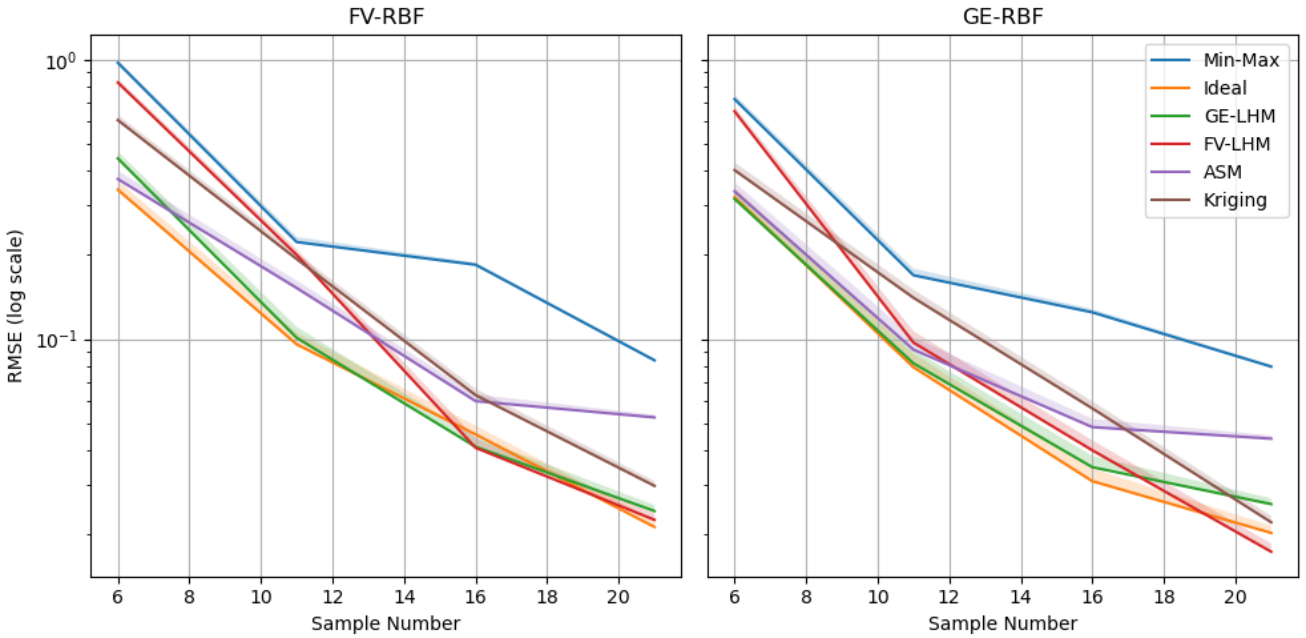

Figure 12: RMSE results for the FV-RBF (left) and the GE-RBF (right) on the 2-dimensional test problem.

