UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

The Shape Optimisation of Compliant Structures to Produce a Desired Snap-Through Load-Displacement Path

by

Johann Mynhardt Bouwer

Submitted in application for
Doctorate of Engineering (Mechanical Engineering)

in the

Department of Mechanical and Aeronautical Engineering
Faculty of Engineering, Built Environment and Information Technology

Supervised by
Prof. Schalk Kok and Prof. Daniel N. Wilke

UNIVERSITY OF PRETORIA

2023

SUMMARY

THE SHAPE OPTIMISATION OF COMPLIANT STRUCTURES TO PRODUCE A DESIRED SNAP-THROUGH LOAD-DISPLACEMENT PATH

by

Johann Mynhardt Bouwer

Keywords: Shape Optimisation, Analytical Sensitivity Analysis, Gradient-Only Optimisation, Surrogates Models, Radial Basis Functions, Coordinate System Transformation, Spatio-Temporal Models, Snap-Through.

The research problem completed in this thesis is to develop an efficient procedure to design the optimal shape of compliant mechanisms for specified load-deflection paths and snap-through behaviour. Here, computationally intensive simulations are required to approximate the entire load path as a function of the shape or spatial variables. Solving this problem efficiently, as will be demonstrated in this thesis, requires unconventional multidisciplinary strategies, as conventional modern techniques are ineffective or impractical.

In the case of simulation in the loop, where the numerical simulation is evaluated directly in the optimisation loop, unavoidable numerical discontinuities are present in the objective function. These discontinuities grow in number and size as the complexity and dimensionality of the problem increases. Modern gradient-based optimisers are incapable of bypassing these discontinuities and terminate prematurely, misrepresenting these discontinuities as local minima. Therefore, this research advocates for the use of non-negative gradient projection points, utilised by gradient-only optimisation techniques, to define meaningful shape optimisation solutions. These techniques ignore the discontinuous changes in function value to find non-negative gradient projection points.

Simulation in the loop is limited by the sequential nature of iterating from design to design, incurring the time penalty of having to wait for time-consuming simulations. Surrogate-based optimisation parallelises the time-consuming computational simulations, enabling computationally efficient surrogate models to be constructed instead. As the load paths evolve with systematic load application, these models evolve not only spatially but also temporally as a function of a pseudo-time variable. Spatial and temporal variables result in two sources of anisotropy. Firstly, the response anisotropy of the function as the temporal evolution of load-deflection curves is distinct from the spatial evolution as a function of shape variables. Secondly, sampling anisotropy as spatial variables are sampled distinctly from the usually densely sampled temporal variable resulting from iteratively evolving the load-deflection path. Response and sampling anisotropies can result in significant mismatches between the model forms from sampled data and typical isotropic kernels used in surrogate construction.

This study develops two solutions that address the response and sampling anisotropies, respectively. The results definitively demonstrate that both sources of anisotropy need to be addressed to construct accurate surrogate models that are meaningful for the shape optimisation problem.

First, a novel coordinate transformation scheme is developed to transform the function response to be more isotropic as a data pre-processing step. The key here is to utilise gradient information to estimate an updated isotropic reference frame, which also makes the strategy more tractable for higher dimensions. Secondly, sampling anisotropy is addressed by redistributing the surrogate kernels over the spatio-temporal domain and relying on regression to fit the surrogate response as opposed to limiting the centres to the sampling points. These improved surrogate models require significantly fewer computational resources to complete the optimisation problem as compared to placing the simulations directly in the optimisation loop.

ACKNOWLEDGEMENTS

No man is an island, entire of itself.

John Donne

I would like to express my sincere gratitude to my supervisors, Prof. Schalk Kok and Prof. Nico Wilke. It has been a privilege to learn from you both. Your guidance and assistance, extending beyond theoretical problems, have had an immeasurable impact on my journey.

To my parents, thank you for your unconditional love and support. You taught me that I alone decide what I am capable of.

Lastly, thank you to my friends for never letting me take anything to seriously.

Table of Contents

Chapter 1 Introduction

The following thesis documents the contribution of the author as a post-graduate student in the Department of Mechanical and Aeronautical Engineering at the University of Pretoria. Each of the following chapters is based on published or submitted papers and are self-contained advancements towards the development of a general and efficient procedure to perform shape optimisation of compliant mechanisms to produce desired snap-through behaviour. At the time of writing, the complete list of published articles are

- J. M. Bouwer, S. Kok and D. N. Wilke, "Challenges and solutions to arc-length controlled structural shape design problems", Mechanics-Based Design of Structures and Machines, pp.1–32, 2021.
- J. M. Bouwer, D. N. Wilke, and S. Kok, "Spatio-temporal gradient enhanced surrogate modeling strategies," Mathematical and Computational Applications, vol. 28, no. 2, 2023.
- J. Bouwer, D. N. Wilke, and S. Kok, "A novel and fully automated coordinate system transformation scheme for near-optimal surrogate construction," Computer Methods in Applied Mechanics and Engineering, vol. 419, p. 116648, 2024.

The finite element solver used in this research is adapted from in-house code of the Mechanical Engineering Department of the University of Pretoria, while all other methods and algorithms, unless stated otherwise, are developed and implemented by the author in Python.

This chapter briefly discusses and provides context for the research problem, and then the layout of the thesis is presented.

1.1 Background

The goal of this research is the shape optimisation of compliant mechanisms to produce a desired highly non-linear load path. Specifically, the desired load path will exhibit snap-through behaviour, i.e., the load path possesses multiple limit and equilibrium points. This optimisation problem exhibits many characteristics that render conventional modern optimisation approaches ineffective. Therefore, to complete this optimisation problem, current techniques will need to be improved, and new methods must be developed.

1.1.1 Problem Statement

In general, the unconstrained optimisation problem is expressed as

$$\underset{w.r.t\ \mathbf{x}}{\text{Minimise}}\, F(\mathbf{x}),\ \mathbf{x} = [x_1, x_2, ..., x_n]^{\mathrm{T}} \in \mathcal{R}^n, \tag{1.1}$$

where $F(\mathbf{x})$ is a scalar objective function to be minimised with respect to the column vector $\mathbf{x}$, which consists of real and continuous values. These values are referred to as design variables or, in the case of shape optimisation, as shape parameters. These shape parameters can include characteristics such as lengths, thicknesses, or radii of any given structure. There are two broad categories of methods that can solve this optimisation problem. These are direct optimisation methods [1–5], or Surrogate-Based Optimisation (SBO) methods [6–9].

Direct optimisation methods refer to the strategy in which the numerical simulation is evaluated directly in the optimisation loop. Although this strategy is typically convenient to implement, the employed methods can be computationally expensive due to the sequential nature in which the numerical simulations must be completed, as the next simulation in the optimisation path depends on the results of the previous simulation. Therefore, surrogate-based techniques were developed to parallelise the computationally expensive simulations. In SBO the simulation is completed for predetermined designs and the results used to construct a computationally efficient surrogate model. Specifically, in this research, the construction of spatio-temporal models is required. These models are needed because the load paths do not only evolve due to shape (or spatial) changes but also due to the systematic increase of a temporal variable that describes the position along the load path. As such, the models must incorporate both spatial and temporal variables to approximate the numerical simulation.

The implementation and the weaknesses of popular direct and SBO methods are discussed in more detail later in the thesis. Both of these options are considered and adapted to handle the complexity of the research problem.

1.1.2 Applications of Snap-through Structures

Snap-through structures are a family of structures that exhibit highly non-linear behaviour and typically possess multiple limit and equilibrium points in their load paths. Figure (1.1) shows an example snap-through structure, commonly known as the Lee Frame, as well as its associated non-linear load path with multiple limit and equilibrium positions [10, 11].

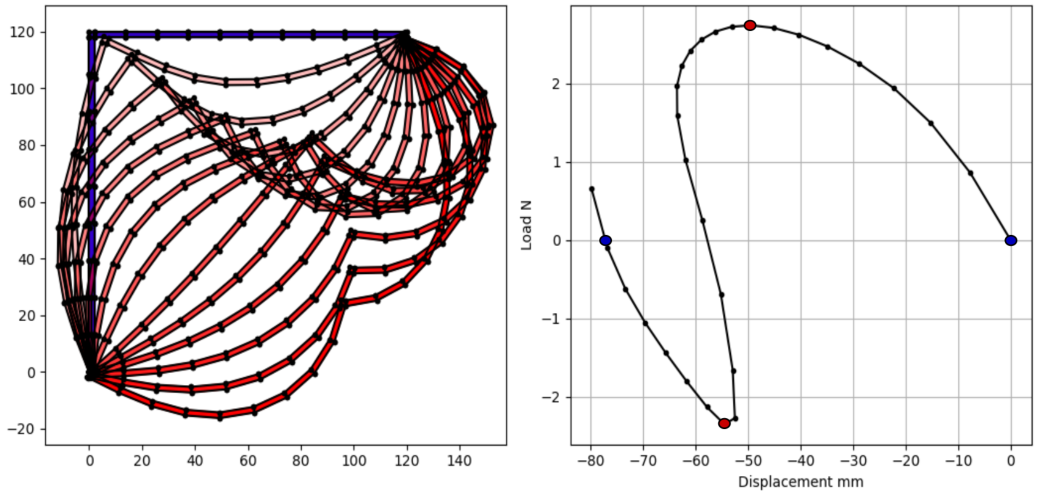


Figure 1.1. Displacement Overlay and Load-Displacement Curve for the Lee Frame with limit points (red dots) and equilibrium positions (blue dots) indicated on the load path.

Therefore, the goal of this research is to specify the load path, with any number of desired limit and equilibrium points, and to return the structure that will most closely exhibit this load path.

Snap-through behaviour is often used in the design of so-called compliant mechanisms. Compliant mechanisms are defined as any mechanism in which some portion of the mobility of the mechanism is gained through the flexibility of the structure itself. These mechanism are used in a many applications such as plastic bottle caps, mechanical switches, or micro-electro mechanical systems (MEMS).

Specifically, snap-through behaviour is used to develop compliant designs that require multiple stable equilibrium positions.

1.1.3 Analysis of Snap-through Structures

In this research, the function $F(\mathbf{x})$ in Equation (1.1) consists of a computationally expensive Finite Element (FE) simulation in which the nonlinear load path of a snap-through structure is solved. In non-linear FE analysis, the governing residual equation is expressed as

$$\mathbf{R} = \mathbf{K}(\mathbf{u})\mathbf{u} - \lambda\mathbf{F}, \tag{1.2}$$

where $\mathbf{K}$, $\mathbf{u}$, λ, and $\mathbf{F}$ denote the stiffness matrix, the nodal displacement vector, the load parameter, and the load vector respectively. In the load control scheme, the goal of some iterative non-linear solver is to find the nodal displacement vector at some specified load parameter value that reduces the governing residual, Equation (1.2), to $\mathbf{0}$ within some desired tolerance.

This iterative solver typically takes the form of Newton's method, where the gradient of Equation (1.2) is computed with respect to the nodal displacement vector. A problem with this solution strategy arises when the load path has a limit point, as Newton's method cannot fully trace the curve past this point.

Therefore, in this work, the Arc Length Control (ALC) algorithm is selected to provide the analysis. To justify this selection, three possible load-deflection paths are presented in Figure (1.2) that may be encountered during the optimisation process. Figure (1.2)(A) shows a curve that can be solved with the ALC method, load control, and displacement control. Figure (1.2)(B) can only be solved with displacement control or the ALC method. Finally, Figure (1.2)(C) can only be solved with the ALC method.

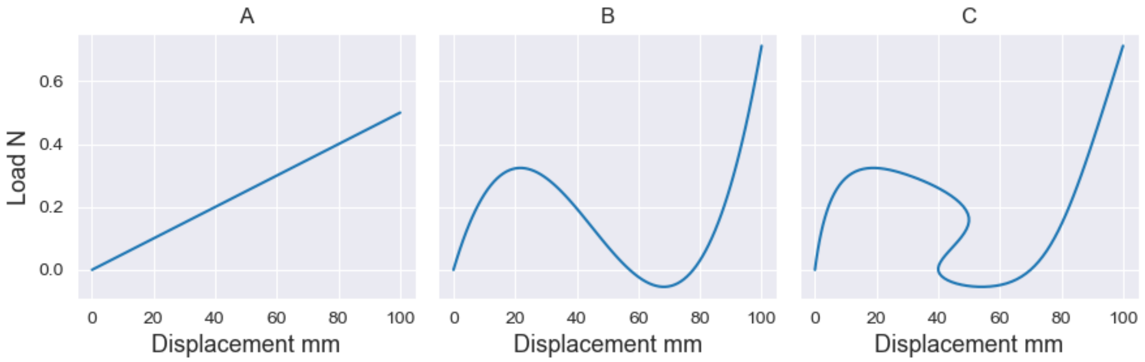


Figure 1.2. Three Possible Load-Deflection Curves that Structures can exhibit during the Optimisation Process.

Therefore, the ALC method allows for more complex behaviour to be simulated and, because of this, will also allow for more complex target curves to be provided for the optimisation problem. The implementation of the ALC method on displacement control means that the restrictions placed in previous work on this topic [1–3] are no longer needed, and far more general desired load-deflection paths can be specified in the shape-optimisation problem.

The ALC algorithm is a path-following method that was first proposed by Riks [12]. It is often implemented when load or displacement control algorithms fail to trace the entire load path due to the highly non-linear behaviour of the simulated structure. The algorithm is implemented in a non-linear Finite Element (FE) research code in the Octave environment.

The ALC algorithm proposes a solution to the Newton-Raphson algorithm's inability to bypass limit points by adding a constraint equation,

$$L^2 = \blacktriangle\mathbf{u}^{\mathrm{T}}\blacktriangle\mathbf{u} + \psi^2\blacktriangle\lambda^2\mathbf{F}^{\mathrm{T}}\mathbf{F}. \tag{1.3}$$

Here, L denotes the prescribed arc length, $\blacktriangle\mathbf{u}$ is the total displacement vector update for the current load step, $\blacktriangle\lambda$ is the total load parameter increment for the current load step, and ψ is some non-dimensional scale factor. The scale factor scales the load increment to a more appropriate value so that the load increment and displacement increment contribute similarly to the constraint equation. This variable ψ is often set to zero, typically referred to as a spherical constraint, but if selected appropriately, it can be beneficial for solution times. For the remainder of this research, it is assumed that the load vector $\mathbf{F}$ is a single-point load with unit magnitude. Therefore, Equation (1.3) can be simplified to

$$L^2 = \blacktriangle\mathbf{u}^{\mathrm{T}}\blacktriangle\mathbf{u} + \psi^2\blacktriangle\lambda^2. \tag{1.4}$$

This constraint equation implies that both the displacement vector update and load parameter increment are now unknowns. The prescribed arc length, L, and ψ variables are problem-dependent hyperparameters. Typically, the ALC algorithm requires the prescribed arc length, L, and then an accumulated arc length that terminates the simulation once reached. The selection of an appropriate prescribed arc length and accumulated arc length is problem specific, and often requires some iteration and experimentation to find a combination that yields satisfactory results.

To enhance the robustness of the ALC algorithm, the prescribed arc length can be automatically adjusted based on one of two scenarios. First, if there is no intersection between the prescribed arc length and the equilibrium path, the prescribed arc length decreases by some factor until an intersection is found. Secondly, if the intersection is not found within some set number of iterations, the prescribed arc length is also decreased by some factor and the equilibrium iterations begin again. This automatic prescribed arc length adjustment increases computational efficiency as the algorithm can now take large solution steps when the load path behaves close to a linear fashion, and then adjust to take smaller solution steps when the load path behaviour is more complex. The automatic prescribed arc length adjustment heuristic is central to the challenges and solution process followed in this research.

The inclusion of the arc length constraint creates a new system of equations that needs to be solved with some iterative solver. The reader is referred to the literature for solution strategies for this new system of equations [13–15].

1.2 Outline of Research Contributions

The overall contribution of this thesis is the development of a general, robust, and efficient optimisation strategy to complete shape optimisation of compliant mechanisms to produce a desired highly non-linear load path. These aforementioned criteria mean that the strategy must be able to accommodate a wide range of specified and simulated load paths, consistently find local and global minima in the design domain, and be efficient with computational resources. These criteria are used to develop and assess the developed optimisation framework.

To begin, the ALC algorithm is selected to facilitate the numerical simulation of highly non-linear load paths, as it is capable of tracing a wide range of load paths with multiple limit and equilibrium points. To improve the computational efficiency of the numerical simulations, the solution step size is automatically adjusted during the simulation. This means that large steps are taken during linear regions of the load path and then automatically reduced during highly non-linear regions of the load path. Although this reduces the computational cost of the numerical simulations, this automatic stepping means that the designer surrenders control of where simulation returns discrete solutions along the load path. This will create discontinuities in any objective function that attempts to quantify the discrepancy between a target and trail load path, as the number and locations of the points that describe these curves can differ greatly. These discontinuities can result in commonly used optimisation strategies to fail, as they can misinterpret these discontinuities as local minima. Figure (1.3) illustrates such an objective function for a two variable problem completed later in the study.

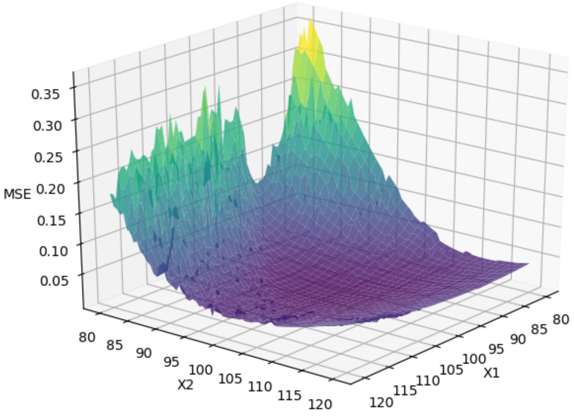


Figure 1.3. An example of a discontinuous objective function encountered in this research problem.

These discontinuities are numerical artefacts that would disappear in the limit of infinite computational resources where an extremely small and constant step size could be implemented. Therefore, to keep the optimisation strategy robust, these discontinuities should be ignored. Common best practice advocates to eliminate these discontinuities with either complex interpolation strategies, or for the implementation of zero-order optimisation methods [4, 5]. Instead, this work will find non-negative gradient projection points with gradient-only optimisation techniques [16–18]. These techniques should be capable of finding the user-specified load path regardless of the size and number of the discontinuities present in the problem.

These gradient-only techniques require that gradient information be made available. Therefore, the first contribution of this thesis is the development and verification of an analytical sensitivity procedure that incorporates the ALC algorithm. To the best of the author's knowledge, documented analytical sensitivity procedures that include the ALC algorithm prior to this research are limited to the first limit point in the load path [19, 20]. This analytical procedure will mean that computationally expensive numerical sensitivity strategies, such as forward difference or complex step, do not need to be implemented. The computational cost of these numerical methods also increases with the dimensionality of the problem, while the developed analytical procedure is relatively insensitive to the number of variables in the problem. This allows for higher-dimensional problems to be completed by the developed optimisation strategies.

Another source of computational cost is the requirement that the simulations must be completed in a sequential fashion. When the numerical simulation is placed in the optimisation loop, the next simulation can only be completed once the result of the previous simulation has dictated the next trail design. Therefore, if these simulations could be completed in parallel the computational cost of the optimisation process will be greatly reduced.

This parallelisation is achieved by implementing surrogate-based optimisation. The computationally expensive simulation is replaced by a computationally inexpensive model trained on generated data from numerous simulations completed in parallel. This model is then used during the optimisation process. These models will need to be a function of both the shape variables as well as the arc length variable to provide an approximation of the entire load path. This requires the implementation of either spatio-temporal or network models. Spatio-temporal models are a function of both the shape (spatial) variables as well as the arc length (temporal) variable, while the network models are a series of only spatial models constructed at predetermined locations in the temporal domain.

The accuracy of the chosen model will have a major impact on the quality of the found optimum design. There are two sources of concern that can negatively impact the performance of these models. First, common surrogate models are typically constructed with isotropic kernels. These isotropic kernels will therefore struggle to fit to anisotropic data manifolds. The anisotropic manifolds present in this problem arise from two sources, specifically,

1. the evolution of the various shape parameters can have vastly different impacts of the evolution of the load path, creating anisotropy in the spatial directions,
2. and the anisotropy due to the load path evolution as a function of the temporal variable compared to evolution as function of the spatial variables.

These sources of anisotropy can be addressed by completing a transformation of the coordinate system that will recast the problem into a reference frame where the function becomes isotropic. This concept is demonstrated in Figure (1.4) where a 2-dimensional problem is linearly transformed into reference frames where the function becomes progressively more isotropic.

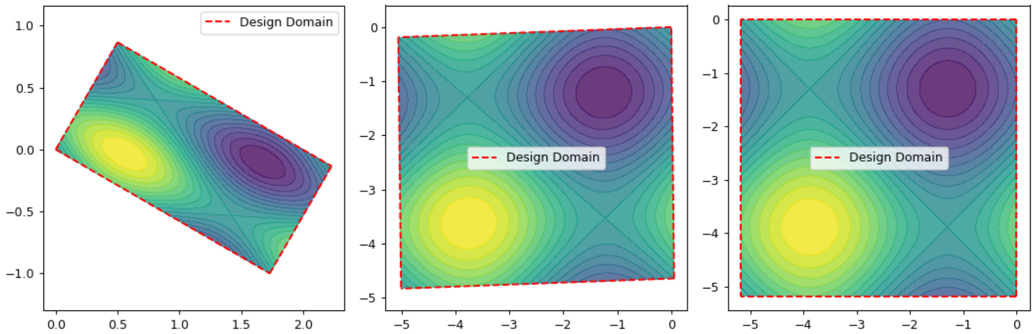


Figure 1.4. A figure illustrating the transformation of a coordinate system into an isotropic reference frame.

Finding a transformation that will complete this task is an information dense problem. For a n-dimensional problem, simply estimating one linear transformation requires a $n \times n$ rotation matrix and n scaling parameters. Therefore, making use of gradient vectors that grow with the dimensionality of the problem but remain computationally practical to compute using the developed sensitivity procedure will be necessary. The developed transformation scheme will assume that the underlying function of the problem is decomposable, i.e., that only a single linear transformation is required to find an isotropic coordinate system.

The second characteristic of this optimisation problem that will be detrimental to surrogate performance is specific to the spatio-temporal models. As these model incorporate both the spatial and temporal variables in the problem, it must deal with the anisotropic locations of the samples in the full spatio-temporal domain. This sampling anisotropy arises because spatial variables are sampled distinctly from the usually densely sampled temporal variable resulting from iteratively evolving the load-deflection path. This phenomena is shown in Figure (1.5) for a 3-dimensional problem where a full spatial problem is compared to a spatio-temporal problem.