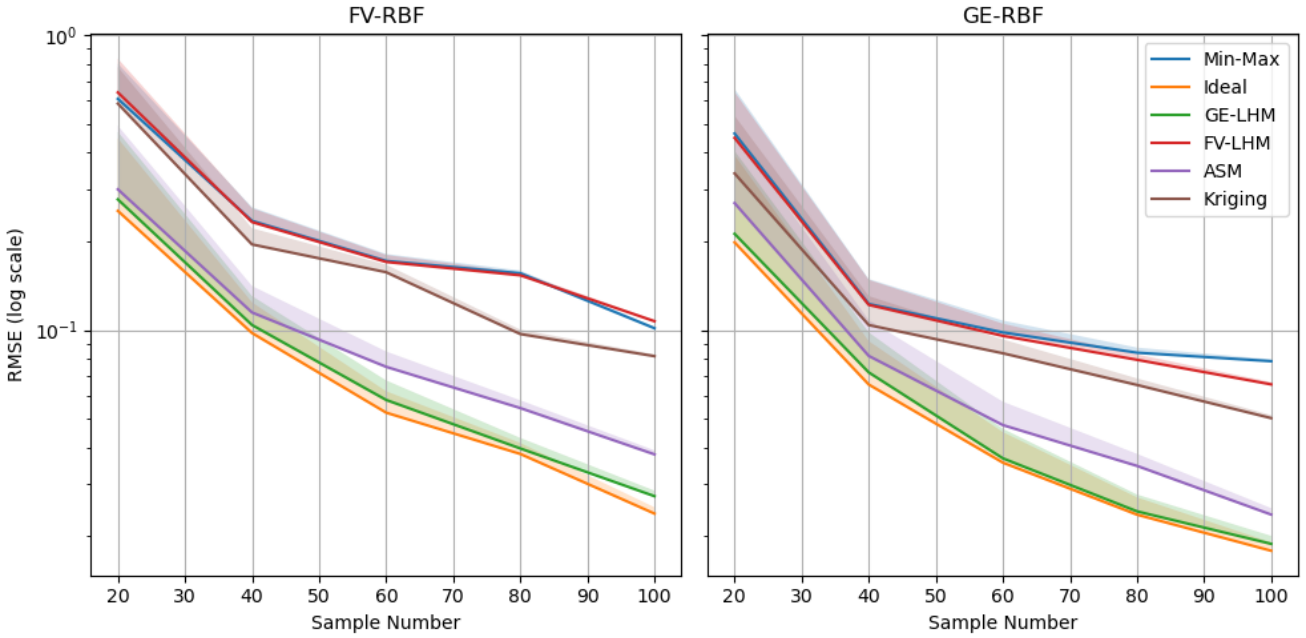

Figure 13: Log RMSE results for the FV-RBF (left) and the GE-RBF (right) on the 4-dimensional test problem.

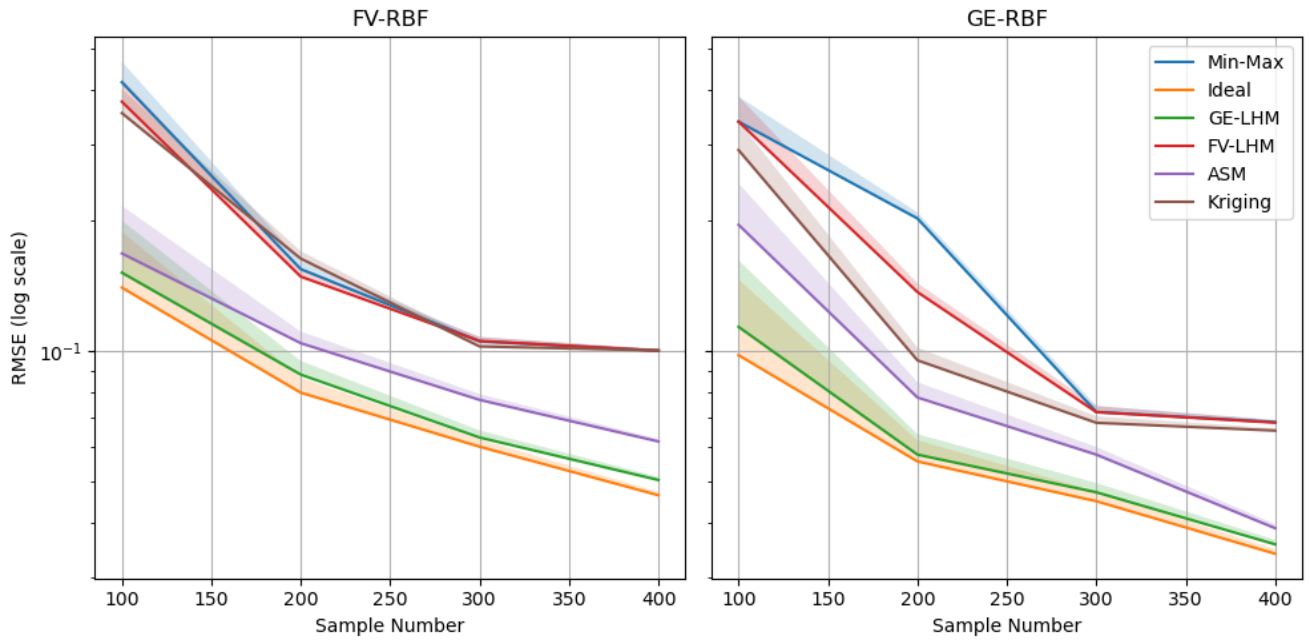

Figure 14: Log RMSE results for the FV-RBF (left) and the GE-RBF (Right) on the 8-dimensional test problem.

ASM. Gradient information offers a better approximation of local curvature, and therefore, returns a near-optimal approximation of the ideal transformed construction coordinate system.

If again the goal accuracy of the models were a RMSE of 10^{-1} the GE-LHM coordinate system requires on average 50% less samples in the 4-dimensional problem. For the FV-RBF models the samples decrease from 100 to 40, and for the GE-RBF models the samples decrease from 60 to 32. As the problem dimension is increased to 8, the benefit of appropriate transformation also increases. For a goal of RMSE of 10^{-1} the FV-RBF and GE-RBF models required almost 60% less samples, from 400 to 160 and 260 to 120 samples for the FV-RBF and GE-RBF models respectively. Therefore, the results in Figures 13 and 14 show that the benefit of coordinate system transformation increases as the problem dimension increases.

The figures also demonstrate that the FV-RBF in the GE-LHM or ASM coordinate systems have better performance than the GE-RBF models in the Min-Max coordinate system. This means that utilising the gradient information to perform a coordinate system transformation, the ASM and GE-LHM transformed coordinate systems, is more beneficial to surrogate performance than utilising the gradient information for the construction of the model. This is because the estimation of a suitable coordinate system is a far more information dense task, that grows with the dimensionality of the problem, than estimating a single scalar value from data. Therefore, the fact that the amount of information in gradient vectors grows with the dimensionality of problem means that gradient vectors are far more efficient at estimating an appropriate coordinate system than function values.