This anisotropy can be eliminated by redistributing the centres, i.e. the locations of the kernels, throughout the full spatio-temporal domain. This redistribution once again creates an isotropic version

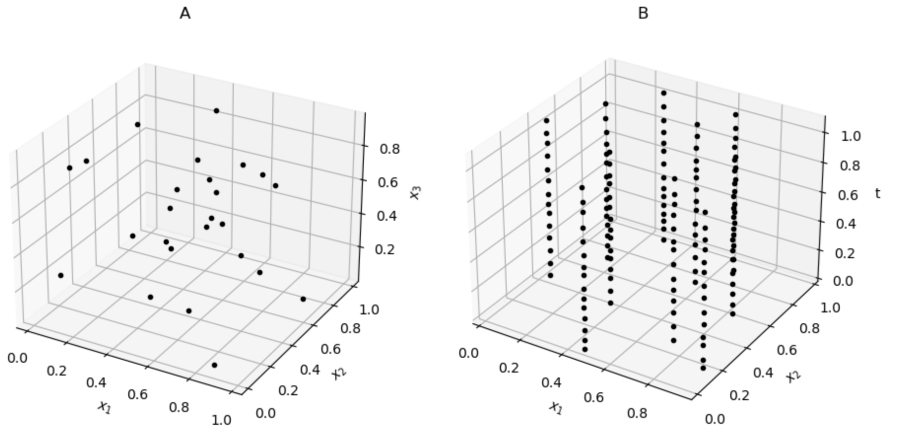


Figure 1.5. The differences between the sample locations in a full spatial problem (sub-figure A) and a spatio-temporal problem (sub-figure B) in three dimensions.

of the original anisotropic problem, which can greatly improve the accuracy of the spatio-temporal models. With the implementation of the discussed techniques, the original shape optimisation problem can them be resolved with hopefully greater computationally efficiency compared to the gradient-only simulation in the loop strategy.

All of these characteristics described in this section are present in a wide range of optimisation problems. Therefore, all the insights that are gained through this research will be applicable to the field of optimisation at large, and not simply on the shape optimisation of compliant snap-through mechanisms.

1.3 Structure of Thesis

The structure of the thesis is then as follows. Firstly, as almost all modern and computationally efficient optimisation techniques require gradient information, the first step of this research completed in Chapter 2 is to develop a design sensitivity procedure that is efficient and general. This procedure is unaffected by and remains reliable regardless of the complexity or non-linearity present in the simulation.

From this sensitivity procedure, it is possible to implement a direct optimisation strategy in Chapter 3. This technique places the computationally expensive simulation directly in the optimisation loop. This technique, due to the adaptive numerical solution strategy required to simulate snap-through structures, creates large and unavoidable discontinuities in the objective function. These discontinuities require the implementation of gradient-only optimisation algorithms, in order to find non-negative gradient projection points, as current widely used optimisation algorithms simply cannot reliably bypass the discontinuities and find the global best solution.

To improve the efficiency of the solution strategy Surrogate-Based Optimisation (SBO) is implemented. SBO is a technique where computationally inexpensive surrogate model replaces the computationally expensive Finite Element (FE) simulation. SBO has the benefit of implicitly storing the information gained from each simulation in the surrogate model, unlike direct optimisation which discards the information from the previous simulation once the optimisation step is completed.

The underlying surrogate models used in SBO often make the isotropic assumption, i.e., that all variables in the problem have an equal impact on the outcome. This assumption is not reasonable on the underlying problem in this research. Therefore, a novel coordinate system transformation scheme is developed in Chapter 4 that makes use of gradient information to recast the problem into a reference frame where the isotropic assumption is once again valid. In this chapter the proposed transformation method is only applied to test problems that are sampled uniformly.

Since the goal of the shape optimisation problem tackled in this work is to match the desired *response*, i.e., a desired load path, and not simply a single scalar value such as maximum displacement or stress. Two options are available to approximate the system response when using surrogates, as detailed in Chapter 5. The first option is to use a network of surrogate models, referred to as the Network Surrogate Model (NSM) technique, constructed at predetermined locations in the pseudo-time or arc length variable domain. The second option is the construction of so-called spatio-temporal surrogate models. In this case, the pseudo-time variable is included alongside the shape parameters as inputs to the surrogate model. Consequently, the system response is sampled in a anisotropic fashion: sparse sampling of the shape parameters and small incremental sampling in the pseudo-time direction. For the spatio-temporal models such nonuniform sampling of the domain is ill-suited to current surrogate modelling methods. Therefore, Chapter 6 demonstrates that these models become a useful addition to the optimisation procedure only once the centres of the model are redistributed throughout the spatio-temporal domain, and the coordinate system transformation scheme is implemented.

The remaining step completed in Chapter 6 is to compare the use of spatio-temporal surrogate models, using the developed techniques to address the anisotropy in the problem, to both the NSM models as well as the direct optimisation completed in Chapter 3. It is shown that the developed surrogate model construction techniques require far less computational resources than the direct optimisation approach.

Lastly, conclusions and areas of future research are offered in Chapter 7.

Chapter 2 Analytical Sensitivity Derivations

2.1 Chapter Abstract

The following chapter develops an analytical shape sensitivity procedure, published in [21], for the case where the arc length control algorithm is implemented with the four nodded assumed stress element. Analytical sensitivities will allow for the implementation and development of powerful gradient-based optimisation techniques throughout the study completed in this thesis.

The procedure generates the gradient information with respect to the node locations for both the load and displacement values throughout the entire load displacement path. It also computes the sensitivity of the load and displacement results with respect to the arc length variable.

The procedure is numerically validated with an example problem that increases in dimensionality that also demonstrates the benefit in solution times for the analytical procedure over the numerical finite difference approach.

2.2 Introduction

This chapter presents a procedure to calculate the analytical shape sensitivities [19,22,23] that includes the ALC procedure for multiple limit points. To the best of the author's knowledge, analytical shape sensitivities that includes the ACL procedure is limited to the first limit point [19,20], without offering detailed guidelines on how to proceed for multiple limit points.

This analytical procedure will greatly improve the efficiency with which snap-through structures can be designed using shape optimisation, for two reasons. First, more modern and efficient optimisation techniques can be implemented that make use of reliable gradient information. Second, these optimisation techniques will not need to rely on the computationally expensive finite difference method of calculating gradients.

2.3 Arc Length Control Sensitivity

The governing residual equation, with its dependencies, of a non-linear path independent FEM problem at equilibrium is expressed as

$$\mathbf{R}(\mathbf{u}(\lambda(\mathbf{x}),\mathbf{x}),\lambda(\mathbf{x}),\mathbf{x}) = \mathbf{0}, \tag{2.1}$$

where $\mathbf{u}$, λ and $\mathbf{x}$ denote the nodal displacement vector, the load parameter, and the design vector respectively. The gradient with respect to the design variable vector $\mathbf{x}$ is then expressed as

$$\frac{\partial \mathbf{R}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \lambda}\frac{\partial \lambda}{\partial \mathbf{x}} + \frac{\partial \mathbf{R}}{\partial \mathbf{u}}\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \frac{\partial \mathbf{R}}{\partial \lambda}\frac{\partial \lambda}{\partial \mathbf{x}} + \frac{\partial \mathbf{R}}{\partial \mathbf{x}} = \mathbf{0}. \tag{2.2}$$

The terms $\frac{\partial \mathbf{R}}{\partial \mathbf{u}}$ and $\frac{\partial \mathbf{R}}{\partial \lambda}$ are simply the tangent stiffness matrix $\mathbf{K}_T$ and the load vector $\mathbf{F}$ respectively. Therefore, Equation (2.2) can be re-written as

$$\mathbf{K}_T\frac{\partial \mathbf{u}}{\partial \mathbf{x}} + \left(\mathbf{K}_T\frac{\partial \mathbf{u}}{\partial \lambda} + \mathbf{F}\right)\frac{\partial \lambda}{\partial \mathbf{x}} = -\frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \tag{2.3}$$

Expressing the derivative of Equation (2.1) with respect to the load parameter results in the equation

$$\mathbf{K}_T \frac{\partial \mathbf{u}}{\partial \lambda} + \mathbf{F} = \mathbf{0}. \tag{2.4}$$

Substituting Equation (2.4) into Equation (2.3) results in the system of equations

$$\mathbf{K}_T \frac{\partial \mathbf{u}}{\partial \mathbf{x}} = -\frac{\partial \mathbf{R}}{\partial \mathbf{x}}. \tag{2.5}$$

This system of equations is then solved to find the displacement gradient term $\frac{\partial \mathbf{u}}{\partial \mathbf{x}}$. As the design variable term is a vector of values the resultant displacement derivative term is a $\mathcal{N} \times M$ matrix where $\mathcal{N}$ is the degrees of freedom in the mesh and M is the number of shape parameters in the design problem. The derivative of the residual vector with respect to the design variable vector, $\frac{\partial \mathbf{R}}{\partial \mathbf{x}}$, is dependent on the implemented solution scheme and the element type used in the mesh. This term is discussed and derived in Section (2.4).

In the case of the ALC algorithm, the displacement vector **u** and the load parameter λ both feature in the Arc Length constraint equation

$$\blacktriangle\mathbf{u}(\blacktriangle\lambda(\mathbf{x}),\mathbf{x})^{\mathrm{T}}\blacktriangle\mathbf{u}(\blacktriangle\lambda(\mathbf{x}),\mathbf{x}) + \psi^2 \blacktriangle\lambda(\mathbf{x})^2 = L^2. \tag{2.6}$$

Take note of the difference between the **u**, λ and $\blacktriangle\mathbf{u}$, $\blacktriangle\lambda$ terms in the Arc Length constraint equation. The Arc Length constraint equation is satisfied exactly at each iteration while attempting to find the equilibrium solution for a particular load step in the solution process [13]. The variables $\blacktriangle\mathbf{u}$ and $\blacktriangle\lambda$ denote the *total incremental* change of the nodal displacements and load parameter from the beginning to the end of a load step (i.e. adding all the iterative changes, $\Delta\mathbf{u}$ and $\Delta\lambda$, for a particular load step). The derivative of the Arc Length constraint equation is then expressed as

$$2\blacktriangle\mathbf{u}^{\mathrm{T}}\left(\frac{\partial \blacktriangle\mathbf{u}}{\partial \blacktriangle\lambda}\frac{\partial \blacktriangle\lambda}{\partial \mathbf{x}} + \frac{\partial \blacktriangle\mathbf{u}}{\partial \mathbf{x}}\right) + 2\psi^2 \blacktriangle\lambda \frac{\partial \blacktriangle\lambda}{\partial \mathbf{x}} = 0, \tag{2.7}$$

which can be simplified into

$$\blacktriangle\mathbf{u}^{\mathrm{T}}\frac{\partial \blacktriangle\mathbf{u}}{\partial \mathbf{x}} + \left(\blacktriangle\mathbf{u}^{\mathrm{T}}\frac{\partial \blacktriangle\mathbf{u}}{\partial \blacktriangle\lambda} + \psi^2 \blacktriangle\lambda\right)\frac{\partial \blacktriangle\lambda}{\partial \mathbf{x}} = 0. \tag{2.8}$$

What remains is to convert the *total incremental* terms in Equation (2.8) to the *total* terms in Equation (2.2). This can be done by expressing the *total* terms as the summation of all the previous *total incremental* terms,

$$\mathbf{u} = \sum_i^I \blacktriangle\mathbf{u}_i, \tag{2.9}$$

$$\lambda = \sum_i^I \blacktriangle\lambda_i, \tag{2.10}$$

and then rearranging the equations such that current *incremental* terms at solution step I, is the difference between the total terms and the summation of all the previous *incremental* terms, meaning from the summation up to $I-1$,

$$\blacktriangle\mathbf{u}_I = \mathbf{u} - \sum_i^{I-1} \blacktriangle\mathbf{u}_i, \tag{2.11}$$

$$\blacktriangle\lambda_I = \lambda - \sum_i^{I-1} \blacktriangle\lambda_i. \tag{2.12}$$

Computing the gradients of Equation (2.11) and (2.12) w.r.t. the design variables results in

$$\frac{\partial \blacktriangle\mathbf{u}_I}{\partial \mathbf{x}} = \frac{\partial \mathbf{u}}{\partial \lambda}\frac{\partial \lambda}{\partial \mathbf{x}} + \frac{\partial \mathbf{u}}{\partial \mathbf{x}} - \sum_i^{I-1}\frac{d\blacktriangle\mathbf{u}_i}{d\mathbf{x}}, \tag{2.13}$$

$$\frac{\partial \blacktriangle\lambda_I}{\partial \mathbf{x}} = \frac{\partial \lambda}{\partial \mathbf{x}} - \sum_i^{I-1}\frac{d\blacktriangle\lambda_i}{d\mathbf{x}}. \tag{2.14}$$

These equations are then substituted into Equation (2.8),

$$\blacktriangle\mathbf{u}^{\mathrm{T}}\left(\frac{\partial\mathbf{u}}{\partial\lambda}\frac{\partial\lambda}{\partial\mathbf{x}}+\frac{\partial\mathbf{u}}{\partial\mathbf{x}}-\sum_{i}^{I-1}\frac{d\blacktriangle\mathbf{u}_i}{d\mathbf{x}}\right)+\left(\blacktriangle\mathbf{u}^{\mathrm{T}}\frac{\partial\blacktriangle\mathbf{u}}{\partial\blacktriangle\lambda}+\psi^2\blacktriangle\lambda\right)\left(\frac{\partial\lambda}{\partial\mathbf{x}}-\sum_{i}^{I-1}\frac{d\blacktriangle\lambda_i}{d\mathbf{x}}\right)=0, \tag{2.15}$$

and then rearranged into

$$\blacktriangle\mathbf{u}^{\mathrm{T}}\frac{\partial\mathbf{u}}{\partial\mathbf{x}}+\left(\blacktriangle\mathbf{u}^{\mathrm{T}}\frac{\partial\mathbf{u}}{\partial\lambda}+\psi^2\blacktriangle\lambda\right)\frac{\partial\lambda}{\partial\mathbf{x}}=\blacktriangle\mathbf{u}^{\mathrm{T}}\sum_{i}^{I-1}\frac{d\blacktriangle\mathbf{u}_i}{d\mathbf{x}}+\left(\blacktriangle\mathbf{u}^{\mathrm{T}}\frac{\partial\mathbf{u}}{\partial\lambda}+\psi^2\blacktriangle\lambda\right)\sum_{i}^{I-1}\frac{d\blacktriangle\lambda_i}{d\mathbf{x}}. \tag{2.16}$$

This equation can then be used to solve for the load parameter derivative $\frac{\partial\lambda}{\partial\mathbf{x}}$ as all the other terms in the equation are known. The incremental terms, $\blacktriangle\mathbf{u}$ and $\blacktriangle\lambda$, are known as the sensitivity calculation occurs at the end of a solution step, and the gradient terms $\frac{\partial\mathbf{u}}{\partial\mathbf{x}}$ and $\frac{\partial\mathbf{u}}{\partial\lambda}$ are calculated from Equations (2.5) and (2.4) respectively. For the first solution increment the terms $\sum_i^{I-1}\frac{d\blacktriangle\mathbf{u}}{d\mathbf{x}}$ and $\sum_i^{I-1}\frac{d\blacktriangle\lambda}{d\mathbf{x}}$ are zero.

A key difference that the derivations demonstrate between typical sensitivity analysis, where load or displacement control is used, and the sensitivity analysis with the ALC algorithm is that the sensitivity calculation is a function of the solution variables at previous iterations within the current load step. Typically the sensitivity calculation is independent of the solution variables at intermediate iterations, being only dependent on the converged values of the solution variables.

2.4 Residual Sensitivity

The residual derivative, $\frac{\partial\mathbf{R}}{\partial\mathbf{x}}$, can be calculated using the chain rule

$$\frac{\partial\mathbf{R}}{\partial\mathbf{x}}=\frac{\partial\mathbf{R}}{\partial\mathcal{X}}\frac{\partial\mathcal{X}}{\partial\mathbf{x}}, \tag{2.17}$$

where $\mathcal{X}$ denotes the global coordinates of the nodes in the mesh. Therefore the term $\frac{\partial\mathcal{X}}{\partial\mathbf{x}}$ quantifies the effect the design variables have on the nodal coordinates of the mesh. A forward finite difference scheme, where the design variables are altered by some small value and the structure re-meshed, is used in this research to compute this term.

Since structures that exhibit highly non-linear behaviour, such as snap-through, typically exhibit this behaviour when loaded in bending, the 4-noded assumed stress element [24] is used for the simulations in this research. This element type is exact in bending for linear elasticity, and remains highly accurate in bending for non-linear elasticity. The internal residual and tangent stiffness matrix of the assumed stress element are expressed as

$$\mathbf{R}^{\mathrm{int}}=\sum_e(t_e\mathbf{G}_e\beta_e-\mathbf{F}_e), \tag{2.18}$$

$$\mathbf{K}=\sum_e t_e(\mathbf{L}_e+\mathbf{G}_e\mathbf{H}_e^{-1}\mathbf{G}_e^{\mathrm{T}}), \tag{2.19}$$

where $\mathbf{F}_e$ is the nodal load vector, t_e is the thickness of the element and the remaining terms, $\mathbf{G}_e$, β_e, $\mathbf{L}_e$, and $\mathbf{H}_e$ are typically calculated using Gauss quadrature. These terms are calculated from

$$\mathbf{G}_e=\sum_{GP}\mathbf{B}^{\mathrm{T}}\mathcal{F}\mathbf{B}\det(\mathbf{J})w_{GP}, \tag{2.20}$$

$$\beta_{\mathbf{e}}=\mathbf{H}_e^{-1}\mathbf{M}_e, \tag{2.21}$$

$$\mathbf{M}_e=\sum_{GP}\mathbf{P}^{\mathrm{T}}\mathbf{E}\det(\mathbf{J})w_{GP} \tag{2.22}$$

$$\mathbf{L}_e=\sum_{GP}\mathbf{B}^{\mathrm{T}}\mathbf{S}\mathbf{B}\det(\mathbf{J})w_{GP}, \tag{2.23}$$

$$\mathbf{H}_e=\sum_{GP}\mathbf{P}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{P}\det(\mathbf{J})w_{GP}, \tag{2.24}$$

where the terms **J**, **B**, **P**, **S**, **E**, and **C** are the Jacobian matrix, strain-displacement matrix, stress interpolation matrix, the second Piola-Kirchoff stress matrix, the Green-Lagrange strain, and the material constant matrix respectively. The Gauss quadrature weights are indicated by w_{GP}.

The derivatives of the internal residual vector and the tangent stiffness matrix with respect to the nodal coordinates are

$$\frac{d\mathbf{R}^{\text{int}}}{d\mathcal{X}} = \sum_e t_e \left(\frac{d\mathbf{G}_e}{d\mathcal{X}} \beta_e + \mathbf{G}_e \frac{d\beta_e}{d\mathcal{X}} \right), \tag{2.25}$$

$$\frac{d\mathbf{K}}{d\mathcal{X}} = \sum_e t_e \left(\frac{d\mathbf{L}_e}{d\mathcal{X}} + \frac{d\mathbf{G}_e}{d\mathcal{X}} \mathbf{H}_e^{-1} \mathbf{G}_e^{\text{T}} + \mathbf{G}_e \frac{d\mathbf{H}_e^{-1}}{d\mathcal{X}} \mathbf{G}_e + \mathbf{G}_e \mathbf{H}_e^{-1} \frac{d\mathbf{G}_e}{d\mathcal{X}} \right). \tag{2.26}$$

Therefore, the required derivatives of Equations (2.20) – (2.24) are expressed as

$$\begin{aligned} \frac{d\mathbf{G}_e}{d\mathcal{X}} = \sum_{GP} \Bigg(& \frac{d\mathbf{B}^{\text{T}}}{d\mathcal{X}} \mathcal{F} \mathbf{B} \det(\mathbf{J}) + \mathbf{B}^{\text{T}} \frac{d\mathcal{F}}{d\mathcal{X}} \mathbf{B} \det(\mathbf{J}) \\ & + \mathbf{B} \mathcal{F} \frac{d\mathbf{B}}{d\mathcal{X}} \det(\mathbf{J}) + \mathbf{B}^{\text{T}} \mathcal{F} \mathbf{B} \frac{d \det(\mathbf{J})}{d\mathcal{X}} \Bigg) w_{GP}, \end{aligned} \tag{2.27}$$

$$\frac{d\beta_{\mathbf{e}}}{d\mathcal{X}} = \frac{d\mathbf{H}_e^{-1}}{d\mathcal{X}} \mathbf{M}_e + \mathbf{H}_e^{-1} \frac{d\mathbf{M}_e}{d\mathcal{X}}, \tag{2.28}$$

$$\frac{d\mathbf{M}_e}{d\mathcal{X}} = \sum_{GP} \left(\frac{d\mathbf{P}^{\text{T}}}{d\mathcal{X}} \mathbf{E} \det(\mathbf{J}) + \mathbf{P}^{\text{T}} \frac{d\mathbf{E}}{d\mathcal{X}} \mathbf{P} \det(\mathbf{J}) + \mathbf{P}^{\text{T}} \mathbf{E} \frac{d \det(\mathbf{J})}{d\mathcal{X}} \right) w_{GP}, \tag{2.29}$$

$$\begin{aligned} \frac{d\mathbf{L}_e}{d\mathcal{X}} = \sum_{GP} \Bigg(& \frac{d\mathbf{B}^{\text{T}}}{d\mathcal{X}} \mathbf{S} \mathbf{B} \det(\mathbf{J}) + \mathbf{B}^{\text{T}} \frac{d\mathbf{S}}{d\mathcal{X}} \mathbf{B} \det(\mathbf{J}) \\ & + \mathbf{B}^{\text{T}} \mathbf{S} \frac{d\mathbf{B}}{d\mathcal{X}} \det(\mathbf{J}) + \mathbf{B}^{\text{T}} \mathbf{S} \mathbf{B} \frac{d \det(\mathbf{J})}{d\mathcal{X}} \Bigg) w_{GP}, \end{aligned} \tag{2.30}$$

$$\frac{d\mathbf{H}_e}{d\mathcal{X}} = \sum_{GP} \left(\frac{d\mathbf{P}^{\text{T}}}{d\mathcal{X}} \mathbf{C}^{-1} \mathbf{P} \det(\mathbf{J}) + \mathbf{P}^{\text{T}} \mathbf{C}^{-1} \frac{d\mathbf{P}}{d\mathcal{X}} \det(\mathbf{J} + \mathbf{P}^{\text{T}} \mathbf{C}^{-1} \mathbf{P} \frac{d \det(\mathbf{J})}{d\mathcal{X}} \right) w_{GP}. \tag{2.31}$$