The problem dimension is then further increased to 16 and the same results are repeated in Figure 15.

From these results, it becomes apparent that the benefit of appropriate complete coordinate system transformation, over both Min-Max scaling or Kriging scaling, grows with problem dimension. As with the lower dimensional problems, the surrogates constructed in ill-suited coordinate systems offer minimal performance improvement in low sample density scenarios when additional samples are added. This slow rate of improvement for the "non-transformed" surrogate means that the proposed gradient-based LHM transformation scheme and the ideal transformation coordinate system require far less computational cost, i.e. fewer samples, to achieve the same accuracy. For example, if the 16-dimensional problem had a goal RMSE of 10^{-1} the proposed transformation scheme would require, on average, 800 and 650 samples for the function value and GE models respectively, while the standard Min-Max scaling would require 2000 and 1750 samples. Therefore, for this simple test function, the proposed transformation scheme results on average in 60% less computational cost compared to the standard Min-Max scaling procedure.

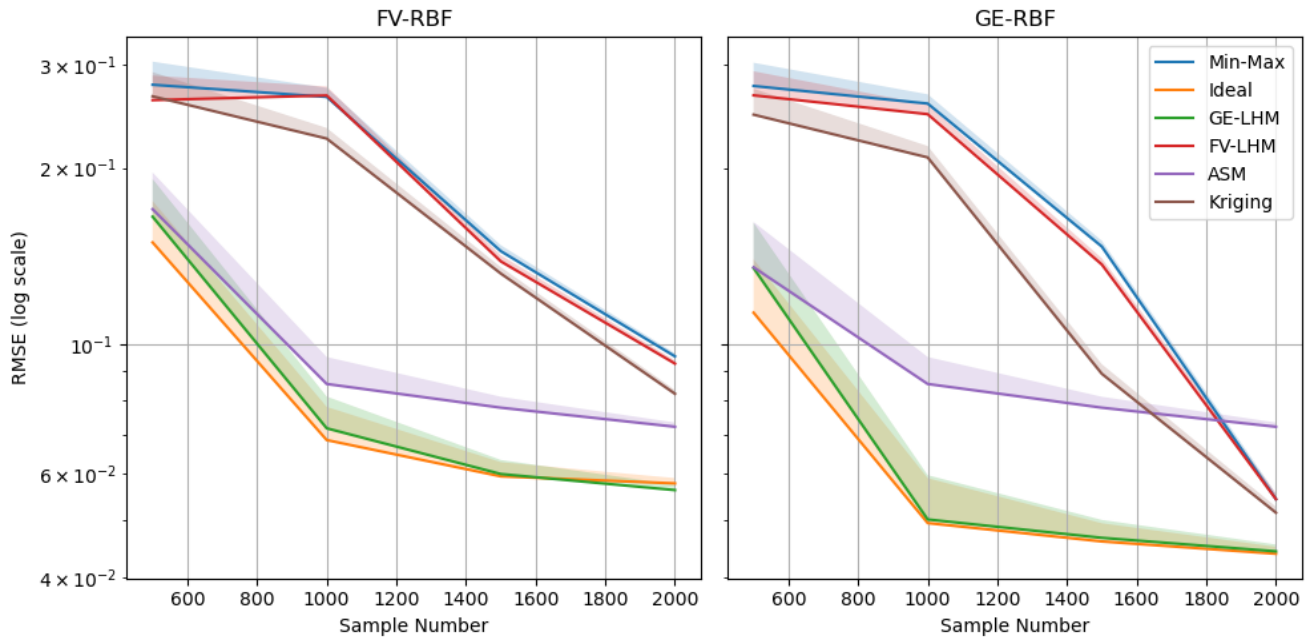

Figure 15: Log RMSE results for the FV-RBF (left) and the GE-RBF (Right) on the 16-dimensional test problem

What is noticeable in the results is that at higher dimensions the FV-LHM offers very little improvement over standard Min-Max scaling. This is due to the number of points needed to estimate curvature from function values growing exponentially as a function of the dimensionality. Therefore, at higher dimensions instead of estimating *local* curvature the FV-LHM instead begins to estimate *global* curvature.

An additional feature of the results in Figure 15 to note is the fact that the ASM performance is worse than simple Min-Max or Kriging based scaling for the GE-RBF models at high sampling densities in this numerical problem. This is most likely due to that fact that in this work the ASM is implemented as a coordinate system transformation scheme instead of, and as it is originally developed for, a coordinate system reduction technique.