These equations require the unknown terms $\frac{d\mathbf{B}}{d\mathcal{X}}$, $\frac{d\mathcal{F}}{d\mathcal{X}}$, $\frac{d \det(\mathbf{J})}{d\mathcal{X}}$, $\frac{d\mathbf{H}_e^{-1}}{d\mathcal{X}}$, $\frac{d\mathbf{P}}{d\mathcal{X}}$, $\frac{d\mathbf{E}}{d\mathcal{X}}$, and $\frac{d\mathbf{S}}{d\mathcal{X}}$. The term $\frac{d\mathbf{H}_e^{-1}}{d\mathcal{X}}$ is calculated using the known relationship

$$\frac{d\mathbf{H}_e^{-1}}{d\mathcal{X}} = -\mathbf{H}_e^{-1} \frac{d\mathbf{H}_e}{d\mathcal{X}} \mathbf{H}_e^{-1}. \tag{2.32}$$

The terms $\frac{d\mathcal{F}}{d\mathcal{X}}$, $\frac{d\mathbf{S}}{d\mathcal{X}}$, and $\frac{d\mathbf{E}}{d\mathcal{X}}$ are calculated from the expressions

$$\frac{d\mathcal{F}}{d\mathcal{X}} = \frac{d\mathbf{B}}{d\mathcal{X}} \mathbf{u}_e, \tag{2.33}$$

$$\frac{d\mathbf{S}}{d\mathcal{X}} = \frac{d\mathbf{P}}{d\mathcal{X}} \beta_e + \mathbf{P} \frac{d\beta_e}{d\mathcal{X}}, \tag{2.34}$$

$$\frac{d\mathbf{E}}{d\mathcal{X}} = \frac{d\mathcal{F}}{d\mathcal{X}} \mathcal{F} + \mathcal{F} \frac{d\mathcal{F}}{d\mathcal{X}}. \tag{2.35}$$

The Jacobian term for 2D analysis is expressed as

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \\ \dfrac{\partial x}{\partial \zeta} & \dfrac{\partial y}{\partial \zeta} \end{bmatrix}, \tag{2.36}$$

where variables x and y are global coordinates and variables η and ζ are local element coordinates. The relationships between these two sets of coordinates are expressed in the shape functions assumed for FEM. In the case where the isoparametric formulation is used, these shape functions are used for both the displacement and geometry. Equation (2.36) can then be rewritten as

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial N_1}{\partial \eta} & \dfrac{\partial N_2}{\partial \eta} & \dfrac{\partial N_3}{\partial \eta} & \dfrac{\partial N_4}{\partial \eta} \\ \dfrac{\partial N_1}{\partial \zeta} & \dfrac{\partial N_2}{\partial \zeta} & \dfrac{\partial N_3}{\partial \zeta} & \dfrac{\partial N_4}{\partial \zeta} \end{bmatrix} \mathcal{X}_e, \tag{2.37}$$

where the shape functions, N_1, N_2, N_3, and N_4 for a 4 noded element are expressed as

$$\begin{aligned} N_1 &= \frac{1}{4}(1-\eta)(1-\zeta), \\ N_2 &= \frac{1}{4}(1+\eta)(1-\zeta), \\ N_3 &= \frac{1}{4}(1+\eta)(1+\zeta), \\ N_4 &= \frac{1}{4}(1-\eta)(1+\zeta). \end{aligned} \tag{2.38}$$

The determinant of $\mathbf{J}$ is described as

$$\begin{vmatrix} \dfrac{\partial x}{\partial \eta} & \dfrac{\partial y}{\partial \eta} \\ \dfrac{\partial x}{\partial \zeta} & \dfrac{\partial y}{\partial \zeta} \end{vmatrix} = \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \zeta} - \frac{\partial y}{\partial \eta}\frac{\partial x}{\partial \zeta}, \tag{2.39}$$

where the terms in Equation (2.39) can easily be found from Equation (2.37). Therefore, the $\frac{d\,\det(J)}{d\mathcal{X}}$ term is described as

$$\begin{aligned} \frac{d\,\det(\mathbf{J})}{dX_n^1} &= \frac{\partial N_n}{\partial \eta}\frac{\partial y}{\partial \zeta} + \frac{\partial N_n}{\partial \zeta}\frac{\partial y}{\partial \eta} \\ \frac{d\,\det(\mathbf{J})}{dX_n^2} &= \frac{\partial N_n}{\partial \zeta}\frac{\partial x}{\partial \eta} + \frac{\partial N_n}{\partial \eta}\frac{\partial x}{\partial \zeta}. \end{aligned} \tag{2.40}$$

The terms $\mathcal{X}_n^d$ denote the element node number n and d indicates the degree-of-freedom.

The strain-displacement term, $\mathbf{B}$, is described as

$$\mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{J}^{-1} & 0 \\ 0 & \mathbf{J}^{-1} \end{bmatrix} \mathbf{T}, \tag{2.41}$$

where $\mathbf{T}$ is only a function of the local co-ordinate variables. Therefore, the derivative of this equation is then simply

$$\frac{d\mathbf{B}}{d\mathbf{D}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{d\mathbf{J}^{-1}}{d\mathcal{X}} & 0 \\ 0 & \frac{d\mathbf{J}^{-1}}{d\mathcal{X}} \end{bmatrix} \mathbf{T}, \tag{2.42}$$

as both the constant matrix and the term **T** are independent of the nodal coordinates of the element. The final remaining unknown term, $\frac{d\mathbf{J}^{-1}}{d\mathcal{X}}$, is then calculated using

$$\frac{d\mathbf{J}^{-1}}{d\mathcal{X}} = -\mathbf{J}^{-1}\frac{d\mathbf{J}}{d\mathcal{X}}\mathbf{J}^{-1}, \tag{2.43}$$

where the term $\dfrac{d\mathbf{J}}{d\mathcal{X}}$ is expressed as

$$\frac{d\mathbf{J}}{d\mathcal{X}_n^1} = \begin{bmatrix} \dfrac{\partial N_n}{\partial \eta} & 0 \\ \dfrac{\partial N_n}{\partial \zeta} & 0 \end{bmatrix}$$

$$\frac{d\mathbf{J}}{d\mathcal{X}_n^2} = \begin{bmatrix} 0 & \dfrac{\partial N_n}{\partial \eta} \\ 0 & \dfrac{\partial N_n}{\partial \zeta} \end{bmatrix}. \tag{2.44}$$

In summary, the total sensitivity procedure can be separated into two basic sub-procedures. The first sub-procedure, Sub-procedure (1), calculates the internal residual derivative and is dependent on both the implemented element and meshing scheme.

```
Sub-procedure 1: Residual Derivative for Assumed Stress Element
Result: $\frac{d\mathbf{R}}{d\mathbf{x}}$
Calculate $\frac{d\mathcal{X}}{d\mathbf{x}}$;
for All Degrees of Freedom in the Mesh do
    Compute Equation (2.44);
    Use this result and Equation (2.43) to compute Equation (2.42);
    Compute Equation (2.40);
    Determine Equations (2.25) and (2.26) using Equations (2.27) to (2.31);
    for All Design Variables do
        Multiply the result by the appropriate component in the $\frac{d\mathcal{X}}{dx}$ matrix;
        Assemble this component into the $\frac{d\mathbf{R}}{dx}$ matrix;
    end
end
```

The other sub-procedure, Sub-procedure (2), incorporates the first sub-procedure and is dependent on the ALC method. The implementation of these sub-procedures creates an analytical sensitivity procedure for highly geometrically non-linear structures.

```
Sub-procedure 2: ALC Analytical Sensitivity
Result: $\frac{d\mathbf{u}}{dx}$ and $\frac{d\lambda}{dx}$ for each solution step.
Set $\frac{d\mathbf{u}}{dx}$ and $\frac{d\lambda}{dx}$ to 0;
for End of Each Solution Steps do
    Complete Algorithm (1);
    Solve Equation (2.16);
    Accumulate the $\frac{d\mathbf{u}}{dx}$ and $\frac{d\lambda}{dx}$ vectors;
end
```

2.5 Sensitivity Verification

To verify the proposed analytical sensitivity procedure, outlined in Sections (2.3) and (2.4), the Deep Semi-Circular Arch structure is used. The original structure has a radius of 100mm and spans 270° (from -45° to 225°). The boundary conditions are asymmetric, with the left end pinned and the right end clamped. This asymmetry ensures that the structure will always initially deform to the left as the pinned boundary condition is less stiff than the clamped boundary condition. The geometry is parameterised using, 1, 2, 4 and 8 design variables, as shown in Figure (2.1). The chosen design variables describe the radius of the arch at equal angle increments from -45° to 225°, excluding the end points which remain at a radius of 100mm. The arch radius for any angle $\theta \in -45^\circ;225^\circ$ is then described by a cubic spline.

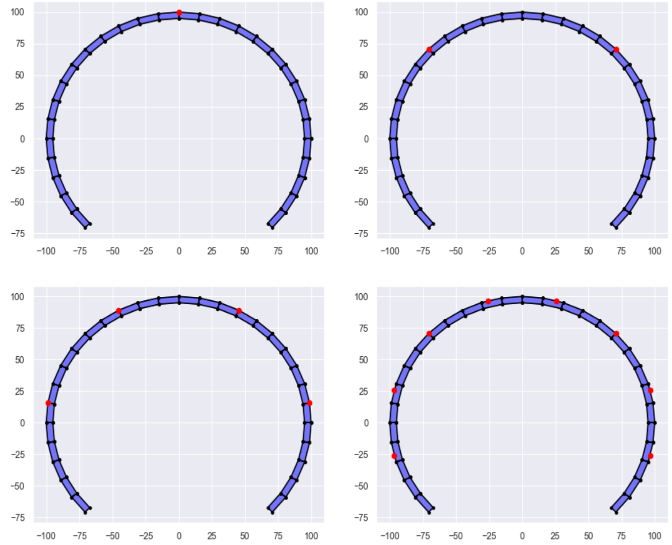


Figure 2.1. Shape parameter positions indicated in red for increasing dimensionality.

Figure (2.2) shows the resultant displacement overlay and load displacement curve.

The effect of these shape variables on the load path can be calculated with a sensitivity study. This is done using a forward finite difference scheme as well as the proposed analytical sensitivity procedure. The two methods are compared at four different locations along the arc length variable for one, two, and four dimensional problems. The sensitivity of the loaded node U_L and λ_L, indicated in red in Figure (2.2), is used for this comparison. The results in Tables (2.1) - (2.6) show that the numerical and analytical partial derivatives of the gradient vector for the circular design shown in Figure (2.2) agree up to the 4th significant figure, when using a perturbation size of 10^{-6} in the forward finite difference scheme. Therefore the proposed analytical sensitivity procedure produces the correct results.

Table 2.1. Displacement Sensitivity, $\frac{dU_L}{d\mathbf{x}}$, Results for Forward Finite Difference and Analytical Procedures for the One Variable Problem at Four locations in the Arc Length Domain.

Accumulated Arc Length	**400**	**1000**	**1600**	**2000**
Analytical Procedure	0.03404428	-0.0507639	-1.04374297	-0.1844686
Forward Finite Difference	0.0340304	-0.0507698	-1.03473532	-0.1844291
Ratio	1.000300	0.9998837	1.0087052	1.000214

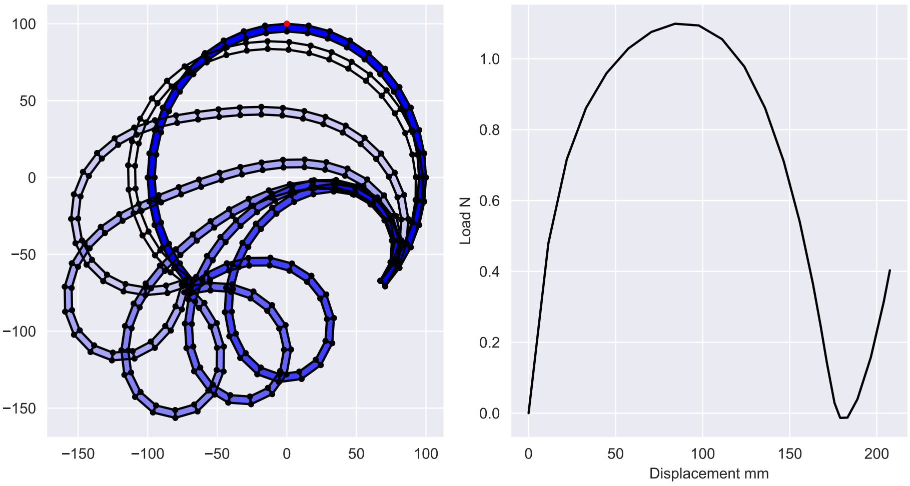


Figure 2.2. Displacement Overlay and Load Displacement Curve for a Symmetrical Deep Semi-Circular Arch.

Table 2.2. Load Sensitivity, $\frac{d\lambda_L}{d\mathbf{x}}$, Results for Forward Finite Difference and Analytical Procedures for the One Variable Problem at Four locations in the Arc Length Domain.

Accumulated Arc Length	**400**	**1000**	**1600**	**2000**
Analytical Procedure	-0.02100906	-0.01510655	0.01866241	-0.0318967
Forward Finite Difference	-0.02100897	-0.01510661	0.01866189	-0.0318977
Ratio	1.0000424	0.99999577	1.0000278	0.999987

Table 2.3. Displacement Sensitivity, $\frac{dU_L}{d\mathbf{x}}$, Results for Forward Finite Difference and Analytical Procedures for the Two Variable Problem at Four locations in the Arc Length Domain.

Accumulated Arc Length	**400**	**1000**	**1600**	**2000**
x_1				
Analytical Procedure	0.08857023	0.04147087	-0.88527241	0.43308858
Forward Finite Difference	0.08856114	0.04146612	-0.88526949	0.43308432
Ratio	1.00010255	1.00011437	1.0000033	1.00081204
x_2				
Analytical Procedure	-0.05027044	-0.09858645	-0.28893837	-0.64061523
Forward Finite Difference	-0.05027264	-0.09859181	-0.28892333	-0.64060926
Ratio	0.99995624	0.99994558	1.00005207	1.00077512

Another benefit of this analytical procedure is the increase in computational efficiency. Using finite difference schemes, multiple simulations per design variable are required to calculate the gradient of the objective function. This can become computationally expensive if multiple design variables are used or if the simulation itself is computationally expensive. The computational cost of the analytical method on the other hand is insensitive to the number of design variables. Table (2.7) shows the recorded solution times to compute the sensitivity results of this section.

Table 2.4. Load Sensitivity, $\frac{d\lambda_L}{d\mathbf{x}}$, Results for Forward Finite Difference and Analytical Procedures for the Two Variable Problem at Four locations in the Arc Length Domain.

Accumulated Arc Length	**400**	**1000**	**1600**	**2000**
		x_1		
Analytical Procedure	-0.00548538	-0.00399829	0.01026828	-0.0456492
Forward Finite Difference	-0.00548509	-0.00399837	0.01026855	-0.04561758
Ratio	1.00005297	0.99997809	0.99997358	1.00069308
		x_2		
Analytical Procedure	-0.0181498	-0.01299657	0.01072692	0.00976539
Forward Finite Difference	-0.01814999	-0.01299666	0.01072569	0.00976615
Ratio	0.99998951	0.99999331	1.00011503	1.0002562

Table 2.5. Displacement Sensitivity, $\frac{dU_L}{d\mathbf{x}}$, Results for Forward Finite Difference and Analytical Procedures for the Four Variable Problem at Four locations in the Arc Length Domain.

Accumulated Arc Length	**400**	**1000**	**1600**	**2000**
		x_1		
Analytical Procedure	-0.140657	-0.03900764	-0.03955682	-0.27462305
Forward Finite Difference	-0.14066252	-0.03901245	-0.03955876	-0.27469613
Ratio	0.99996072	0.99987662	0.99995097	1.00019186
		x_2		
Analytical Procedure	0.21768181	0.12473369	-1.01310569	0.17237048
Forward Finite Difference	0.21767762	0.12473147	-1.01311008	0.17127932
Ratio	1.00001928	1.0000178	0.99999566	1.00095851
		x_3		
Analytical Procedure	-0.01029556	-0.1702627	0.10939974	1.05805826
Forward Finite Difference	-0.01029873	-0.17026403	0.10941162	1.05898409
Ratio	0.99969202	0.99999216	0.99989148	0.99256477
		x_4		
Analytical Procedure	-0.1172282	0.0279739	-0.23573261	-1.85925168
Forward Finite Difference	-0.11722784	0.0279685	-0.23573114	-1.85375379
Ratio	1.00000376	1.00019304	1.00000625	1.00296582

2.6 Chapter Conclusion

This chapter develops and verifies an efficient and general procedure that can calculate the design sensitivities for structures that are simulated with the ALC method. Unlike previous research this procedure is not limited to the first limit point in the load path, but rather the sensitivity information is available throughout the entire load path regardless of its complexity.

This procedure can now be incorporated in the reminder of the research into all the investigated and developed gradient-based optimisation methods in both the direct and surrogate-based techniques.

Table 2.6. Load Sensitivity, $\frac{d\lambda_L}{d\mathbf{x}}$, Results for Forward Finite Difference and Analytical Procedures for the Four Variable Problem at Four locations in the Arc Length Domain.

Accumulated Arc Length	**400**	**1000**	**1600**	**2000**
x_1				
Analytical Procedure	-0.00130446	-0.00195405	-0.00044107	0.00680901
Forward Finite Difference	-0.00130379	-0.00195405	-0.00044083	0.00681081
Ratio	1.00051449	0.99999686	1.00053829	0.99973507
x_2				
Analytical Procedure	-0.00695875	-0.0036526	0.02563156	-0.03213392
Forward Finite Difference	-0.00695905	-0.00365266	0.02563178	-0.03207152
Ratio	0.99995668	0.99998548	0.99999142	1.0019456
x_3				
Analytical Procedure	-0.00409695	-0.0041402	-0.03110107	-0.05417153
Forward Finite Difference	-0.00409705	-0.00414017	-0.03110138	-0.05417311
Ratio	0.999977	1.00000733	0.99999018	0.99996378
x_4				
Analytical Procedure	-0.01493864	-0.00996073	0.03780535	0.07281055
Forward Finite Difference	-0.01493856	-0.0099608	0.03780438	0.07265629
Ratio	1.00000517	0.99999302	1.00002583	1.00021231

Table 2.7. Analysis and Sensitivity Calculation Times in seconds for Increasing Number of Variables

Number of Variables	**1**	**2**	**4**	**8**	**16**
Forward Difference	54.2	81	135	243	459
Analytical	48.1	48.2	48.1	48.6	48.5

Chapter 3 Direct Optimisation Methods

3.1 Chapter Abstract

The following chapter makes use of the sensitivity procedure developed in Chapter 2 to complete the optimisation of load paths exhibited by snap-through structures [21].

A general and reliable optimisation scheme that performs shape optimisation of snap-through structures in order to match a target load-deflection curve is presented. The developed optimisation scheme is insensitive to the complexity of the target or simulated load-deflection paths.

The objective function for the optimisation problem is the mismatch between the simulated load-deflection curve and the target load-deflection curve. The simulation of the snap-through structures typically requires the implementation of the Arc Length Control (ALC) method. This path-following algorithm allows the solution of problems that typical non-linear finite element simulations with load or displacement control are unable to simulate fully.

The most general implementation of this algorithm automatically adjusts the solution step size during the simulation, to enhance the algorithm's ability to solve highly non-linear problems. This adaptive stepping results in the presence of discontinuities in the objective function. The discontinuities are a direct result of the changing number of solution steps in the simulation, and the location of these solutions along the load-deflection curve. Conventional wisdom dictates that these discontinuities must be eliminated by using a small constant step size, or the discontinuous objective functions must be solved by zero-order methods. However, the author opts to solve the optimisation problems using gradient-only optimisation. Gradient-only optimisation defines non-negative gradient projection points as the solution to the optimisation problem.

Numerical examples demonstrate that the discontinuities in the objective function render conventional gradient-based approaches ineffective [25, 26]. This is due to the use of function value information to determine if a local minimiser was found. Function value based methods can misinterpret discontinuities as minima and hence terminate at poor solutions. In contrast, gradient-only algorithms are demonstrated to be insensitive to numerical discontinuities, and they locate the correct solutions reliably.

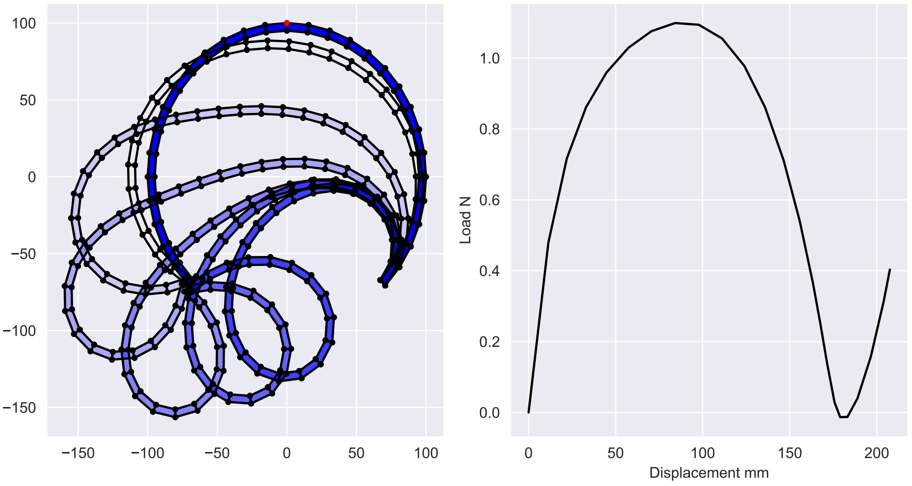
3.2 Introduction

The goal of this chapter is to develop a general and consistent method to complete shape optimisation of highly non-linear snap-through structures where the FE simulation is located directly in the optimisation loop. The developed optimisation scheme must be insensitive to the complexity or non-linearity present in the problem. This task involves optimising the values of various shape parameters that describe a known snap-through structure such that the structure exhibits a desired load-deflection path.

Previous work in the task of optimising these structures [1–5], are almost exclusively concerned with topology optimisation of these structures. The majority of previous work [1–3] limit the complexity of the simulated load-deflection paths such that displacement control is adequate to fully simulate the structure's behaviour. This research on the other hand places no limit on the complexity of the simulated behaviour and implements a far more reliable simulation and optimisation strategy.

The main contribution of this research over previously established work is the handling of complications that arise due to the complex solution strategy required to always be able to solve these highly non-linear load-deflection paths [27]. Although discussed in more technical detail later in this chapter, the solution strategy implemented to solve these load-deflection paths requires the designer to surrender control of where the simulation returns known solution locations. To overcome this lack of control, researchers needed to implement complex interpolation strategies to find the solutions at standard locations [4, 5]. The results in this chapter show that these complex interpolation strategies are a redundant ingredient in an already complex design problem.

Figure (3.1) shows an example snap-through structure, commonly known as the Deep Semi-Circular Arch [10], as well as its associated load-deflection curve. Although this chapter exclusively makes use of the Deep Semi-Circular Arch for demonstration purposes there are many other known snap-through structures. Leahu-Aluas and F. Abed-Meraim [11] offer many examples of snap-thorough structures as benchmark buckling problems. In addition, although the proposed solution strategy is applied to a shape optimisation problem, it can also be applied to solve topology optimisation problems where the structural analysis is performed by the arc length control algorithm.



(a) (b)

Figure 3.1. (a) The Deep Semi-Circular Arch as an example snap-through structure with the loaded node indicated in red on the initial undeformed state, and (b) the associated load-deflection curve of the loaded node.

The simulation of these structures solve for a discrete number of positions along the load-deflection path, typically using some variant of the Arc Length Control (ALC) algorithm [12]. But the real challenge is not analyzing a given structure to obtain the load-deflection curve, it is rather that the desired load-deflection curve is known and the structure that exhibits this behaviour is unknown. Therefore an inverse problem needs to be solved that minimizes the discrepancy between the desired load-deflection curve and a simulated curve. Since subtle changes in the structure's shape parameters could drastically change the resultant load-deflection curve, automatic arc length adjustment is used to ensure that the analyses can be performed for any design.

However, an unintended consequence of automatic arc length adjustment is that it creates discontinuities in the objective function that quantifies the discrepancy between the target load-deflection curve and the simulation counterpart. Note however that these discontinuities are purely a numerical artefact: their presence should be ignored by any algorithm that seeks to find the optimal solution. Figure (3.2) shows such a discontinuous objective function for a simple two variable load path optimisation problem. It is evident that classical gradient-based optimisation algorithms will struggle to navigate this objective function landscape, as they often getting stuck at the discontinuities. A lesser known strategy is to locate non-negative gradient projection points (NN-GPPs) as defined in the gradient-only optimisation problem [28]. This strategy was developed specifically to optimise discontinuous objective functions, where the discontinuities are numerically induced [16–18, 25, 26, 28, 29]. Gradient-only optimisers developed to locate NN-GPPs [16, 26, 28, 29] have been demonstrated to solve problems of this type reliably and efficiently.

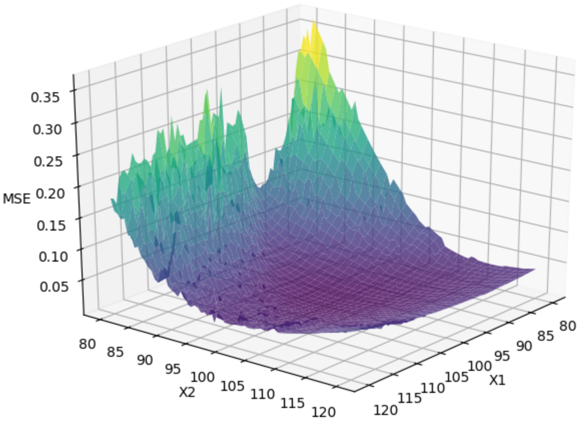


Figure 3.2. Example of a discontinuous objective function for highly non-linear load path optimisation

The structure of this chapter is as follows. Firstly, following Snyman and Wilke [28], Section (3.3) presents three formulations to define the solution to an optimisation problem. These formulations are the function minimiser, optimality criteria (necessary and sufficiency conditions) and NN-GPPs. The importance of NN-GPPs as a reliable optimiser formulation for piece-wise discontinuous functions is highlighted. From the ALC description the source of the discontinuities depicted in Figure (3.2) are detailed in Section (3.4). The chapter concludes with a systematic numerical investigation in Section (3.5), which highlights the inverse shape optimisation problem to recover a given load-displacement path for a selected snap-through structure [11]. The number of shape parameters are increased from two to eight, allowing the behaviour and performance of the optimisation algorithms to be investigated. The selected algorithms are the function minimization algorithm, Sequential Least Squares Programming (SLSQP) [30], the gradient-only sequential spherical approximation method (GOSSA) [28], and a Modified Subgradient method [26, 28].

3.3 Three Formulations for Optimisers

The problem considered in this research, which is to reduce the discrepancy between two curves, is an unconstrained optimisation problem. Three formulations exist to define solutions to this problem, namely function minimization (Section (3.3.1)), optimality criteria (first and second-order conditions) (Section (3.3.2)) and NN-GPPs (Section (3.3.3)).

3.3.1 Formulation 1 - Function minimiser

In general the unconstrained optimisation problem is expressed as

$$\underset{w.r.t\ \mathbf{x}}{\text{Minimize}}\, F(\mathbf{x}),\ \mathbf{x} = [x_1, x_2, ..., x_n]^{\mathrm{T}} \varepsilon \mathcal{R}^n, \tag{3.1}$$

where $F(\mathbf{x})$ is a scalar objective function to be minimized with respect to the column vector $\mathbf{x}$, which consists of real and continuous values. These values are referred to as design variables or, in the case of shape optimisation, shape parameters such as lengths, thicknesses, or radii. In this chapter the shape parameters consist of radii at specified angles.

The first option to define the optimal solution, $\mathbf{x}^*$, uses objective function values. The definition is

$$F(\mathbf{x}^*) \leq F(\mathbf{x}) \text{ for all } \mathbf{x}. \tag{3.2}$$

As it is typically impossible to evaluate the function $F(\mathbf{x})$ for all values of $\mathbf{x}$, the solution $\mathbf{x}^*$ is defined as a strong local minima if the above condition is met for some local region in multi-dimensional space that defines $\mathbf{x}$. The optimisation problem at this point has been defined as a minimization problem. gradient-based algorithms are typically considered as the most efficient to solve this optimisation problem. A conventional gradient-based optimisation algorithm will use the gradient information to determine a direction to update the proposed optimum solution until the minimization criterion, $F(\mathbf{x}^*) \leq F(\mathbf{x})$, is met. This implies that *only* the value of the function $F(\mathbf{x})$ is considered when defining an optimal solution. Considering the objective function landscape in Figure (3.2), the presence of discontinuities will result in premature convergence of such an algorithm. This necessitates an alternative definition of the optimal solution $\mathbf{x}^*$.

3.3.2 Formulation 2 - Optimality criteria

The optimal solution to an unconstrained minimization problem can also be defined by the optimality criteria. The 1st and 2nd order criteria, also defined as the necessary and sufficiency condition respectively, are given by

$$\frac{dF(\mathbf{x})}{d\mathbf{x}} = \mathbf{0}, \tag{3.3}$$

and

$$\mathbf{x}^{\mathrm{T}} \frac{d^2F(\mathbf{x})}{d\mathbf{x}^2} \mathbf{x} > 0 \ \forall \ \mathbf{x} \neq \mathbf{0}. \tag{3.4}$$

These criteria require that at the proposed solution the norm of the objective function gradient is zero, and the Hessian of the objective function is positive definite (the curvature in any direction is non-negative). These two conditions are sufficient to define a candidate local minimum. One can therefore define a candidate optimum position, $\mathbf{x}^*_{\mathbf{NS}}$, as any vector that results in the gradient vector to be zero. Of course there are potentially multiple solutions for which a scalar function can exhibit a zero gradient vector, such as a local maximum or a saddle point. Therefore, to define a local minimum, the sufficiency condition is required to define an optimum solution vector $\mathbf{x}^*_{\mathbf{N}}$. Algorithms that solve the optimality criteria typically require the Hessian of the objective function. Therefore these algorithms are not as popular as gradient-based algorithms that only require the gradient of the objective function.

3.3.3 Formulation 3 - NN-GPP

The last alternative to define the optimal solution to the unconstrained minimization problem is to locate non-negative gradient projection points $\mathbf{x}^*_{\mathbf{nngpp}}$ [16]. A non-negative gradient projection point is defined as a position in the design space for which any update to the solution vector, projected onto the objective function gradient vector, is positive (or zero). These points are defined formally as a point that for every $\mathbf{u}\{\mathbf{y} \in \mathcal{R}^n, ||\mathbf{y}|| = 1\}$ there exists a real number that satisfies the condition

$$\nabla_A F(\mathbf{x}^*_{\mathbf{nngpp}} + \lambda \mathbf{u})^{\mathrm{T}} \mathbf{u} \geq \ 0 \ \forall \ \lambda \ \in (0, r_u], \tag{3.5}$$

where λ is the magnitude of the vector in the search direction.

This implies that as one approaches and passes such a point, the projected gradient changes sign from negative to positive at $\mathbf{x}^*_{\mathbf{nngpp}}$. Note that this definition of the optimal solution requires *only* the gradient information of the objective function, since the objective function values themselves are never used. This criterion is sufficient to define a local optimum. Such a definition of optimality will be able to ignore the numerically induced discontinuities present in the problem of interest, as long as the gradient information remains consistent across such a discontinuity. A visual illustration of these non-negative gradient projection points, and how they differ from the other formulations, are presented in Section (3.3.4).

3.3.4 Optimiser Characteristics

To highlight the importance to differentiate between the four optimisers, $\mathbf{x}^*, \mathbf{x}^*_{\mathbf{NS}}, \mathbf{x}^*_{\mathbf{N}}, \mathbf{x}^*_{\mathbf{nngpp}}$, consider the four uni-variate objective functions shown in Figure (3.3):

- Two infinitely differentiable functions (C^∞), one being convex (Subproblem A) and the other non-convex (Subproblem C),
- a C^0 continuous function (Subproblem B), and
- a piece-wise discontinuous function (Subproblem D).

The associated derivatives are depicted in Figure (3.4).

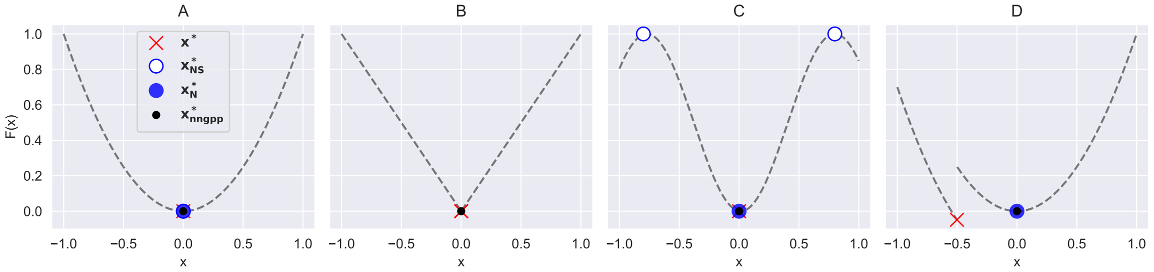


Figure 3.3. The Four Optimisers for Four One-Dimensional Problems

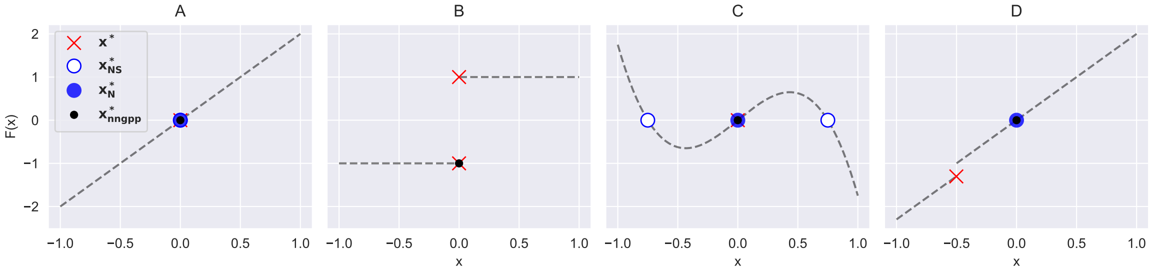


Figure 3.4. The Four Optimisers for Four One-Dimensional Problems in the Gradient Space

The four optimisers are equivalent for sub-problem A, the infinitely differentiable convex function shown in Figure (3.3). The infinitely differentiable non-convex function in sub-problem C distinguishes between $\mathbf{x}^*_{\mathbf{N}}$ and the other three optimisers, $\mathbf{x}^*$, $\mathbf{x}^*_{\mathbf{NS}}$ and $\mathbf{x}^*_{\mathbf{nngpp}}$, which are again equivalent for this problem.

The non-smooth function in sub-problem B demonstrates the differences between the two optimums $\mathbf{x}^*_{\mathbf{N}}$ and $\mathbf{x}^*_{\mathbf{NS}}$ and the remaining two optimums $\mathbf{x}^*$ and $\mathbf{x}^*_{\mathbf{nngpp}}$. The gradient plot shows that no x-position has a gradient of zero. Therefore, the two optimal positions $\mathbf{x}^*_{\mathbf{N}}$ and $\mathbf{x}^*_{\mathbf{NS}}$ do not exist.

Lastly, sub-problem D is the same as sub-problem A, but it contains a discontinuity at $x = -0.5$. This discontinuity creates a difference in the $\mathbf{x}^*$ point when compared to the other optimum points. Optimisers $\mathbf{x}^*_{\mathbf{NS}}, \mathbf{x}^*_{\mathbf{N}}$ and $\mathbf{x}^*_{\mathbf{nngpp}}$ agree exactly even for non-continuous problems.

Therefore, these simple problems demonstrate the reliable nature of the positive gradient projection points $\mathbf{x}^*_{\mathbf{nngpp}}$. These points are unaffected by local maxima, which can impact $\mathbf{x}^*_{\mathbf{N}}$, or discontinuities in either the function or gradient space which impact $\mathbf{x}^*$.

3.4 Discontinuities in the Design Problem

The automatic prescribed arc length adjustment used in this research creates discontinuities in any calculation that compares one load-deflection curve to another. This is because the number and locations of the discrete solution positions along the load path can differ suddenly due to an infinitesimal change in the structure shape. This can be shown by reconsidering the design problem in Chapter 2.

As the goal is to find a structure that exhibits the same load path as some specified load path, an objective function that quantifies the difference between two curves needs to be established. In general these load-deflection curves need not be functions, such as the curves presented in Figures (1.2)(A) and (B), which can be solved with displacement control as the load can be expressed as a function of the displacement. Instead the load-deflection curves are represented as parametric curves, such as Figure (1.2)(C), where the accumulated arc length is chosen as the independent variable and therefore requires the implementation of the ALC algorithm. However, one complication is that both curves are represented by a finite number of points on the curve. Also, the solution points are unlikely to be available at the identical accumulated arc lengths used to represent the target curve.

Previously, this complication was either avoided or overcome with two broad categories of methods. The first category involves limiting the complexity of the prescribed load-deflection path. Previous research [1–3] require the desired load-deflection behaviour to be non-parametric. This means that the problems must be such that the load can be expressed as a function of the displacement, such as in Sub-Figure (1.2)B. The desired path is then described with a set number of load values at predetermined displacement values. This method eliminates the aforementioned arc length adapting complication, as a simple displacement control solution scheme can be implemented and the ALC method is then no longer needed. However, this method cannot be used in general to solve more complex load-deflection path optimisation problems where the load cannot be expressed as a function of the displacement. Therefore, this method requires a restriction of the design space in order to avoid more complex load-deflection paths.

The second more general method implemented by Bruns et al. [4, 5] handles this problem by constructing a linear interpolation function that allows the designer to find the deformed state of the structure at accumulated arc length locations other than the returned simulation locations. This method requires the previously deformed state $\mathbf{q}$ at some accumulated arc length $\mathbf{a}$ and the current deformed state $\mathbf{q}^*$ at some greater accumulated arc length value $\mathbf{a}^*$. Then all the degrees of freedom in the mesh are interpolated linearly to some deformed state $\bar{\mathbf{q}}$ at some intermediate accumulated arc length value $\bar{\mathbf{a}}$ while ensuring that the ALC constraint condition is still met. This method can be computationally expensive for meshes with a large number of degrees of freedom. Therefore, a far simpler interpolation strategy is proposed whereby only interpolation functions for the parametric load and displacement curves are needed instead of interpolation functions for each degree of freedom in the mesh.

The analyst must still choose between two interpolation options, namely for each iteration to:

1. interpolate the target points to the simulated points at the same accumulated arc lengths, or
2. interpolate the simulated points to target points at the same accumulated arc lengths.

For the rest of this chapter interpolating the target curve is referred to as objective function **A** and interpolating the simulated curve as objective function **B**.

Interpolation between target points or simulated points is required each time the objective function is called, to ensure that the error terms are calculated at the same accumulated arc length. For the sake of simplicity, only linear interpolation is considered. Figure (3.5) depicts example load curves that illustrate both interpolation options graphically.

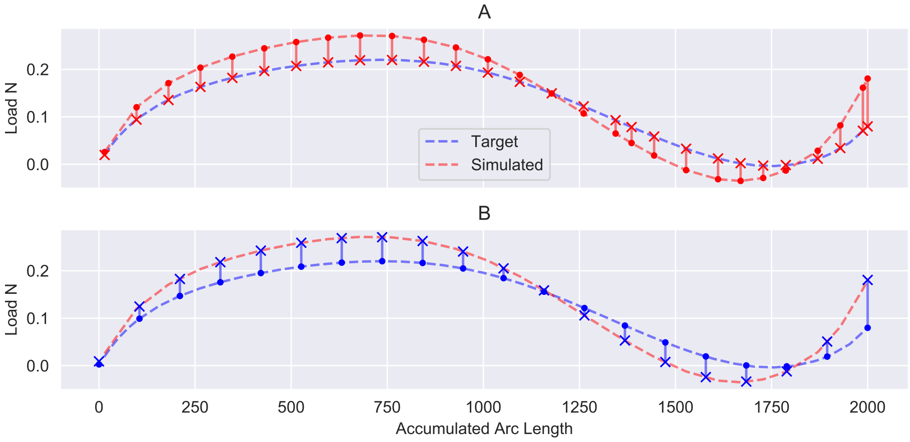


Figure 3.5. Comparison of two linear interpolation schemes with points denoting known locations and crosses interpolated locations. A - Interpolating the target curve, B - Interpolating the simulated curve.

Interpolating to the same accumulated arc lengths then allows for the computation of the mean square error,

$$MSE = \frac{1}{N}\sum_{i}^{N}(\lambda_i^{\mathrm{T}} - \lambda_i^d)^2 + \frac{1}{N}\sum_{i}^{N}(u_i^{\mathrm{T}} - u_i^d)^2, \tag{3.6}$$

between points on the target curve, λ_i^{T} and u_i^{T}, and points on the simulated curve of the current design, λ_i^d and u_i^d, at identical accumulated arc lengths. Here N is the number of solution steps taken to solve the curve (objective function **A**) or the number of points used to describe the target curve (objective function **B**). The gradient of this cost function is expressed as

$$\frac{dMSE}{d\mathbf{x}} = -\frac{2}{N}\sum_{i}^{N}(\lambda_i^{\mathrm{T}} - \lambda_i^d)\frac{d\lambda_i^d}{d\mathbf{x}} - \frac{2}{N}\sum_{i}^{N}(u_i^{\mathrm{T}} - u_i^d)\frac{du_i^d}{d\mathbf{x}}. \tag{3.7}$$

Sampling the proposed objective functions for the Deep Semi-Circular Arch 200 times between arc radii from 80mm to 120mm, where the load path at 100mm is the target curve, computes Figure (3.6). All the simulations were completed with the same prescribed arc length as well as a fixed accumulated arc length that terminates the solution once reached. The prescribed arc length is decreased by a factor of $\sqrt{2}$ when more than nine iterations are required to find a solution.

Objective function **A** contains numerous large discontinuities even for this simple one-dimensional problem. Objective function **B** lessens the severity of the discontinuities, but, there still seem to be a few small discontinuities present in the problem. To gain a better understanding of the discontinuities present, in either objective function, Figure (3.7) presents an explicit line search for a small variation in the design variable at the same shape.

Notice that both interpolation schemes create a discontinuity at the same location in the design space. The magnitude and direction of this discontinuity may differ, but neither interpolation scheme can

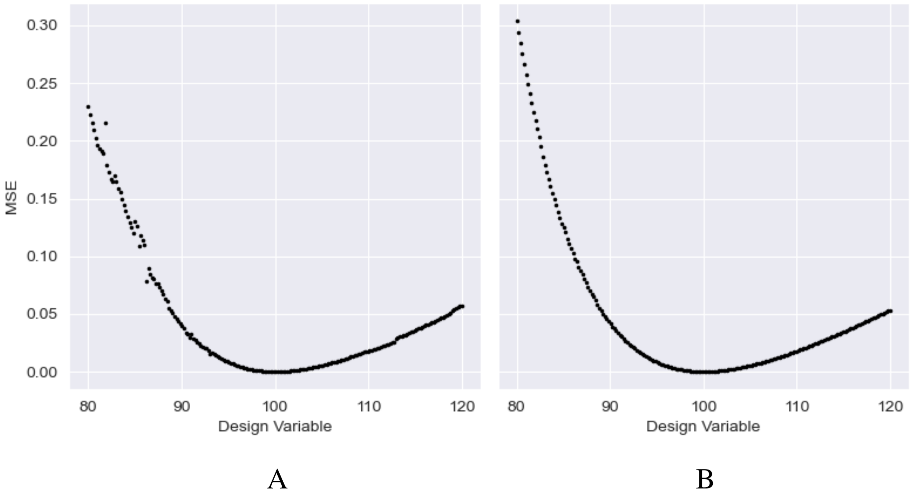


Figure 3.6. One-dimensional objective functions sampled 200 times between arc radii of 80 mm and 120 mm. Sub-figure A shows objective function **A**, while Sub-figure B shows objective function **B**.