7 Non-Decomposable Functions

The developed transformation technique assumes that the underlying function is decomposable, meaning there exists a scaling and rotation transformation that will recast the problem into a coordinate system where the variables in the transformed function are uncoupled. To investigate the performance of the method on a non-decomposable problem the well known $n-$dimensional Rosenbrock function in the domain $[-1, 1]^n \to \mathbb{R}$ is used. This problem is expressed by

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100 \cdot (x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right] \tag{36}$$

and the contours of the 2-dimensional version are shown in Figure 17.

The GE-RBF model is then constructed for various sample numbers for various problem dimensions, using the different transformation schemes. As this function is not decomposable there is not a clear or obvious optimum reference frame as there is with the crafted test problem. Therefore, there is no ideal transformation to compare to in this example.

The problem is completed for 2, 4, 8, and 16 dimensions at increasing sample numbers. As before, to account for the randomness in the sample locations the RBFs are constructed on 50 sets of sampled data and the mean RMSE error is recorded at 10000 randomly sampled points. The results are shown in Figure 17

Although there is some benefit in the proposed transformation scheme, there is very little difference between the results of all the transformation schemes. These results are expected, as all the pre-processing schemes are

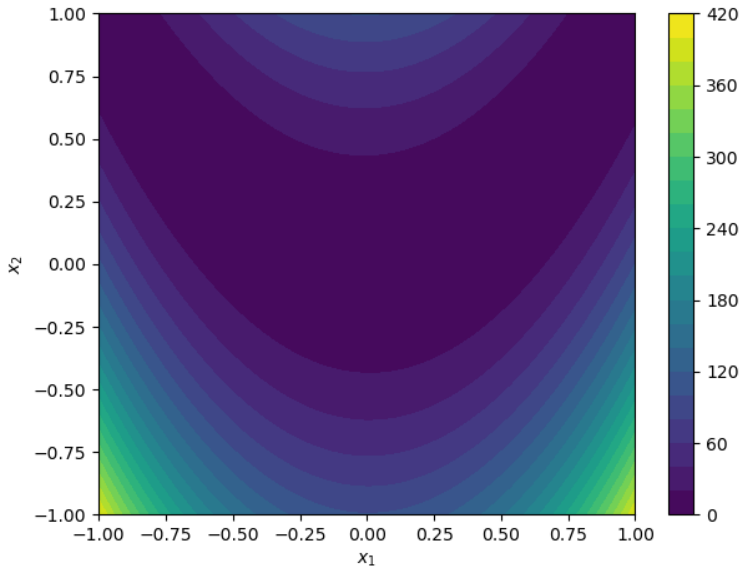

Figure 16: Contour plot of the 2 dimensional Rosenbrock function

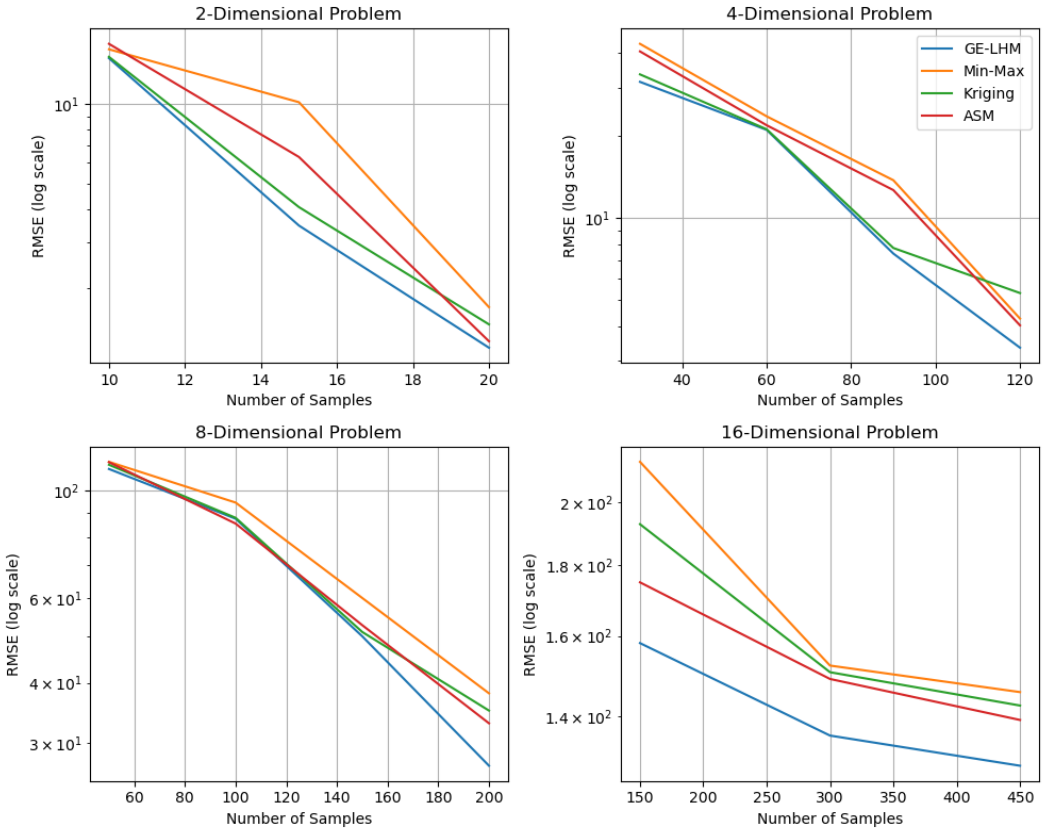

Figure 17: Results for the Rosenbrock function

designed for decomposable functions or functions that are already uncoupled. Therefore, in order to handle non-decomposable functions the pre-processing transformation will need be to non-linear and a function of the location

in the design domain.