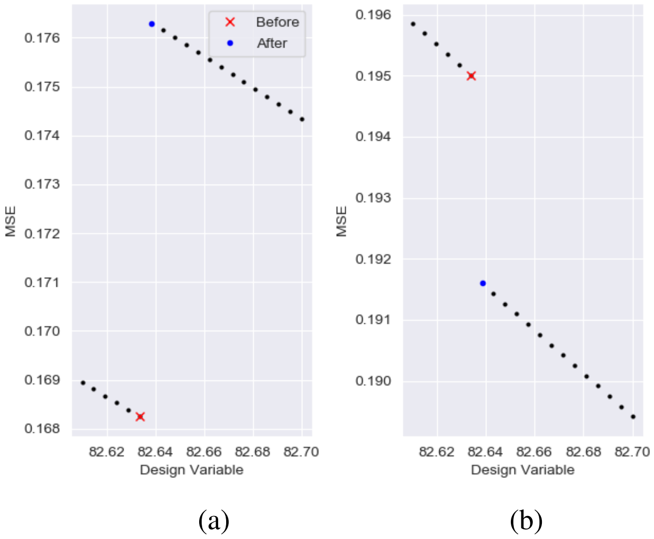


Figure 3.7. Dense line search at the discontinuity around 82.6mm for the same structure, for (a) objective function **A** and (b) objective function **B**. Note the reduction in the magnitude of the discontinuity in objective function **B** compared to objective function **A**.

avoid them from manifesting in the objective function. The underlying source of these discontinuities is the automatic arc length adjustment that creates a change in the number of discrete points along the curve as well as the coordinates of these points. This can be shown by plotting the solved load paths before and after the discontinuity in Figure (3.8).

These two load paths in Figure (3.8) are nearly identical but the ALC method returns significantly different discrete locations along these curves. Both the arches require arc length adjustments during the simulation but the size and number of these adjustments differ. This creates sudden and discontinuous changes in the objective function for either interpolation scheme.

For objective function **A** the source of these discontinuities is shown in Figure (3.9), where the load

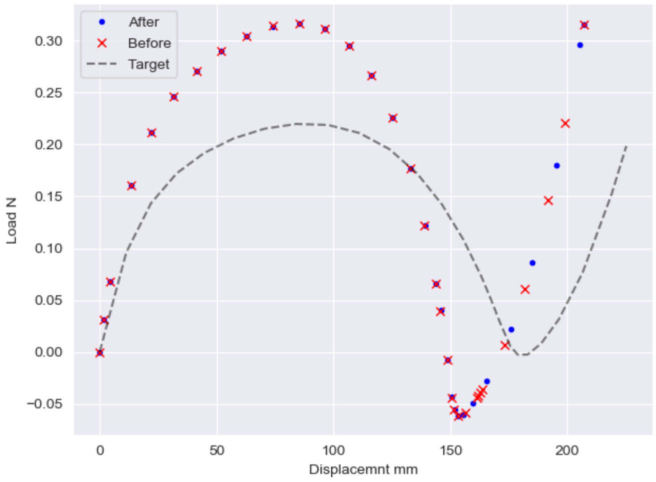


Figure 3.8. Discrete points on the solved load paths before and after the discontinuity.

and displacement curves before and after the discontinuity are shown in the arc length space. The change in the accumulated arc positions means the error calculation takes place at locations that could be further away from the target curve, creating an increase in the error calculation, even though these are still discrete locations on the same curve.

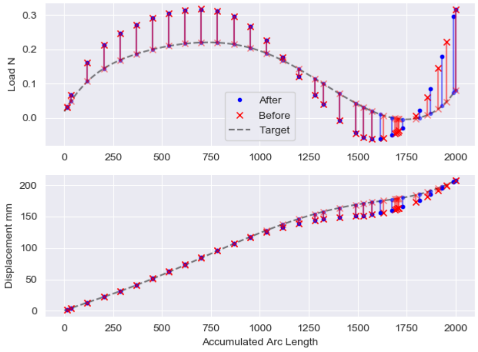


Figure 3.9. Discrete solved points before and after the discontinuity in the accumulated arc length space for the load and displacement curves.

In the case of objective function **B** it is not as obvious what the source of these discontinuities is. The error calculations consistently contain the same number of points and these points are consistently at the same accumulated arc length coordinates. Therefore, an analyst can easily assume that the objective function will contain no discontinuities, but, as can be seen in Figures (3.6) and (3.7), this is not the case.

Figure (3.10) explains the source of these discontinuities, where an example curve in the accumulated arc length domain is shown. When the locations of the solved positions along the curve change, the interpolation can suddenly return significantly different results. Assume that a structure is analysed, resulting in 5 solutions along the load path (the blue plus signs 1 to 5). An infinitesimal adjustment in the structure could trigger the prescribed arc length adjustment, resulting in 4 solutions along the load path (the green plus signs 1 to 4). The objective function is computed at different locations altogether, indicated with the solid blue dots. Since none of the solutions are available at the same locations, linear interpolation (dashed lines) is used to interpolate the solution coordinates to the required coordinates. Notice that the error contributions at locations i and $i+1$ undergo a step change due to different points along the same curve being used to compute the error contributions.

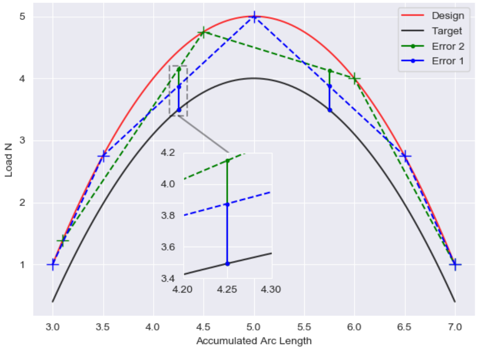


Figure 3.10. Illustration of how objective function **B** is computed, and how it can result in discontinuities.

Although Bruns et al. [4] make no mention of discontinuities due to their interpolation scheme in the solved optimisation problem, it is the author's opinion that there is anecdotal evidence of the presence of these discontinuities. Bruns et al. [4] report an improvement in the optimum result after decreasing the step size of the optimisation process and attribute this to a local minimum in the objective function. From Figure (3.10) it can be seen that a smaller step size will reduce the magnitude of the discontinuities in the objective function, therefore, the author believes that what is attributed to the optimiser bypassing a local minimum is in fact the optimiser initially misinterpreting a discontinuity as a minimum. When the step size was reduced the severity of the discontinuity was likely reduced such that the optimiser was then able to bypass it. Therefore, even the complex interpolation strategy implemented by Bruns et al. [4, 5] may still result in discontinuities in the objective function. This is consistent with Bruns et al. [5] reporting large oscillations in the objective function, referred to as "chattering", and stating that efforts to avoid these oscillations were ineffective.

To summarise, the differences in the number of solution points and their respective locations, combined with subsequent linear interpolation, result in discontinuities in the respective objective functions. These discontinuities are not from any physical property of the structure or inherent to the problem, but are rather numerical artefacts arising from the automatic adjustment in arc length that makes analyses more computationally efficient and reliable. Note that even though the objective function contains discontinuities, the gradient or derivative of the objective function is computable for every design in the design space.

3.4.1 Benefits of automatic arc length adjustment

To make use of any gradient-based optimisation algorithm, the gradient of the objective function is required. If the gradient of the objective function is calculated at or near a discontinuity using a numerical approximation (such as forward differences), the returned gradient could be incorrect in both magnitude and direction [28]. To avoid these discontinuities two steps can be taken, namely, the analyst can use a small enough prescribed arc length such that the simulation never requires an arc length adjustment, or the analysis code can be adapted to always produce a solution coordinate at the accumulated arc length value used to represent the target curve.

The first option, the small prescribed arc length, results in an increase in solution times. This can be demonstrated by solving the same symmetrical Deep Semi-Circular Arch for various prescribed arc lengths. Table (3.1) shows how for small prescribed arc length step sizes the solution times start to increase drastically when compared to a more appropriate arc length selection. Selecting a sufficiently small arc length step size is further complicated in that it cannot be determined *a priori* whether the arc length will be adjusted over all possible designs in the design space.

Table 3.1. Solution Times for Various Arc Length Step Sizes

Step Arc Length	**100**	**50**	**25**	**12.5**	**6.25**
Time (s)	26.2	42.5	53.67	89.61	163.75
Arc Adjustment	True	True	True	True	False

The second option of sampling the target and design curves at standard locations still requires the constructions of some interpolation functions for both the load and displacement as well as their respective gradients (whenever the original prescribed arc length step is too large to converge). As with any interpolation scheme, this will result in some loss of accuracy and can therefore still result in discontinuities.

Therefore, neither method of attempting to eliminate the discontinuities can entirely eliminate discontinuities from the objective function. In addition, the first proposed method can be computationally expensive. The complex-step method [28, 31] offers a means to compute sensitivities at a discontinuity but scales poorly with an increase in the number of design variables. This means that to make efficient use of gradient-based algorithms it is necessary to compute sensitivities analytically. That is why Chapter 2 developed an analytical sensitivity procedure for the ALC algorithm.

3.5 Load Path Optimisation

The performance of three different optimisers, that all make use of gradient information, are evaluated on the Deep Semi-Circular Arch load path optimisation problem. The optimisers are the Sequential Least Squares Programming (SLSQP) method, the Gradient Only Sequential Spherical Approximation method (GOSSA) [28], and a Modified Subgradient method [26, 28]. The SLSQP method is a well known optimiser and has been used on a wide variety of numerical optimisation problems [30]. The GOSSA and Modified Subgradient methods are gradient-only methods that have been shown by Wilke [26], Wilke et al. [16], [17], [18] and Snyman and Wilke [28] to be reliable when solving problems that contain discontinuities.

The GOSSA algorithm estimates the Hessian of the objective function, making use of the spherical assumption, for each iteration. The efficiency of this algorithm should be comparable to the SLSQP algorithm, but it will not misinterpret discontinuities in the objective function as minima. The Modified Subgradient method is a simple steepest descent method with a chosen fixed step size (no line searches are performed) and the function minimiser along the search trajectory is *not* stored. This algorithm is also reliable in the presence of discontinuities but its performance is dependant on a hyper-parameter, meaning it can be computationally expensive when compared to the other algorithms.

The goal of the optimisation problem is to find the values of the design variables that result in the same load path as the Deep Semi-Circular Arch, depicted in Figure (2.2). The load path of a fully symmetrical arch is chosen as the target curve as no matter the level of dimensionality, the fully symmetrical arch can be recovered exactly. This means the performance of the optimisers can be assessed easily as the global optimum is known (zero). The converged solutions of the SLSQP, GOSSA and Modified Subgradient methods will be referred to as $\mathbf{x}^*_{SLSQP}$, $\mathbf{x}^*_{GOSSA}$ and $\mathbf{x}^*_{MS}$ respectively in the sections that follow.

Inverse problems, such as the problem presented here, are often ill-posed i.e. different designs (structures) have the same load-deflection path. In the case of this design problem this is inconsequential.

The goal is to find *any* structure with the desired load-deflection path, not to find a *specific* structure with the target load-deflection path. Therefore, the uniqueness of the optimised solutions is not investigated.

3.5.1 Bifurcation of the Load-Deflection Path

The example design problem presented in this research, the Deep Semi-Circular Arch, can exhibit bifurcated load-deflection paths. Bifurcation refers to a loss of uniqueness in the solution to the non-linear FEM equilibrium equations. This loss of uniqueness typically presents itself as a "fork" in the equilibrium path, with there being multiple equilibrium positions of the structure for some value of the accumulated arc length parameter. The load-deflection curve then exhibits multiple branches. Figure (3.11) demonstrates an example of a bifurcation present in the Deep Semi-Circular Arch problem (produced by a perturbation of the original constant radius shape).

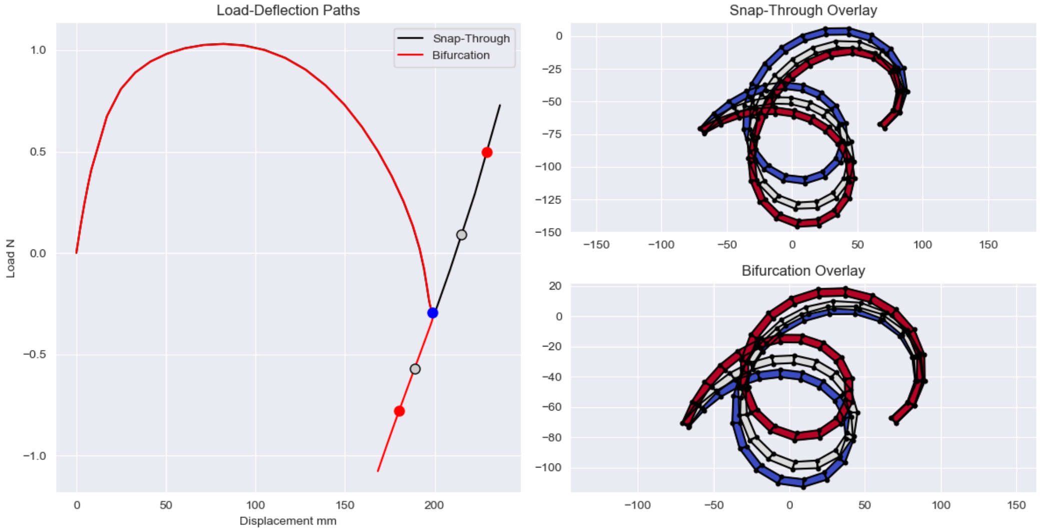


Figure 3.11. Demonstration of the type of bifurcation present in the example problem. The overlays show the "fork" equilibrium state in blue (identical for both figures) and the next two equilibrium states are shown in grey and red and their locations on the equilibrium paths.

Although there are numerical strategies to consistently stay on the same branch, often referred to as branch switching algorithms [32, 33], the solver implemented in this research makes no effort to avoid these bifurcated load paths. Rather, the sudden increase in the objective function value if the solver finds equilibrium on a different branch, should result in the optimisers moving away from areas in the design space where bifurcation is present. This can be demonstrated by selecting some initial starting points for the optimisers that describe structures that produce bifurcated load paths.

3.5.2 Two Shape Design Variables

The Two Variable Inverse Problem consists of two design variables at 45°and 135°that describe the radius at these locations (shown in Figure (2.1)). To visualize the two objective functions described in Equation ((3.6)), a grid of 75×75 samples between 80 mm and 120 mm is used. The resultant plots are shown in Figure (3.12). These 3D surface plots show the presence of similar discontinuities that were present in the one variable objective function. The global minimum is at the point $[100, 100]$.

The problem is solved with the selected optimisers using five randomly generated starting positions, two of which generate bifurcated load paths. The initial load path curves, and the target curve, are shown in Figure (3.13), while the returned optimised curves for the three methods are shown in

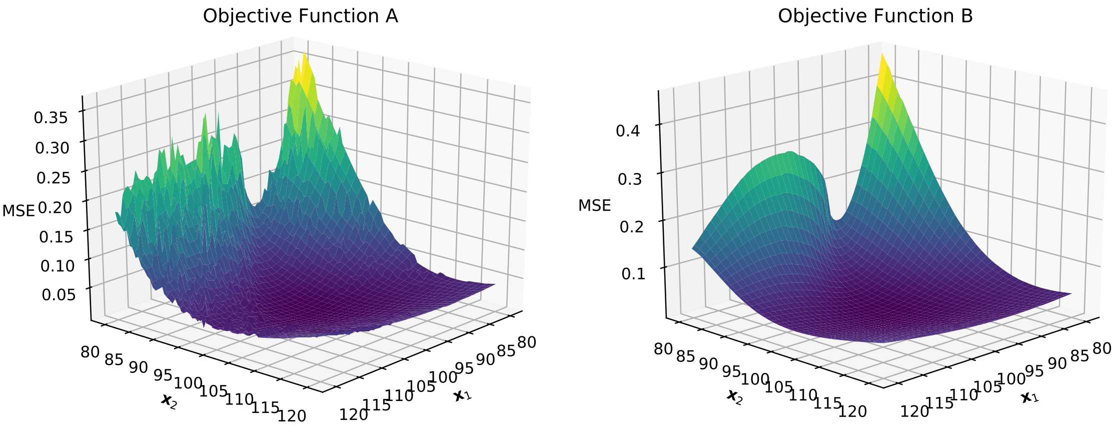


Figure 3.12. Objective functions of the Two Shape Design Variable Inverse Problem

Figure (3.14) for both objective functions. None of the returned optimal structures have bifurcated load-deflection curves, therefore the optimisers are able to bypass and avoid regions in the design space that result in bifurcated load-deflection curves.

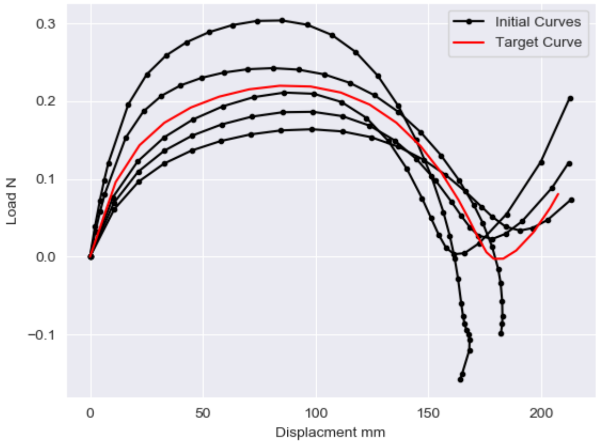


Figure 3.13. Initial Curves of the two variable optimisation problem.

The SLSQP algorithm fails to find the optimal solution for three of the five initial designs for both objective functions although objective function **B** does offer a small improvement in results. The GOSSA and Modified Subgradient algorithms consistently find the global minimum for all the starting positions and both objective functions. To explain the cause of the occasional failure of the SLSQP algorithm, the optimisation trajectories of the algorithms are presented in Figures (3.15) and (3.16).

These trajectories are overlaid with a quiver plot that shows the normalized gradient of the objective function (i.e. the arrows only show the direction and not the magnitude of the gradient vector). An important aspect to note of these gradient fields is the similarity between the two objective functions. Objective function **A** has larger discontinuities when compared to objective function **B** (see Figure (3.12)), but the resultant gradient fields for either objective function remain far more consistent.

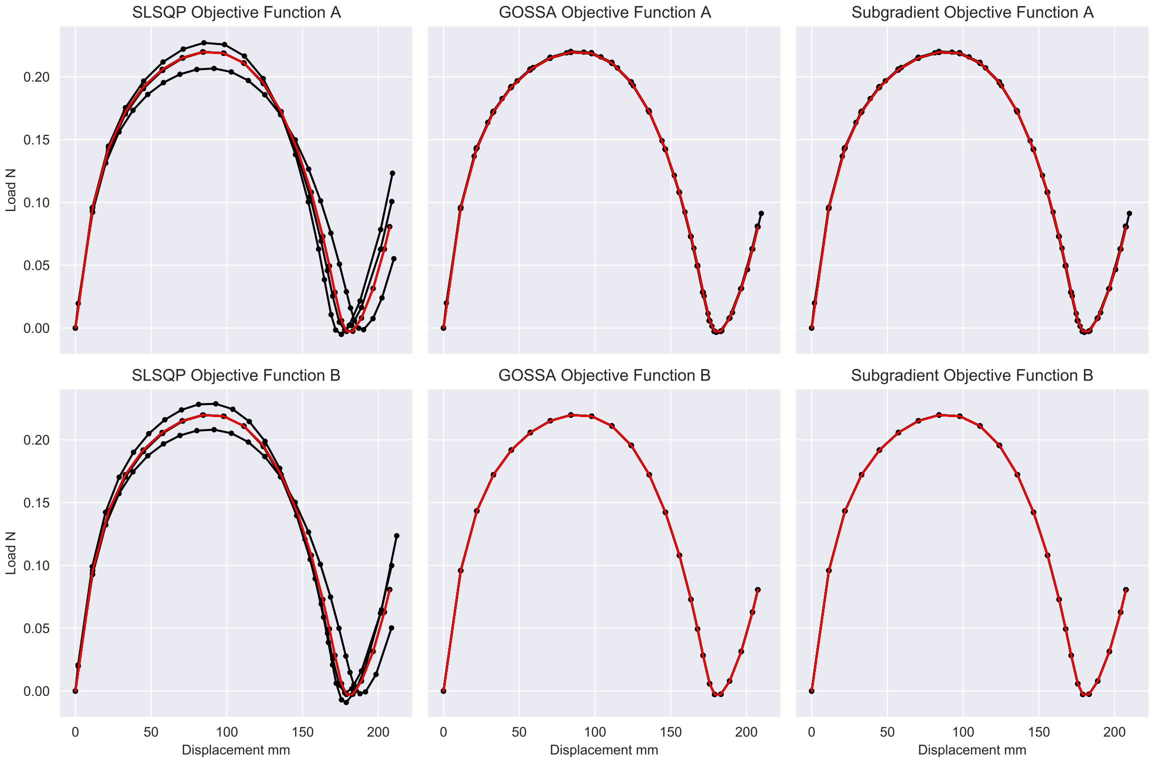


Figure 3.14. optimised Curves for the Three Algorithms and both objective functions for the Two Variable Inverse Problem.

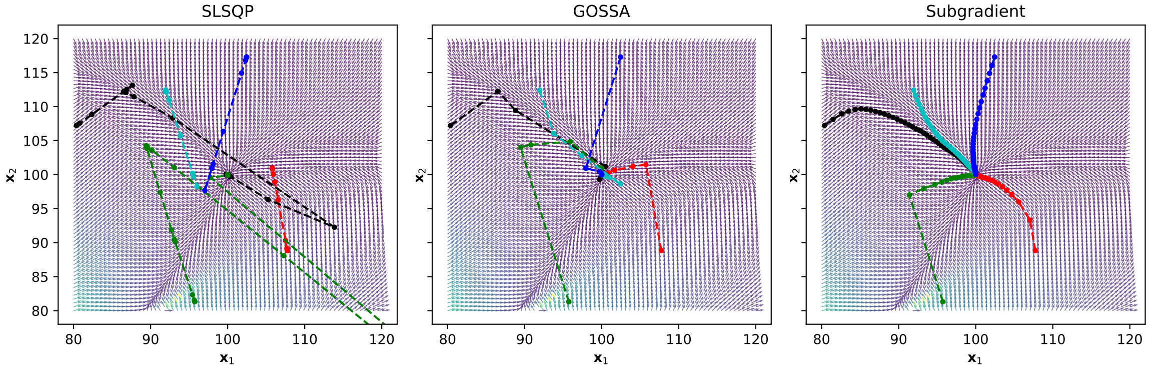


Figure 3.15. Five different optimisation trajectories, indicated as dashed coloured lines, for objective function **A** on the Two Variable Inverse Problem.

This means that the gradient information for the objective functions is far more reliable when compared to the discontinuous function value information.