What is clear from the results presented in this paper, is that a general non-linear transformation scheme will need to be based of local curvature information, and that this curvature information will ideally be estimated from sampled gradient information.

8 Conclusion

The work presented in this paper demonstrates that the coordinate system in which common radius basis function surrogate models are constructed in can have a significant influence on the predictive performance of the surrogate. This is done with a few main findings.

Firstly, the addition of gradient information into the construction of a surrogate model will not result in the expected improvement of the predictive performance of the surrogate if the coordinate system is not suitable. Therefore, attention needs to be given to a pre-processing step that will adequately transform the coordinate system in which the surrogate model will be constructed.

Secondly, a full coordinate system transformation, including both scaling and rotation, is required to address the isotropic assumption. Simple component-wise scaling is not a sufficient strategy.

The information needed to inform the pre-processing step is a collection of local curvature information rather than one global estimation of the curvature. This local curvature will need to be estimated in most practical engineering problems. Although this estimation can be completed with either gradient or function information, the results in this work demonstrate that gradient information offers a more efficient and accurate approximation of the local curvature. This is seen clearly at higher dimensions where to estimate local curvature from function values a large number of samples is required.

The coordinate system the models are constructed in impacts the performance of the surrogate model *regardless* of the information used to construct the model. There is improvement in both the FV and GE surrogate models when the coordinate system the models are constructed in is transformed using the developed coordinate system transformation scheme. The transformation must be a fully coupled rotation and scaling as only scaling the coordinate system is not sufficient.

The ASM method offers a noticeable performance improvement over standard Min-Max or Kriging based scaling. The proposed transformation scheme does outperform the ASM on this numerical problem, but it may come at a greater computational cost. The ASM completes one eigenvalue decomposition, on the approximated $\hat{\boldsymbol{C}}$ matrix, while LHM completes $p+1$ decompositions. The computational cost can be reduced by computing these $p+1$ decompositions in parallel, or by only using some subset of the p sampled points. If however the computational cost of evaluating the function value and gradient vector is high (as expected), the cost of the proposed transformation scheme is negligible in comparison.

Lastly, the use of gradient information allows for the estimation of local curvature to complete a powerful, automatic, and fully coupled coordinate system transformation scheme that results in near-optimal performance. Therefore, using the gradient information to transform the coordinate system can be far more beneficial to surrogate performance than including this information directly in the construction of the surrogate model.

9 Future Work

Although the proposed transformation scheme offers a significant improvement over the standard Min-Max scaling scheme on decomposable problems, there are two main scenarios that were not investigated:

- when the underlying function curvature varies greatly along a principal direction (commonly referred to as non-stationary problems), and
- when one dimension is sampled more densely than the other dimensions, such as with time series data.

These scenarios may require adaptation of the proposed transformation scheme, to achieve the same level of improvement as demonstrated in this paper.

Conflict of interest

The authors declare that they have no conflict of interest.

Replication of results

All necessary algorithms and problem parameters for possible replication of all result presented in this work have been detailed and referenced.

Funding Sources

There are no funding sources for this research.

A Basic Surrogate Models

Radial basis function surrogates refer to the family of surrogates that use a linear summation of basis functions that depend on a distance measure between two points. Popular options as basis functions include

- Inverse quadratic: $\phi(\boldsymbol{x}, \boldsymbol{c}, \epsilon) = \frac{1}{1+\epsilon||\boldsymbol{x}-\boldsymbol{c}||}$,
- Multi-quadratic: $\phi(\boldsymbol{x}, \boldsymbol{c}, \epsilon) = \frac{1}{\sqrt{||\boldsymbol{x}-\boldsymbol{c}||+\epsilon^2}}$,
- Gaussian: $\phi(\boldsymbol{x}, \boldsymbol{c}, \epsilon) = e^{-\epsilon||\boldsymbol{x}-\boldsymbol{c}||^2}$,

where the variable ϵ is referred to as the shape parameter and the point $\boldsymbol{c}$ is the centre of the basis function. The most widely used basis function is the Gaussian function. The RBF surrogate is expressed as a linear combination of k basis functions

$$f_{\text{RBF}} = \sum_{i=1}^{k} w_i \phi_i(\boldsymbol{x}, \boldsymbol{c}_i, \epsilon). \tag{37}$$

This equation becomes a system of equations

$$\boldsymbol{f} = \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{c}, \epsilon)\boldsymbol{w}, \tag{38}$$

where the variable $\boldsymbol{\Phi}$ is a $k \times p$ matrix where p is the number of samples. This matrix is then expressed as

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi(\boldsymbol{x}_1, \boldsymbol{c}_1, \epsilon) & \phi(\boldsymbol{x}_1, \boldsymbol{c}_2, \epsilon) & \dots & \phi(\boldsymbol{x}_1, \boldsymbol{c}_k, \epsilon) \\ \phi(\boldsymbol{x}_2, \boldsymbol{c}_1, \epsilon) & \phi(\boldsymbol{x}_2, \boldsymbol{c}_2, \epsilon) & \dots & \phi(\boldsymbol{x}_2, \boldsymbol{c}_k, \epsilon) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(\boldsymbol{x}_p, \boldsymbol{c}_1, \epsilon) & \phi(\boldsymbol{x}_p, \boldsymbol{c}_2, \epsilon) & \dots & \phi(\boldsymbol{x}_p, \boldsymbol{c}_k, \epsilon) \end{bmatrix}_{k \times p}. \tag{39}$$

The remaining parameters of the surrogate include the number and locations of the centres $\boldsymbol{c}$ and the value of the shape parameter ϵ.

A popular choice for the centres is to select $p = k$, meaning that the number of centres is equal to the number of sampled points and to position the centres at the location of the sampled points. For this choice the matrix $\boldsymbol{\Phi}$ becomes square and the weight vector can be solved directly from Equation (38). This is the method implemented for this research.

Some research, for example, [34] implement a fussy K-means clustering scheme to allocate the centres in the coordinate system. In this scenario the system becomes over-determined and the least squares solution

$$\boldsymbol{\Phi}^{\boldsymbol{T}}\boldsymbol{f} = \boldsymbol{\Phi}^{\boldsymbol{T}}\boldsymbol{\Phi}\boldsymbol{w}, \tag{40}$$

must be implemented.

A.1 GE-RBF Models

GE-RBF models directly include the gradients in the model construction. This can either be done in an interpolating sense, such that the model directly interpolates both the function and gradient information at every point in the design space, or in a regression sense, such that the model neither exactly fits the function or gradient information, but rather attempts to fit both in the least squares sense.

A regression-based model is typically preferred to a fully interpolating model for two main reasons. Firstly, computational simulations that require discretisation and iterative solvers can result in noisy solutions. Therefore, if the model fits the solutions exactly the model may fit more to the noise in the data than to the underlying function. Secondly, a full interpolation matrix in either higher dimensional or densely sampled problems may become prohibitively large to solve, while a regression-based model can still offer useful results at a more reasonable computational cost. Therefore, regression-based derivations are offered in this section for the discussed surrogate models.

Another reason that regression models are preferred in this research is that the goal of the numerical investigations is to isolate the effect that the coordinate system transformation has on the performance of the surrogate model. Therefore, the flexibility of the function and gradient-enhanced models are kept constant (by keeping the number and location of the centres the same), so that the only variable that is altered is the coordinate system transformation strategy. The effect of increased flexibility in gradient-enhanced models, and how this increased flexibility is achieved, are outside the scope of this research.

The construction of GE-RBF begin by firstly taking the gradient of the Gaussian basis function

$$\frac{d\phi(\boldsymbol{x},\boldsymbol{c},\epsilon)}{d\boldsymbol{x}} = -2\epsilon\phi(\boldsymbol{x},\boldsymbol{c},\epsilon)(\boldsymbol{x}-\boldsymbol{c}), \tag{41}$$

where Equation (41) returns a column vector.

A new system of equations can then be created from the gradient information at each sampled point for p samples for the RBF surrogate model

$$\begin{bmatrix} \frac{df_1}{d\boldsymbol{x}} \\ \frac{df_2}{d\boldsymbol{x}} \\ \vdots \\ \frac{df_p}{d\boldsymbol{x}} \end{bmatrix} = \begin{bmatrix} \frac{d\phi(\boldsymbol{x}_1,\boldsymbol{c}_1,\epsilon)}{d\boldsymbol{x}} & \frac{d\phi(\boldsymbol{x}_1,\boldsymbol{c}_2,\epsilon)}{d\boldsymbol{x}} & \cdots & \frac{d\phi(\boldsymbol{x}_1,\boldsymbol{c}_k,\epsilon)}{d\boldsymbol{x}} \\ \frac{d\phi(\boldsymbol{x}_2,\boldsymbol{c}_1,\epsilon)}{d\boldsymbol{x}} & \frac{d\phi(\boldsymbol{x}_2,\boldsymbol{c}_2,\epsilon)}{d\boldsymbol{x}} & \cdots & \frac{d\phi(\boldsymbol{x}_2,\boldsymbol{c}_k,\epsilon)}{d\boldsymbol{x}} \\ \vdots & \vdots & \vdots & \\ \frac{d\phi(\boldsymbol{x}_p,\boldsymbol{c}_1,\epsilon)}{d\boldsymbol{x}} & \frac{d\phi(\boldsymbol{x}_p,\boldsymbol{c}_2,\epsilon)}{d\boldsymbol{x}} & \cdots & \frac{d\phi(\boldsymbol{x}_p,\boldsymbol{c}_k,\epsilon)}{d\boldsymbol{x}} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix}. \tag{42}$$