The optimisation trajectory plots show that the SLSQP and GOSSA algorithms both follow similar optimisation paths, but the SLSQP method terminates early. This early termination is due to the discontinuities in the objective function caused by the adaptive stepping method of the ALC algorithm. This can be proven by completing a dense line search at these early termination points. Figure (3.17)

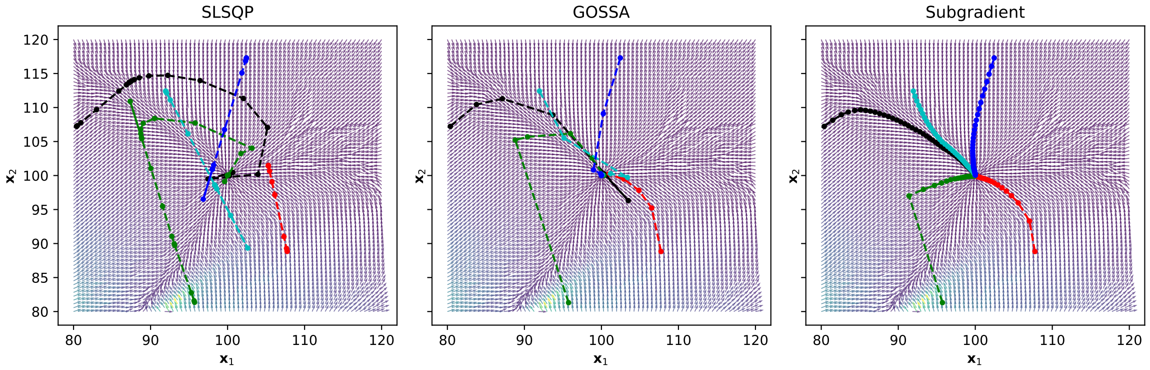


Figure 3.16. Five different optimisation trajectories, indicated as dashed coloured lines, for objective function **B** on the Two Variable Inverse Problem.

presents an explicit line search

$$MSE(\alpha) = MSE(\mathbf{x}^*_{SLSQP} + \alpha \mathbf{u}^*_{SLSQP}), \tag{3.8}$$

along the SLSQP descent direction at convergence, $\mathbf{u}^*_{SLSQP}$, over a small range of α.

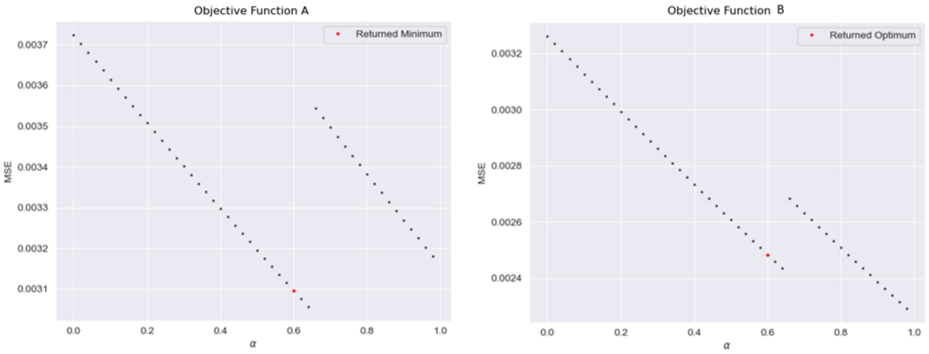


Figure 3.17. Line Searches at discontinuous local minimisers resulting in premature termination of SLSQP

SLSQP convergences to a discontinuity for both objective functions as sampling to the right of this discontinuity results in an increase in the MSE. Consequently, SLSQP identifies the discontinuity as a local minimum and terminates. However, the directional derivative remains negative to the left and right of the discontinuity. Therefore, gradient-only optimisers proceed past the discontuinuity unhindered until the directional derivative changes sign from negative to positive, indicating a NN-GPP along a descent direction.

Tables (3.2) and (3.3) show the detailed results of the optimisers for this two dimensional problem, with the error term being the norm of the difference between the returned optimum vector and the known optimum. The norm of the gradient vector is two orders of magnitude larger at $\mathbf{x}^*_{SLSQP}$ than at $\mathbf{x}^*_{GOSSA}$ and $\mathbf{x}^*_{MS}$. This demonstrates that the gradient information at these discontinuities is not impacted as severely as the function value information. This means that by locating NN-GPPs, $\mathbf{x}^*_{\mathbf{NNGPP}}$, instead of minimisers, $\mathbf{x}^*$, the optimisation algorithm is able to bypass the numerically induced discontinuities in the objective function. For this design problem, as the objective function is unimodal, the $\mathbf{x}^*_{\mathbf{N}}$ and $\mathbf{x}^*_{\mathbf{NNGPP}}$ optima agree exactly as the norm of the gradient vector at these positive gradient projection points are zero within a tolerance of 10^{-5}.

Table 3.2. Detailed objective function **A** Results for Two Variable Inverse Problem

Method	SLSQP				GOSSA				Modified Subgradient			
	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error
Start 1	14	23 / 14	2.62×10^{-3}	5.86	18	– / 27	2.38×10^{-5}	0.011	31	– / 31	2.38×10^{-5}	0.051
Start 2	20	22 / 20	3.24×10^{-5}	**0.016**	26	– / 33	4.13×10^{-6}	0.006	63	– / 63	2.20×10^{-5}	0.057
Start 3	20	25 / 20	8.53×10^{-5}	0.023	14	– / 19	2.53×10^{-5}	0.007	31	– / 31	2.17×10^{-5}	**0.051**
Start 4	18	30 / 18	3.13×10^{-3}	4.46	18	– / 26	9.43×10^{-6}	**0.003**	51	– / 51	2.48×10^{-5}	0.066
Start 5	10	11 / 10	4.66×10^{-3}	2.27	17	– / 27	4.67×10^{-6}	0.007	49	– / 49	2.19×10^{-5}	0.061

Table 3.3. Detailed objective function **B** Results for Two Variable Inverse Problem

Method	SLSQP				GOSSA				Modified Subgradient			
	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error
Start 1	10	20 / 10	1.42×10^{-3}	5.49	8	– / 10	2.77×10^{-6}	0.0012	20	– / 20	1.23×10^{-6}	0.002
Start 2	25	25 / 25	6.66×10^{-7}	**0.0017**	9	– / 14	2.80×10^{-6}	**0.001**	42	– / 42	2.41×10^{-6}	**0.0016**
Start 3	22	23 / 22	1.05×10^{-5}	0.028	8	– / 11	3.39×10^{-6}	0.0045	21	– / 21	1.89×10^{-6}	0.0048
Start 4	12	20 / 12	8.65×10^{-4}	2.3	7	– / 13	2.99×10^{-6}	0.0019	25	– / 25	2.85×10^{-6}	0.0032
Start 5	10	10 / 11	8.73×10^{-4}	2.17	6	– / 10	1.33×10^{-6}	0.0018	30	– / 30	3.45×10^{-6}	0.0024

3.5.3 Four and Eight Shape Design Variables

To further assess the ability of gradient-only optimisers, more complex problems are solved. Firstly, the problem is adapted to the Four Variable Inverse Problem (shown in Figure (2.1)). As before, the target curve is that of a fully symmetrical arch. The initial curves are shown in Figure (3.18) while the final optimised curves are shown in Figure (3.19).

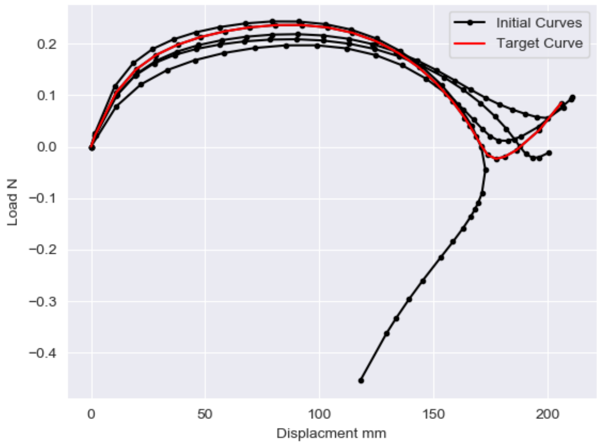


Figure 3.18. Initial Curves of Four Variable Optimisation Problem

The GOSSA and Modified Subgradient algorithms again find the global optimiser independent of the starting position in the domain and the implemented objective function. In contrast, the SLSQP algorithm struggled to find the optimum with none of the returned solutions fully tracing the target curve for either objective function, but objective function **B** again performs slightly better. Tables (3.4) and (3.5) again show detailed results of the three optimisers on the two objective functions. Although the GOSSA and Subgradient methods both returned the correct optimum curve, the increase in

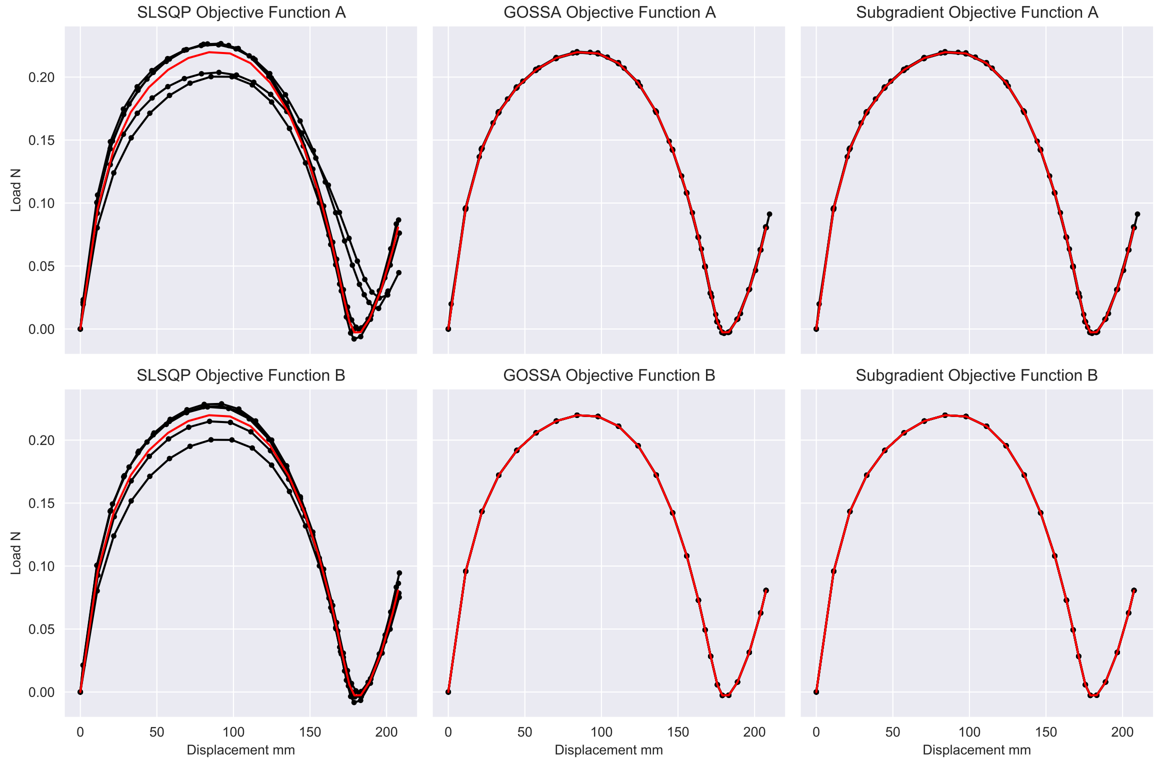


Figure 3.19. optimised Curves for the Three Algorithms and both objective functions for the Four Variable Inverse Problem

dimensionality begins to show the benefit in efficiency the GOSSA algorithm has compared to the Modified Subgradient method. The Modified Subgradient method took as many as three times the number of gradient computations to converge when compared to GOSSA. An indication that the SLSQP method again misinterpreted the discontinuities as minima is the large gradient norm at a discontinuous local minimiser when compared to the gradient norms at the GOSSA and Modified Subgradient optimisers.

Table 3.4. Detailed objective function **A** Results for Four Variable Inverse Problem

Method	SLSQP				GOSSA				Modified Subgradient			
	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error
Start 1	9	20 / 9	4.23×10^{-3}	8.17	22	– / 39	1.01×10^{-5}	1.34	38	– / 38	2.33×10^{-5}	1.29
Start 2	16	17 / 16	3.72×10^{-3}	7.86	34	– / 61	1.05×10^{-5}	2.58	189	– / 189	2.29×10^{-5}	4.95
Start 3	7	8 / 7	3.25×10^{-4}	9.53	20	– / 32	2.16×10^{-5}	3.15	94	– / 94	2.48×10^{-5}	3.04
Start 4	13	25 / 13	5.34×10^{-3}	8.84	20	– / 34	2.15×10^{-5}	**0.785**	78	– / 78	2.42×10^{-4}	**0.76**
Start 5	18	18 / 18	6.81×10^{-4}	**4.51**	26	– / 44	7.48×10^{-6}	3.01	71	– / 71	2.48×10^{-5}	3.02

To verify that the SLSQP algorithm terminates at a discontinuous local minimiser along the descent direction, Figure (3.20) presents a graph of the objective function values returned by a line search conducted at one of the $\mathbf{x}^*_{SLSQP}$ solutions. As with the Two Variable Inverse Problem, the SLSQP method mistakes a discontinuity in the objective function for a local minimum, and terminates prematurely. As before, the GOSSA and Modified Subgradient algorithms are able to bypass these discontinuities and continue to the optimum solution by only making use of the gradient information.

Table 3.5. Detailed objective function **B** Results for the Four Variable Inverse Problem

Method	SLSQP				GOSSA				Modified Subgradient			
	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error
Start 1	15	15 / 15	2.13×10^{-4}	**2.44**	30	– / 30	1.25×10^{-5}	1.34	35	– / 35	1.36×10^{-5}	1.45
Start 2	18	18 / 18	3.14×10^{-4}	7.85	52	– / 52	2.34×10^{-5}	2.01	167	– / 167	1.86×10^{-5}	3.78
Start 3	8	8 / 8	8.24×10^{-4}	9.57	26	– / 26	2.14×10^{-5}	2.54	90	– / 90	2.52×10^{-5}	3.01
Start 4	18	18 / 18	4.43×10^{-4}	3.51	28	– / 28	1.65×10^{-5}	**0.43**	75	– / 75	1.12×10^{-5}	**0.35**
Start 5	16	17 / 16	4.41×10^{-4}	4.62	20	– / 20	2.89×10^{-6}	3.15	65	– / 65	2.23×10^{-6}	2.98

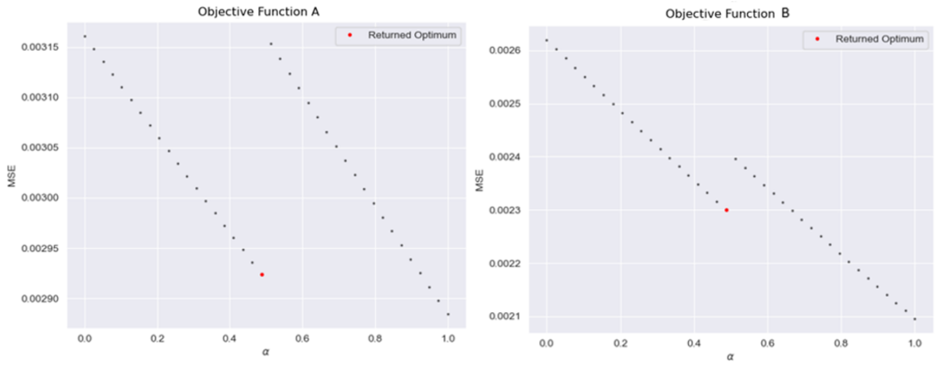


Figure 3.20. Line Searches at discontinuous local minimisers resulting in premature termination of SLSQP for the Four Variable Inverse Problem

The complexity of the problem is increased further to contain eight design variables (Figure (2.1)). The initial curves are shown in Figure (3.21), while the final optimised curves are shown in Figure (3.22) for the three optimisers.

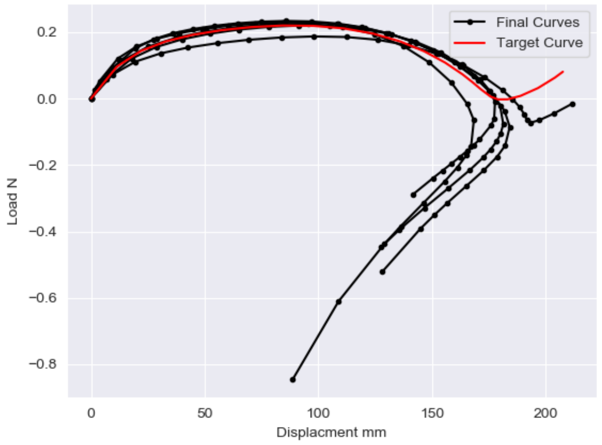


Figure 3.21. Initial Curves of Eight Variable Inverse Problem

The performance of the SLSQP algorithm, when using objective function **A**, deteriorates to such an extent that none of its optimised curves offer any reasonable similarity to the desired target curve. Objective function **B** performs better, with the majority of the solutions being similar to the target curve. The GOSSA and Modified Subgradient algorithms however remained reliable and still optimised fully

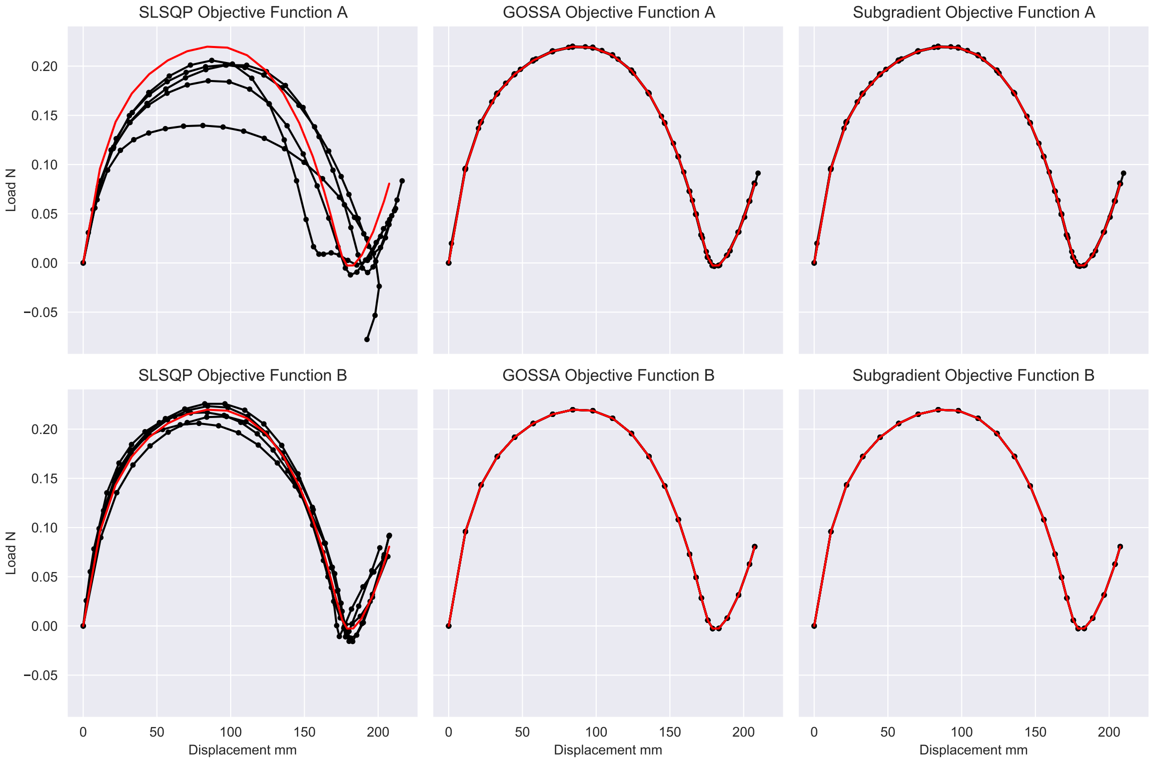


Figure 3.22. optimised Curves for the Three Algorithms and both objective functions for the Eight Variable Inverse Problem

to the desired target curve. Figure (3.23) illustrates some of the unsatisfactory final designs computed by the SLSQP algorithm.

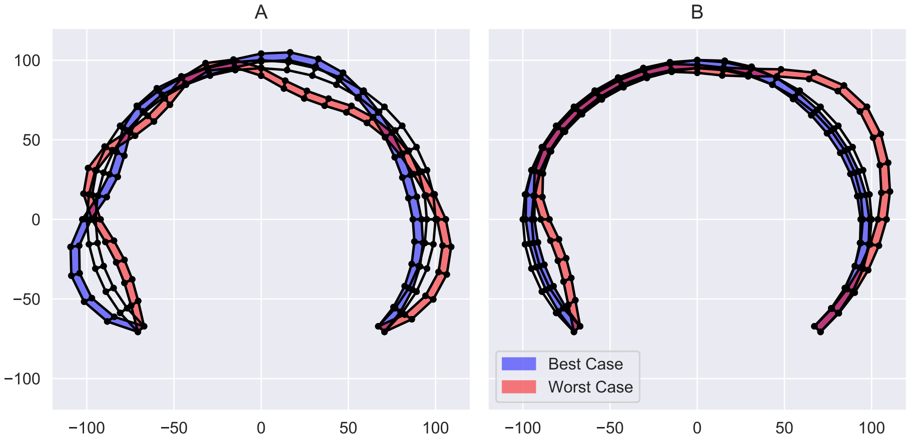


Figure 3.23. optimised Shapes returned by the SLSQP Algorithm for the Eight Variable Inverse Problem Overlaid with the Global Optimum. The best and worst results are indicated in blue and red respectively, using Objective Function **A** and Objective Function **B**.

Tables (3.6) and (3.7) show the detailed results for the implemented optimisers. The efficiency of the GOSSA algorithm is apparent at this level of dimensionality, as it consistently requires six to ten times fewer simulations than the Modified Subgradient method to converge. Again, by simply inspecting the gradient norm at the returned optimums, it becomes apparent that the SLSQP algorithm struggled to bypass the discontinuities in the objective function.