The system of Equations (42), can then be written as

$$\nabla \boldsymbol{f} = \boldsymbol{\Phi}_{fo}\boldsymbol{w}_{fo}. \tag{43}$$

The subscript fo denotes that first-order information is used in the system. The gradient information can then be added to the original function-based system to create a new system of equations

$$\begin{bmatrix} \boldsymbol{f} \\ \nabla \boldsymbol{f} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{\Phi}_{fo} \end{bmatrix} \boldsymbol{w}_{GE}. \tag{44}$$

The weight vector now contains the subscript GE to show that the weights solved from this system are for the gradient-enhanced versions of the surrogate models.

An important characteristic to note of the GE models is the size of the systems that need to be solved. In the function-value based models p scalar samples are taken of the underlying function, creating a system of size $p \times k$, while in the GE models p scalars and p gradient vectors of size $n \times 1$ are sampled, creating a $(p + p \times n) \times k$ system. As the weight vector, $\boldsymbol{w}_{GE}$, is the same size, specifically $k \times 1$ in both the function value and GE models, the models are of equal flexibility. The difference between the function value and GE models is therefore that the GE models are constructed by regressing the model to the gradient information using the least squares formulation (similar to Equation (40)).

References

[1] F. A. Viana, C. Gogu, T. Goel, Surrogate modeling: tricks that endured the test of time and some recent developments, Structural and Multidisciplinary Optimization 64 (5) (2021) 2881–2908. doi:10.1007/s00158-021-03001-2.

[2] M. A. Bouhlel, J. R. Martins, Gradient-enhanced kriging for high-dimensional problems, Engineering with Computers 35 (1) (2019) 157–173. arXiv:1708.02663, doi:10.1007/s00366-018-0590-x.

[3] D. J. Toal, N. W. Bressloff, A. J. Keane, Kriging hyperparameter tuning strategies, AIAA Journal 46 (5) (2008) 1240–1252. doi:10.2514/1.34822.

[4] M. A. Bouhlel, N. Bartoli, A. Otsmane, J. Morlier, An Improved Approach for Estimating the Hyperparameters of the Kriging Model for High-Dimensional Problems through the Partial Least Squares Method, Mathematical Problems in Engineering 2016 (2016). doi:10.1155/2016/6723410.

[5] M. Urquhart, E. Ljungskog, S. Sebben, Surrogate-based optimisation using adaptively scaled radial basis functions, Applied Soft Computing Journal 88 (2020) 106050. doi:10.1016/j.asoc.2019.106050.
URL https://doi.org/10.1016/j.asoc.2019.106050

[6] D. R. Jones, A Taxonomy of Global Optimization Methods Based on Response Surfaces, Journal of Global Optimization 21 (4) (2001) 345–383. doi:10.1023/A:1012771025575.

[7] S. Koziel, D. E. Ciaurri, L. Leifsson, Surrogate-based methods, Computational optimization, methods and algorithms (2011) 33–59.

[8] S. Ulaganathan, I. Couckuyt, F. Ferranti, E. Laermans, T. Dhaene, Performance study of multi-fidelity gradient enhanced kriging, Structural and Multidisciplinary Optimization 51 (2015) 1017–1033. doi:10.1007/s00158-014-1192-x.

[9] I. C. Kampolis, E. I. Karangelos, K. C. Giannakoglou, Gradient-assisted radial basis function networks: Theory and applications, Applied Mathematical Modelling 28 (2) (2004) 197–209. doi:10.1016/j.apm.2003.08.002.

[10] L. Laurent, R. Le Riche, B. Soulier, P. A. Boucard, An Overview of Gradient-Enhanced Metamodels with Applications, Archives of Computational Methods in Engineering 26 (1) (2019) 61–106. doi:10.1007/s11831-017-9226-3.

[11] M. D. McKay, R. J. Beckman, W. J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 42 (1) (1979) 55–61. doi:10.1080/00401706.2000.10485979.

[12] C. D. Vu Khac Ky, Surrogate-based methods for black-box optimization, International Transactions in Operational Research 24 (3) (2019) 393–424.

[13] K. Cheng, Z. Lu, C. Ling, S. Zhou, Surrogate-assisted global sensitivity analysis: an overview, Structural and Multidisciplinary Optimization 61 (3) (2020) 1187–1213. doi:10.1007/s00158-019-02413-5.

[14] Y. S. Ryu, M. Haririan, C. C. Wu, J. S. Arora, Structural design sensitivity analysis of nonlinear response, Computers and Structures 21 (1-2) (1985) 245–255. doi:10.1016/0045-7949(85)90247-0.

[15] N. Olhoff, E. Lund, Finite Element Based Engineering Design Sensitivity Analysis and Optimization, Ph.D. thesis, Aalborg University (1995).

[16] T. Hisada, Recent Progress in Nonlinear FEM-Based Sensitivity Analysis, JSME International Journal 38 (3) (1995) 430 – 433.

[17] J. Parente, L. E. Vaz, On evaluation of shape sensitivities of non-linear critical loads, International Journal for Numerical Methods in Engineering 56 (6) (2003) 809–846. doi:10.1002/nme.587.

[18] G. Dhondt, K. Wittig, Calculix (1988).

[19] V. Komkov, K. K. Choi, E. J. Haug, F.-d. S. Systems, Design sensitivity analysis of structural systems, Mathematics in Science and Engineering 177 (C) (1986) 1–82. doi:10.1016/S0076-5392(09)60320-9.

[20] D. Balagangadhar, S. Roy, Design sensitivity analysis and optimization of steady fluid-thermal systems, Computer Methods in Applied Mechanics and Engineering 190 (42) (2001) 5465–5479. doi:10.1016/S0045-7825(01)00224-9.

[21] J. C. Newman, A. C. Taylor, R. W. Barnwell, P. A. Newman, G. J.-W. Hou, Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations, Journal of Aircraft 36 (1) (1999) 87–96. doi:10.2514/2.2416.
URL https://doi.org/10.2514/2.2416

[22] J. A. Snyman, D. N. Wilke, Practical Mathematical Optimization, 2nd Edition, Springer, 2005. doi:10.1007/b105200.

[23] J. Laurenceau, P. Sagaut, Building efficient response surfaces of aerodynamic functions with kriging and cokriging, AIAA Journal 46 (2) (2008) 498–507. doi:10.2514/1.32308.

[24] J. Laurenceau, M. Meaux, Comparison of gradient and response surface based optimization frameworks using adjoint method, in: 49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 16th AIAA/ASME/AHS Adaptive Structures Conference, 10th AIAA Non-Deterministic Approaches Conference, 9th AIAA Gossamer Spacecraft Forum, 4th AIAA Multidisciplinary Design Optimization Specialists Conference, 2008, p. 1889.

[25] J. R. Koehler, A. B. Owen, Handbook of Statistics, Elsevier Science, 1996.

[26] J. Snyman, D. Wilke, Practical Mathematical Optimization: Basic Optimization Theory and Gradient-Based Algorithms, Springer Optimization and Its Applications, Springer International Publishing, 2018.
URL https://books.google.co.za/books?id=n1dLswEACAAJ

[27] P. G. Constantine, E. Dow, Q. Wang, Active subspace methods in theory and practice: Applications to kriging surfaces, SIAM Journal on Scientific Computing 36 (2014) A1500–A1524. doi:10.1137/130916138.

[28] N. Namura, K. Shimoyama, S. Obayashi, Kriging surrogate model with coordinate transformation based on likelihood and gradient, Journal of Global Optimization 68 (2017) 827–849. doi:10.1007/s10898-017-0516-y.

[29] J. Li, J. Cai, K. Qu, Surrogate-based aerodynamic shape optimization with the active subspace method, Structural and Multidisciplinary Optimization 59 (2019) 403–419. doi:10.1007/s00158-018-2073-5.

[30] T. W. Lukaczyk, P. Constantine, F. Palacios, J. J. Alonso, Active subspaces for shape optimization, in: 10th AIAA multidisciplinary design optimization conference, 2014, p. 1171.

[31] Z. Li, J. Zhu, C. C. Foo, C. H. Yap, A robust dual-membrane dielectric elastomer actuator for large volume fluid pumping via snap-through, Applied Physics Letters 111 (21) (2017). doi:10.1063/1.5005982.
URL http://dx.doi.org/10.1063/1.5005982

[32] E. Liski, K. Nordhausen, H. Oja, A. Ruiz-Gazen, Averaging orthogonal projectors (2012). arXiv:1210.2575.

[33] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, Y. Bengio, Identifying and attacking the saddle point problem in high-dimensional non-convex optimization, Advances in Neural Information Processing Systems 4 (January) (2014) 2933–2941. arXiv:1406.2572.

[34] Y. Zhang, C. Gong, H. Fang, H. Su, C. Li, A. D. Ronch, An efficient space division–based width optimization method for rbf network using fuzzy clustering algorithms, Structural and Multidisciplinary Optimization 60 (2019) 461–480. doi:10.1007/s00158-019-02217-7.