Table 3.6. Detailed objective function **A** Results for Eight Variable Inverse Problem

Method	SLSQP				GOSSA				Modified Subgradient			
	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error	Iterations	Evaluations (Func/Jac)	Gradient Norm	Error	Iterations	Evaluations (Func/Jac)	Gradient Norm	Error
Start 1	23	42 / 23	4.10×10^{-2}	25.93	95	– / 150	2.25×10^{-5}	10.62	359	– / 359	1.24×10^{-5}	4.61
Start 2	16	19 / 16	1.62×10^{-2}	20.81	53	– / 92	1.07×10^{-5}	12.72	279	– / 279	4.99×10^{-5}	13.61
Start 3	11	22 / 11	3.64×10^{-2}	47.53	16	– / 29	5.61×10^{-5}	18.91	728	– / 728	5.44×10^{-5}	17.31
Start 4	18	30 / 18	2.00×10^{-2}	16.70	37	– / 59	2.11×10^{-5}	**3.93**	221	– / 221	4.99×10^{-5}	**3.89**
Start 5	16	46 / 16	2.81×10^{-2}	**13.19**	64	– / 111	1.56×10^{-5}	4.96	241	– / 241	4.99×10^{-5}	5.16

Table 3.7. Detailed objective function **B** Results for Eight Variable Inverse Problem

Method	SLSQP				GOSSA				Modified Subgradient			
	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error	# Iter.	Evaluations (Func/Jac)	Gradient Norm	Error
Start 1	41	51 / 41	4.31×10^{-4}	22.35	80	– / 80	1.89×10^{-5}	9.86	301	– / 301	1.45×10^{-5}	6.81
Start 2	35	36 / 35	4.51×10^{-4}	14.21	45	– / 45	2.56×10^{-6}	13.21	254	– / 254	2.71×10^{-6}	12.34
Start 3	27	35 / 27	3.56×10^{-4}	28.98	20	– / 20	3.54×10^{-5}	17.41	689	– / 689	3.78×10^{-5}	16.31
Start 4	34	38 / 34	2.21×10^{-4}	**5.06**	30	– / 30	1.34×10^{-5}	**3.54**	214	– / 214	1.93×10^{-5}	**3.84**
Start 5	20	21 / 20	3.48×10^{-4}	10.11	54	– / 54	2.84×10^{-5}	4.84	231	– / 231	2.46×10^{-5}	4.96

Figure (3.24) again shows a line search at a local minimiser $\mathbf{x}^*_{SLSQP}$ along the descent direction for SLSQP. As the complexity and dimensionality of the design problem increased, the size and number of the discontinuities increased to the point where the SLSQP algorithm can no longer offer reasonable performance. The gradient-only optimisers isolating NN-GPPs remained reliable as the complexity of the problem increased. The number or size of the discontinuities did not affect the quality of the results.

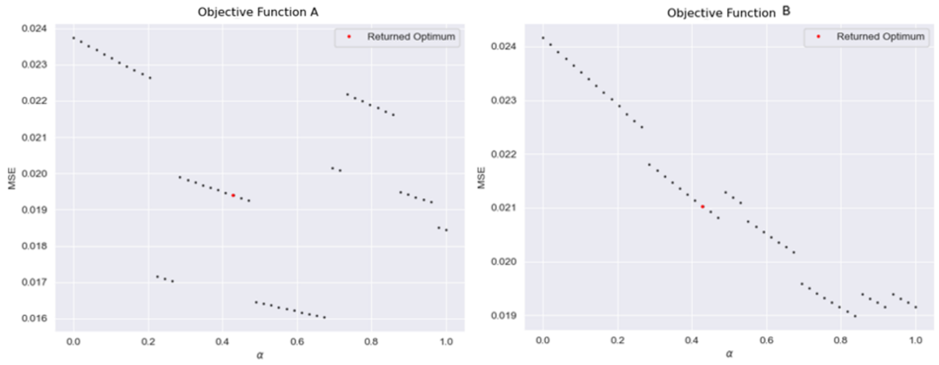


Figure 3.24. Line Searches for Eight Variable Inverse Problem at discontinuous local minimisers resulting in premature termination of SLSQP

.

3.6 Chapter Conclusion

The results in this chapter show that an optimisation problem that requires the ALC algorithm when computing the objective function, will exhibit discontinuities in the objective function if automatic arc length adjustment takes place during the simulation. Instead of deactivating automatic arc length control, it is retained in this work since it ensures that the algorithm is computationally efficient, and that the load-deflection path is computable for all designs. However, the resulting discontinuities are purely numerical artefacts, and the chosen optimisation algorithms should ignore them when searching for the optimal solutions.

The presence of discontinuities also implicitly limits the use of numerical approximations (such as forward finite difference) to compute the gradient of the objective function, as such methods can return incorrect results if the calculation takes place across one of these discontinuities. Although the complex-step method might offer an alternative to compute sensitivities, an analytical sensitivity procedure is implemented to calculate the gradient of the objective function that remains efficient as the dimensionality increases.

As SLSQP aims to locate function minimisers $\mathbf{x}^*$, it is prone to terminate prematurely at discontinuities in the objective function, that presents as local minimisers along the search directions. This renders classical gradient-based algorithms vulnerable. In turn, gradient-only optimisers that aim to isolate NN-GPPs are able to reliably locate the global optimiser, making it computationally efficient for high-dimensional problems. In particular, GOSSA proved more efficient than the Modified Subgradient method, and also does not require any hyper-parameter tuning.

The results in this chapter provide compelling evidence that it is possible to design snap-through structures that match the desired load-deflection path without limiting the complexity of both the desired and simulated load-deflection paths. Essential features of the proposed approach are automatic arc length adjustment to ensure efficient analysis, combined with gradient-only optimisation algorithms that reliably locate NN-GPPs in the design space.

A question that arises naturally at this point is if the optimisation procedure can be accelerated. One option is to replace the expensive FE simulations with accurate approximations, so-called surrogate models. These models can be trained with data obtained through simulations completed in parallel, thereby eliminating the computationally expensive sequential nature of the direct simulation in the loop strategy completed in this chapter. The next chapter is dedicated to accurate construction of surrogates, including the possible benefits of using the available analytical sensitivity information.

Chapter 4 A Novel Coordinate System Transformation for Isotropic Kernel-Based Surrogate Models

4.1 Chapter Abstract

To complete the surrogate-based optimisation strategy the performance of commonly used models must first be investigated. Specifically, as the shape optimisation problem in this study can be highly-nonlinear, it is unlikely that the defined shape parameters will have similar importance or impact on the load path the structure will exhibit.

Therefore, this chapter develops a novel coordinate system transformation scheme to improve the performance of common radial basis function surrogate models [34]. This coordinate system transformation scheme is based on the fact that commonly used basis functions are isotropic, and that underlying functions in typical engineering problems can contain anisotropic data manifolds.

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 chapter 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 chapter 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.

4.2 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 (4.1).

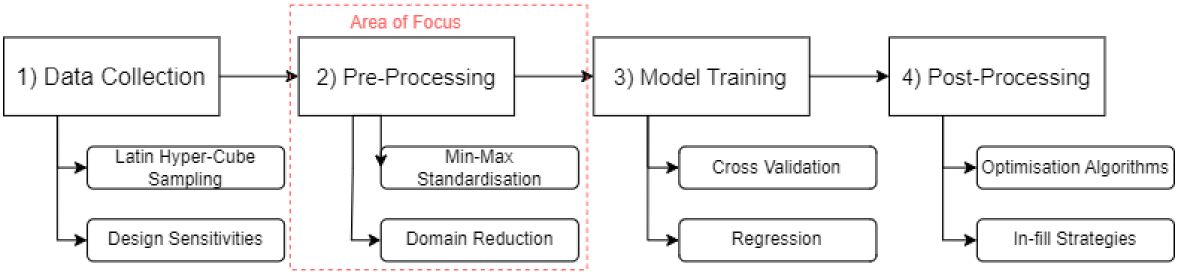


Figure 4.1. Flow diagram showing the main areas of research in the context of surrogate-based optimisation (SBO) as well as some of the main techniques used in each step.

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 [9,35–37], component scaling methods [38,39], the regression of high fidelity and low fidelity information [7,40], and the inclusion of gradient information into the model [35,41]. 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 [35,42].

The work in this chapter 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 [9,36]. 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.

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 [7,38]. 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 (4.2) 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 chapter 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 chapter then

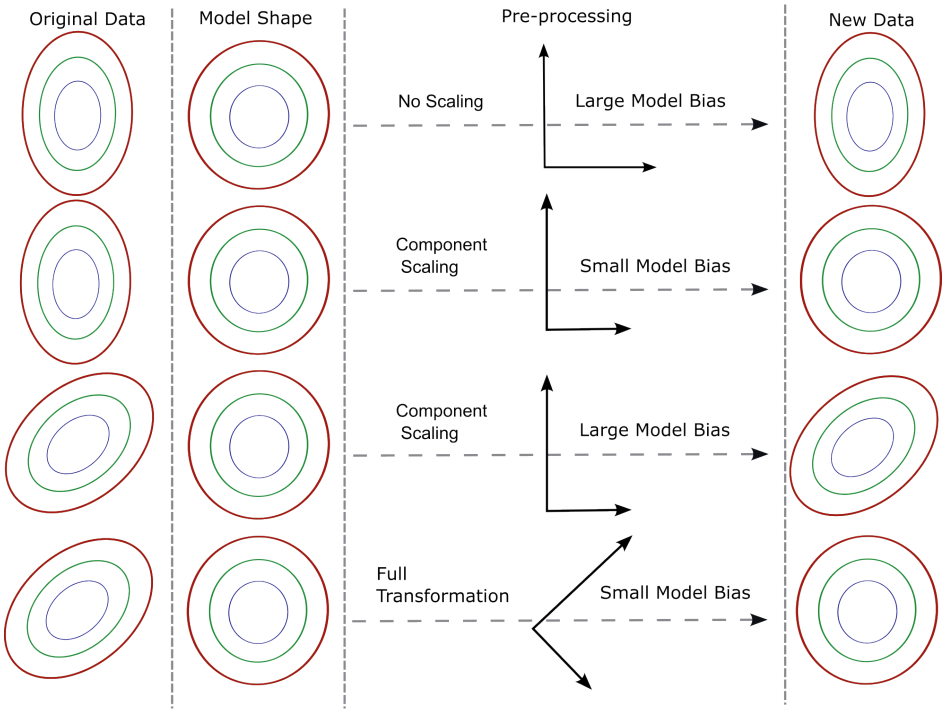


Figure 4.2. Visual demonstration of how different pre-processing strategies applied to different datasets that exhibit behaviour ranging from uncoupled to scaled and coupled, can lessen the model bias in basis function surrogate models.

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 chapter 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 chapter. The proposed transformation scheme is then derived in Section (4.5) and a test problem and its important characteristics are discussed in Section (4.6). Results are then generated in Section (4.7) where the proposed transformation scheme is compared to standard and more modern approaches. Section (4.8) demonstrates the limitations of the proposed method in the case of non-decomposable functions. Lastly, some conclusion and recommendations are offered in Section (4.9).

4.3 Related Work

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

Although the other areas of research shown in Figure (4.1) 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.

4.3.1 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 [43].

During the sampling phase it is also possible to obtain gradient information at the sampled locations in the design coordinate system. Although many papers [6–9] assume that in this scenario gradient information of the function is not available, it is often not the case. Many papers [19, 20, 22, 23] 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 [44]. 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 [19, 22, 45–47]. Therefore, in this work sampling scenarios with and without gradient information are considered.

4.3.2 Common Surrogate Models

Surrogate models can be classified into function-value based, gradient-enhanced, and gradient-only [48]. Note that surrogate models that regresses through both function value and gradient information are referred to as either gradient-enhanced (GE) models [35, 42], cooperative models (CO) [49, 50], or first order (FO) models [48, 51]. 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 [6, 9, 35–38]. In this chapter the function value (FV-RBF) and gradient enhanced RBF (GE-RBF) models are implemented.

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

where the variable ε is referred to as the shape parameter and the point $\mathbf{c}$ 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

$$f_{\text{RBF}} = \sum_{i=1}^{k} w_i \phi_i(\mathbf{x},\mathbf{c}_i,\varepsilon). \tag{4.1}$$

This equation becomes a system of equations

$$\mathbf{f} = \mathbf{\Phi}(\mathbf{x},\mathbf{c},\varepsilon)\mathbf{w}, \tag{4.2}$$

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

$$\mathbf{\Phi} = \begin{bmatrix} \phi(\mathbf{x}_1,\mathbf{c}_1,\varepsilon) & \phi(\mathbf{x}_1,\mathbf{c}_2,\varepsilon) & \dots & \phi(\mathbf{x}_1,\mathbf{c}_k,\varepsilon) \\ \phi(\mathbf{x}_2,\mathbf{c}_1,\varepsilon) & \phi(\mathbf{x}_2,\mathbf{c}_2,\varepsilon) & \dots & \phi(\mathbf{x}_2,\mathbf{c}_k,\varepsilon) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(\mathbf{x}_p,\mathbf{c}_1,\varepsilon) & \phi(\mathbf{x}_p,\mathbf{c}_2,\varepsilon) & \dots & \phi(\mathbf{x}_p,\mathbf{c}_k,\varepsilon) \end{bmatrix}_{k\times p} . \tag{4.3}$$

The remaining parameters of the surrogate include the number and locations of the centres $\mathbf{c}$ 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 $\mathbf{\Phi}$ becomes square and the weight vector can be solved directly from Equation (4.2). This is the method implemented for this research.

Some research, for example, [52] 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

$$\mathbf{\Phi}^T\mathbf{f} = \mathbf{\Phi}^T\mathbf{\Phi}\mathbf{w}, \tag{4.4}$$

must be implemented.

4.3.2.1 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

$$\frac{d\phi(\mathbf{x},\mathbf{c},\varepsilon)}{d\mathbf{x}} = -2\varepsilon\phi(\mathbf{x},\mathbf{c},\varepsilon)(\mathbf{x}-\mathbf{c}), \tag{4.5}$$

where Equation (4.5) 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

$$\begin{bmatrix} \frac{df_1}{d\mathbf{x}} \\ \frac{df_2}{d\mathbf{x}} \\ \vdots \\ \frac{df_p}{d\mathbf{x}} \end{bmatrix} = \begin{bmatrix} \frac{d\phi(\mathbf{x}_1,\mathbf{c}_1,\varepsilon)}{d\mathbf{x}} & \frac{d\phi(\mathbf{x}_1,\mathbf{c}_2,\varepsilon)}{d\mathbf{x}} & \dots & \frac{d\phi(\mathbf{x}_1,\mathbf{c}_k,\varepsilon)}{d\mathbf{x}} \\ \frac{d\phi(\mathbf{x}_2,\mathbf{c}_1,\varepsilon)}{d\mathbf{x}} & \frac{d\phi(\mathbf{x}_2,\mathbf{c}_2,\varepsilon)}{d\mathbf{x}} & \dots & \frac{d\phi(\mathbf{x}_2,\mathbf{c}_k,\varepsilon)}{d\mathbf{x}} \\ \vdots & \vdots & \vdots & \\ \frac{d\phi(\mathbf{x}_p,\mathbf{c}_1,\varepsilon)}{d\mathbf{x}} & \frac{d\phi(\mathbf{x}_p,\mathbf{c}_2,\varepsilon)}{d\mathbf{x}} & \dots & \frac{d\phi(\mathbf{x}_p,\mathbf{c}_k,\varepsilon)}{d\mathbf{x}} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix}. \tag{4.6}$$

The system of Equations (4.6), can then be written as

$$\nabla\mathbf{f} = \mathbf{\Phi}_{fo}\mathbf{w}_{fo}. \tag{4.7}$$

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

$$\begin{bmatrix} \mathbf{f} \\ \nabla\mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} \\ \mathbf{\Phi}_{fo} \end{bmatrix} \mathbf{w}_{GE}. \tag{4.8}$$

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, $\mathbf{w}_{GE}$, 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 (4.4)).

4.3.3 The Isotropic Assumption

The isotropic behaviour of RBF surrogate models arise 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: $\phi(\mathbf{x},\mathbf{c},\varepsilon) = e^{-\varepsilon||\mathbf{x}-\mathbf{c}||^2}$,
2. Inverse quadratic: $\phi(\mathbf{x},\mathbf{c},\varepsilon) = \frac{1}{\sqrt{||\mathbf{x}-\mathbf{c}||+\varepsilon^2}}$,
3. Multi-quadratic: $\phi(\mathbf{x},\mathbf{c},\varepsilon) = \frac{1}{1+\varepsilon||\mathbf{x}-\mathbf{c}||}$,

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 (4.3).

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 [6, 53]. 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

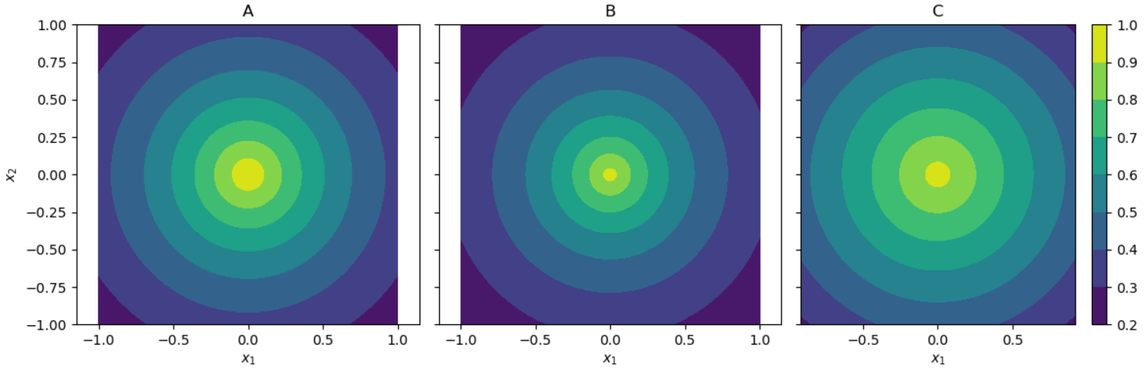


Figure 4.3. Contour plots of the Gaussian, Inverse Quadratic, and Multi-quadratic basis function with $\varepsilon = 1$ in Subfigures A, B, and C respectively.

error is selected. The basis function amplitudes are computed using the least squres formulation in Equation (4.4).

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:

$$\frac{d^2\phi(\mathbf{x},\mathbf{c},\varepsilon)}{d\mathbf{x}^2} = -2\varepsilon\frac{d\phi}{d\mathbf{x}}(\mathbf{x}-\mathbf{c})^T - 2\varepsilon\mathbf{I}\phi(\mathbf{x}). \tag{4.9}$$

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

$$\left.\frac{d^2\phi(\mathbf{x},\mathbf{c},\varepsilon)}{d\mathbf{x}^2}\right|_{\mathbf{x}=\mathbf{c}} = -2\varepsilon\mathbf{I}, \tag{4.10}$$

where $\mathbf{I}$ is an identity matrix. Therefore, the act of altering or optimising the shape parameter therefore 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.

4.3.4 Anisotropic Scaling Strategies

There are many cross-validation strategies in literature that attempt to alleviate the negative consequences of the implicit isotropic assumption and find appropriate hyper-parameters. 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 [38, 39],
- adapting the basis function to explicitly handle anisotropic functions by using a shape parameter for each principal direction in the design coordinate system [9, 35–37],
- or more recently, implementing the so-called Active Subspace Method (ASM) [54–57].

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) [36]. In higher dimensions, this becomes computationally expensive, so much so that it can become the bottleneck in computation time for SBO. Toal *et al.* [36] 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.* [35, 37] 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) [35], and in the case where the variables are coupled (see Figure (4.2)), 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.* [36], 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.

4.3.5 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 [54].

This method is described in detail by Constantine *et al.* [54], 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

$$\mathbf{C} = \mathbb{E}(\nabla f(\mathbf{x}) \nabla f(\mathbf{x})^T), \tag{4.11}$$

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

$$\mathbf{C} = \tilde{\mathbf{C}} = \frac{1}{p} \sum_{i}^{p} \nabla f(\mathbf{x}_i) \nabla f(\mathbf{x}_i)^T, \tag{4.12}$$

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

$$\tilde{\mathbf{C}} = \mathbf{V} \mathbf{\Sigma} \mathbf{V}^T, \tag{4.13}$$

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

$$\mathbf{\Sigma} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}, \tag{4.14}$$

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 [57,58], but it is included here as proposed in the original formulation [54]. The ASM method can then be summarised as

1. Compute $\tilde{\mathbf{C}}$ using Equation (4.12) from the sampled data set.
2. Decompose $\tilde{\mathbf{C}}$ into its eigenvalues and eigenvectors $\mathbf{V}$ and $\mathbf{\Sigma}$ respectively.
3. Rotate the coordinate system using the eigenvectors $\mathbf{V}$ and then scale each direction i with $\sqrt{\lambda_i}$.

4.4 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:

$$F(\mathbf{x}) = \sin(2\pi x_1) + \sin(2\pi x_2). \tag{4.15}$$

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 $\mathbf{x}^*$ is defined in which the inputs of the function are scaled using the equation

$$\mathbf{x}^* = \mathbf{S}\mathbf{x}, \tag{4.16}$$

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

$$\mathbf{S} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}. \tag{4.17}$$

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

$$\hat{\mathbf{x}} = \mathbf{R}\mathbf{x}^* = \mathbf{R}\mathbf{S}\mathbf{x} \tag{4.18}$$

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

$$\mathbf{R} = \begin{bmatrix} \cos(30^\circ) & -\sin(30^\circ) \\ \sin(30^\circ) & \cos(30^\circ) \end{bmatrix}. \tag{4.19}$$

The functions in these three coordinate system, namely the “original”, “scaled”, and “scaled and rotated” coordinate systems are shown in Figure (4.4).

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

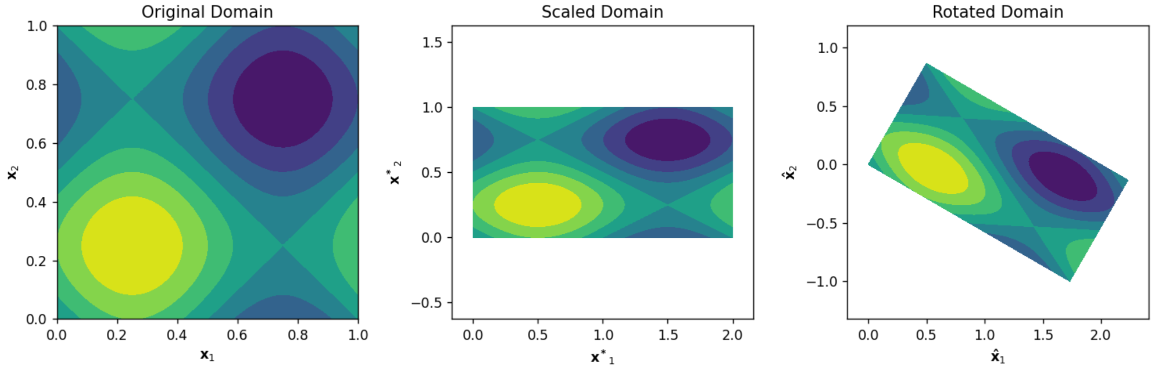


Figure 4.4. Contour plots of the example function illustrated in the "original" coordinate system, "scaled" coordinate system and "scaled and rotated" coordinate system.

measure is an 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

$$\text{RMSE} = \sqrt{\frac{\sum_i^N (V_T^i - V_P^i)^2}{N}} \tag{4.20}$$

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 (4.5) 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. $\mathbf{x}^*$ and $\hat{\mathbf{x}}$, 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 $\hat{\mathbf{x}}$ coordinate system to the original $\mathbf{x}$ 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 (4.6).

4.5 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 (4.3.3) and (4.4) 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

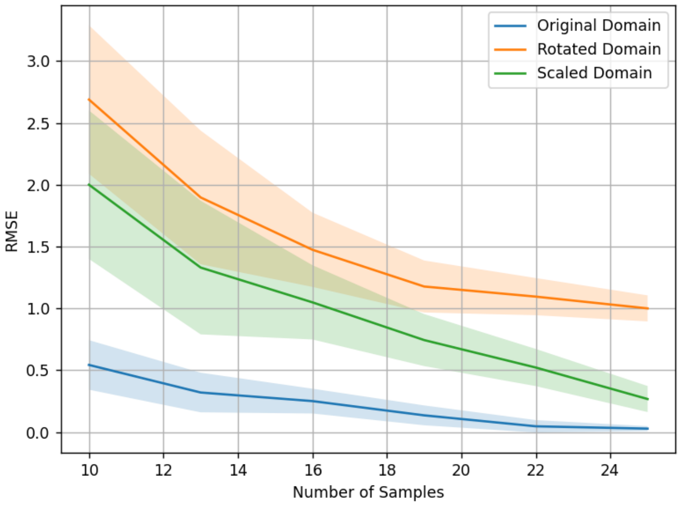


Figure 4.5. The mean RMSE (solid lines) and the mean variance (shaded regions) in the RMSE for the surrogates constructed in the three coordinate systems for an increasing number of samples. Means are computed across 50 instances.

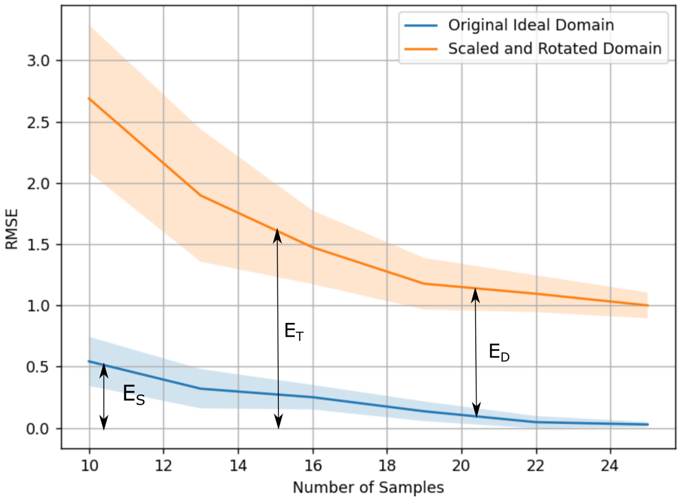


Figure 4.6. The sources of poor performance of a surrogate. The total error E_T consists of the sparsity of information error E_S and the construction coordinate system error E_D.

scheme, and that this curvature information needs to be obtained from a collection of *local* estimations and not a single *global* estimation.

4.5.1 Second-Order Non-linearity

To begin this argument consider a simple quadratic function given by

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathsf{T}}\mathbf{A}\mathbf{x}, \tag{4.21}$$

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

$$\mathbf{A}_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{A}_2 = \begin{bmatrix} 3 & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{A}_3 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}. \tag{4.22}$$

Figure (4.7) depicts the contour plots for these three cases. The dashed lines in these figures indicate the eigenvalues and eigenvectors of the $\mathbf{A}$ matrices.

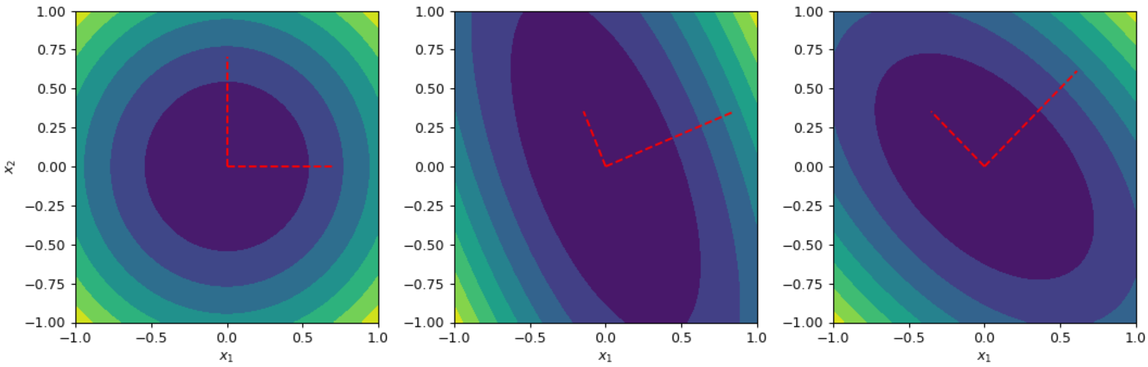


Figure 4.7. Contour plots of quadratic functions with Hessians given by $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$. The dashed lines indicate the eigenvectors and their lengths are chosen proportional to the eigenvalues.

The shape of the function in the case of $\mathbf{A}_1$, 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 $\mathbf{A}$, the function evaluated in the transformed coordinate system should resemble the case where $\mathbf{A} = \mathbf{A}_1$.

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 (4.8) shows the contours using this transformation scheme, for the 3 different $\mathbf{A}$ matrices. Clearly, by taking into account the curvature in all directions the problem can be recast into a coordinate system where the function is isotropic.

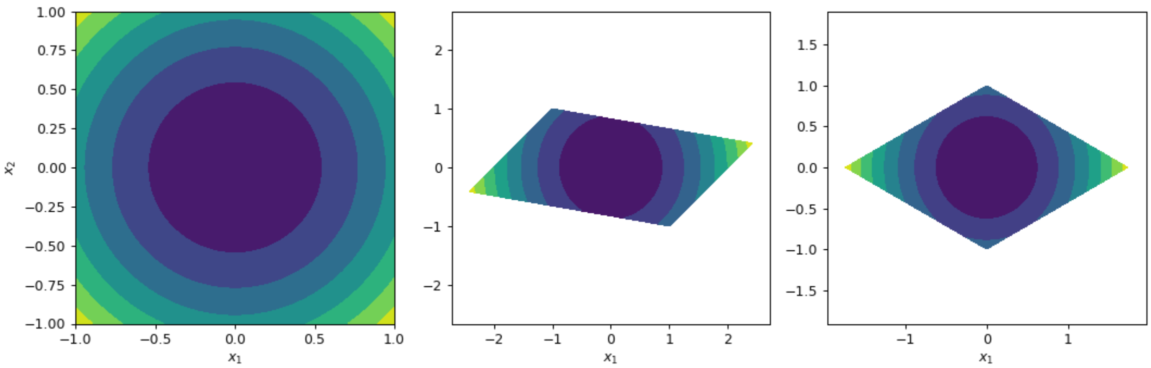


Figure 4.8. Contour plots of quadratic functions with Hessians given by $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$, after applying the proposed transformation scheme to the original coordinate system.

4.5.2 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*, meaning at the location of the basis function, the underlying function is anisotropic 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 functions 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. Firstly, Consider the 1D function

$$F(x) = \sin(f\pi x) + 15(x-0.5)^2, \tag{4.23}$$

where f is 6 and 12 for two example problems depicted in Figure (4.9). 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 the functions are scaled such that the global curvature estimates become similar, no scaling will be required as the global estimates both return a similar curvature estimations. However, if the functions are scaled 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.

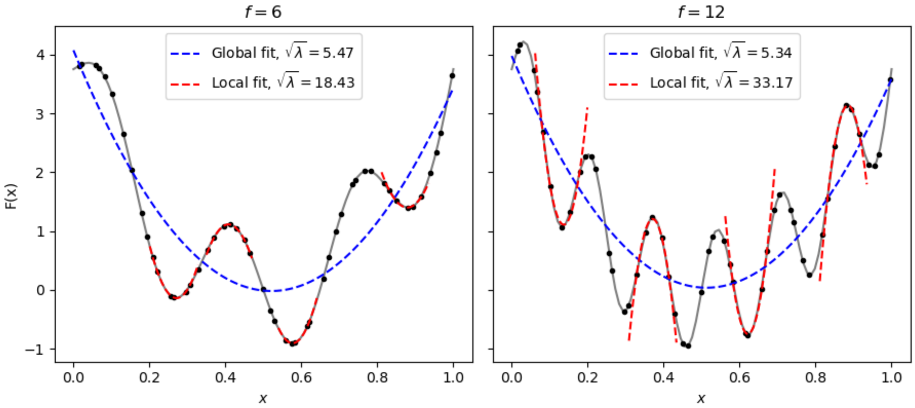


Figure 4.9. Two 1D example problems illustrating the difference between a single global fit and many local fits. The average optimum scaling factor proposed by both estimations, $\sqrt{\lambda}$, for the local and global fits are also denoted on the figure.

The shortcomings of a single global approximation method is also demonstrated on the 2D problem from Equation (4.15), in the scaled and rotated coordinate system using 25 samples. The function is approximated using a full $n-$dimensional quadratic fit of the form

$$\mathbf{f} = \sum_{i}^{n}\sum_{j}^{n} w_{ij}x_i x_j + \sum_{k}^{n} w_k x_k + w_c, \tag{4.24}$$

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

$$\mathbf{H} = \begin{bmatrix} 2w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & 2w_{22} & \dots & w_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & \dots & 2w_{nn} \end{bmatrix}, \tag{4.25}$$

where $w_{12} = w_{21}$ as the matrix is symmetric. In the implementation of this chapter, 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 (4.10) 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.

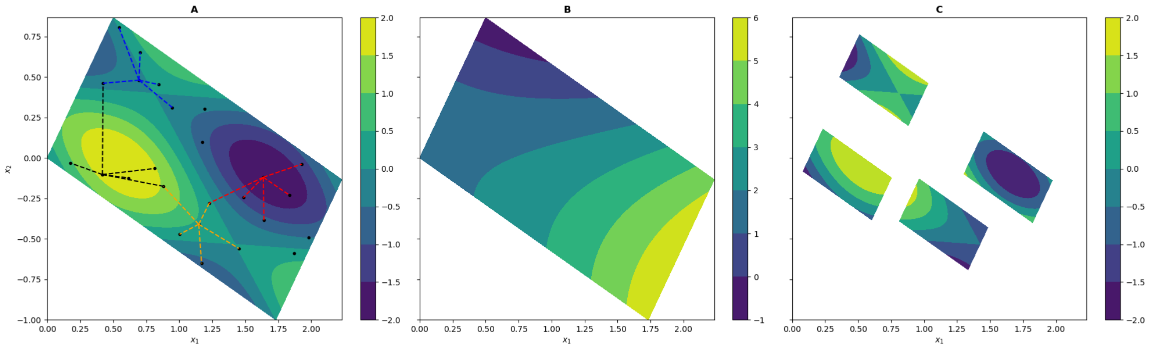


Figure 4.10. Contour plots of the original function, the global quadratic fit and local fits in subplots **A**, **B** and **C** respectively. The local cluster of points used for each local fit are shown by the coloured dashed lines.

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 [59]. 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 [60]

$$\mathbf{H} = \mathbf{V}\mathbf{\Sigma}\mathbf{V}^T, \tag{4.26}$$

where $\mathbf{V}$ and $\mathbf{\Sigma}$ 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,

$$\mathbf{H}_{\text{rec}} = \mathbf{V}|\mathbf{\Sigma}|\mathbf{V}^T. \tag{4.27}$$

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

$$\mathbf{H}_{\text{avg}} = \frac{1}{N}\sum_{i}^{N}\mathbf{H}_{\text{rec}}. \tag{4.28}$$

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 (4.9). 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, meaning with positive eigenvalues, such that only the *magnitude* and *direction* of the local curvature is kept.

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.

4.5.3 Hessian Estimation

For the research completed in this chapter 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 (4.5.2) are implemented. In the case where gradient information is available, the Symmetric Rank 1 (SR1) Hessian update method [48] is used:

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{(\mathbf{y}_k - \mathbf{H}_k\Delta\mathbf{x}_k)(\mathbf{y}_k - \mathbf{H}_k\Delta\mathbf{x}_k)^T}{(\mathbf{y}_k - \mathbf{H}_k\Delta\mathbf{x}_k)^T\Delta\mathbf{x}_k}. \tag{4.29}$$

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

$$\mathbf{y}_k = \nabla F(\mathbf{x}_k + \Delta\mathbf{x}_k) - \nabla F(\mathbf{x}_k). \tag{4.30}$$

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).

4.5.4 Numerical Transformation Example

To demonstrate the proposed method Figure (4.11) shows contour plots of Equation (4.15) 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 (4.4)). Figure (4.11) demonstrates that for this example problem the GE-LHM quickly converges to the original ideal coordinate system, while the FV-LHM and ASM will need additional samples.

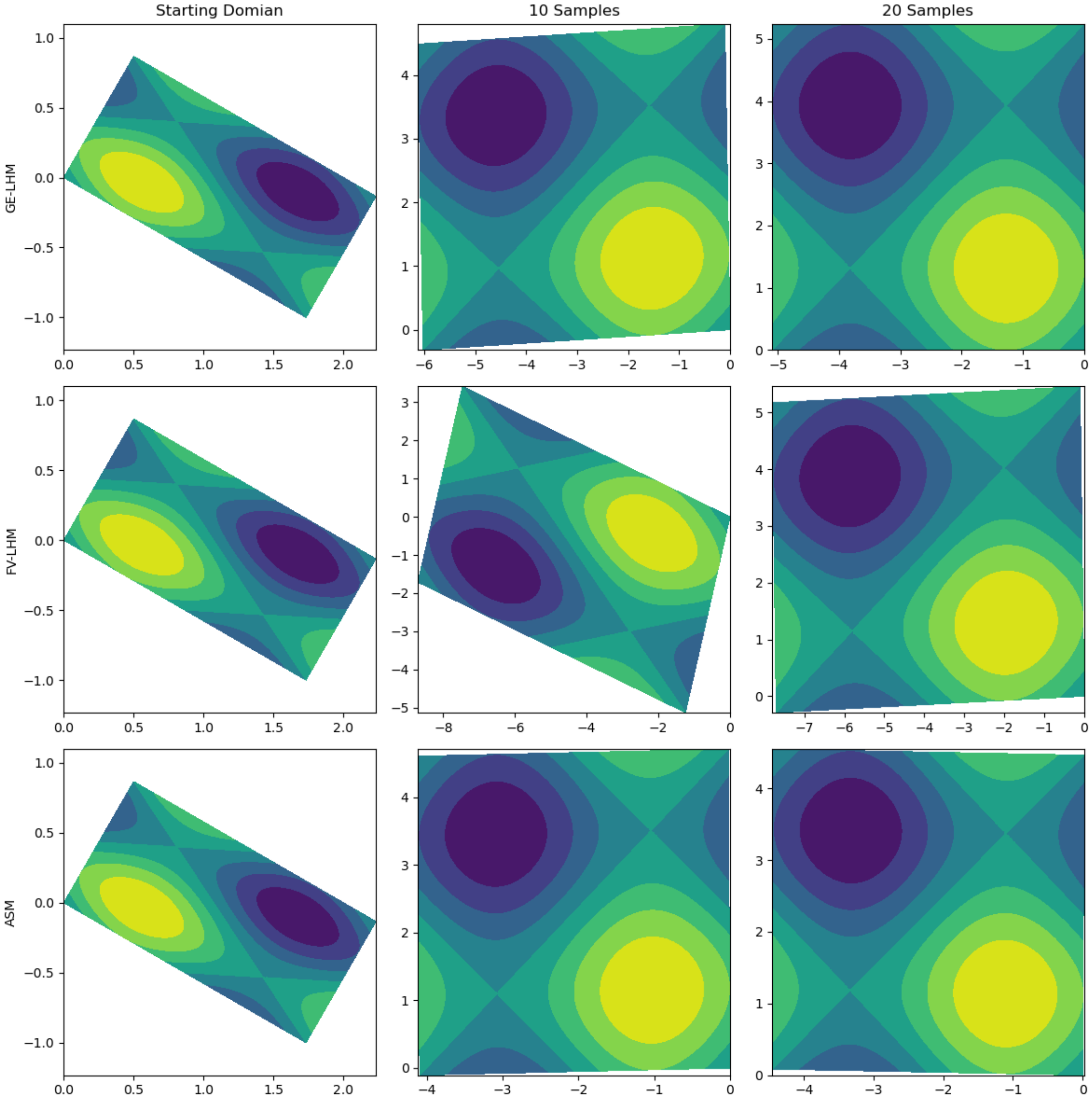


Figure 4.11. Contour plots of Equation (4.15) in the scaled and rotated coordinate systems, a transformed coordinate system using 10 samples, and a transformed coordinate system using 20 samples.

4.5.5 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

$$\mathbf{F}(\hat{\mathbf{x}}) = \mathbf{F}(\hat{\mathbf{x}}(\mathbf{x})), \tag{4.31}$$

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

Using the chain rule and Equation (4.31), the function gradient can be expressed as

$$\frac{d\mathbf{F}}{d\mathbf{x}} = \frac{d\mathbf{F}}{d\hat{\mathbf{x}}}\frac{d\hat{\mathbf{x}}}{d\mathbf{x}}, \tag{4.32}$$

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

$$\frac{d\mathbf{F}}{d\hat{\mathbf{x}}} = \frac{d\mathbf{F}}{d\mathbf{x}}\left(\frac{d\hat{\mathbf{x}}}{d\mathbf{x}}\right)^{-1}, \tag{4.33}$$

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

$$\hat{\mathbf{x}} = \mathbf{RSx}. \tag{4.34}$$

Taking the gradient of Equation (4.34) yields

$$\frac{d\hat{\mathbf{x}}}{d\mathbf{x}} = \mathbf{RS}. \tag{4.35}$$

Since $\mathbf{R}$ is a orthogonal matrix, $\mathbf{R}^{-1} = \mathbf{R}^{\mathsf{T}}$. Therefore,

$$\left(\frac{d\hat{\mathbf{x}}}{d\mathbf{x}}\right)^{-1} = (\mathbf{RS})^{-1} = \mathbf{S}^{-1}\mathbf{R}^{-1} = \mathbf{S}^{-1}\mathbf{R}^{\mathsf{T}}. \tag{4.36}$$

Since the scaling matrix $\mathbf{S}$ 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 (4.33) then becomes

$$\frac{d\mathbf{F}}{d\hat{\mathbf{x}}} = \frac{d\mathbf{F}}{d\mathbf{x}}\mathbf{S}^{-1}\mathbf{R}^{\mathsf{T}}. \tag{4.37}$$

4.5.6 Summary of Proposed Transformation Procedure

The implementation of the proposed transformation procedure can be separated into three Sub-procedures. The first Sub-procedure, Sub-procedure (3), iterates through all the sampled points and calculates an average Hessian estimation.

Sub-procedure 3: Transformation Procedure

```
Input  : Sampled Information of the Underlying Function.
Output : The transformed coordinate system $\hat{\mathbf{x}}$ and the gradients $\frac{d\mathbf{F}}{d\hat{\mathbf{x}}}$
for All sampled points do
    if Gradient Information is available then
        Use Procedure (4)
    else
        Use Procedure (5)
    end
end
for All Hessian Estimations do
    Compute Equation (4.27)
end
Compute Equation (4.28);
Find the eigenvalues and eigenvectors of the average Hessian;
if Gradient Information is available then
    Compute Equation (4.37);
end
Compute Equation (4.34) using the eigenvalues and eigenvectors;
Return $\hat{\mathbf{x}}$, and $\frac{d\mathbf{F}}{d\hat{\mathbf{x}}}$
```

Sub-procedures (4) and (5) compute local Hessian estimations from some subset of points in the sample set.

Sub-procedure 4: Gradient Information Based Hessian Estimation

```
Input  : A sampled point
Output : A Local Hessian Estimation
Find the $n+1$ closest points;
Initialise $\mathbf{H_0}$ as an Identity matrix ;
Arrange from furthest to closest;
for Closest Points Subset do
    Compute Equation (4.29)
end
Return The local Hessian Estimation.
```

Sub-procedure 5: Function Information Based Hessian Estimation

```
Input  : A sampled point
Output : A Local Hessian Estimation
Find the $n(n-1)/2+n+1$ closest points;
Fit local Quadratic function using Equation (4.24) ;
Rearrange weight vector into Hessian using Equation (4.25);
Return The local Hessian Estimation.
```

4.6 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 (4.4) to a more general form where the dimensionality of the problem can be altered.

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

$$f(\mathbf{x}) = f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n), \tag{4.38}$$

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 the test function is deliberately selected as a decomposable function, ensuring that the optimal reference frame in which to express the function is known. The remaining feature that is deliberately embed into the test function, is varying length scales in different coordinate directions. This results in a test function for which certain characteristics can be easily altered, 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

$$f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{N} A_i \sin(F_i x_i), \tag{4.39}$$

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

$$A_i = -2\exp\frac{-(2i-N)^2}{N} + 3, \tag{4.40}$$

$$F_i = \frac{3\pi}{2 + 2\exp\frac{-20i+N}{2}} + \frac{\pi}{2}. \tag{4.41}$$

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 **R** created from

$$\mathbf{R} = \text{expm}(\pi(\mathbf{A} - \mathbf{A}^\intercal)), \tag{4.42}$$

where **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 $(\mathbf{A} - \mathbf{A}^\intercal)$ results in an orthogonal matrix [48].

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 (4.39) is simply

$$\frac{\partial F_i}{\partial x_i} = \frac{1}{n} A_i F_i \cos(F_i x_i), \tag{4.43}$$

where in the case of coordinate system rotation, the process detailed in Section (4.5.5) is followed.

4.7 RMSE Results

The numerical results 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 (4.3.4),
- 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 an analytical expression for the underlying function, where the optimal rotation matrix **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 (4.12) 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 GE 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.

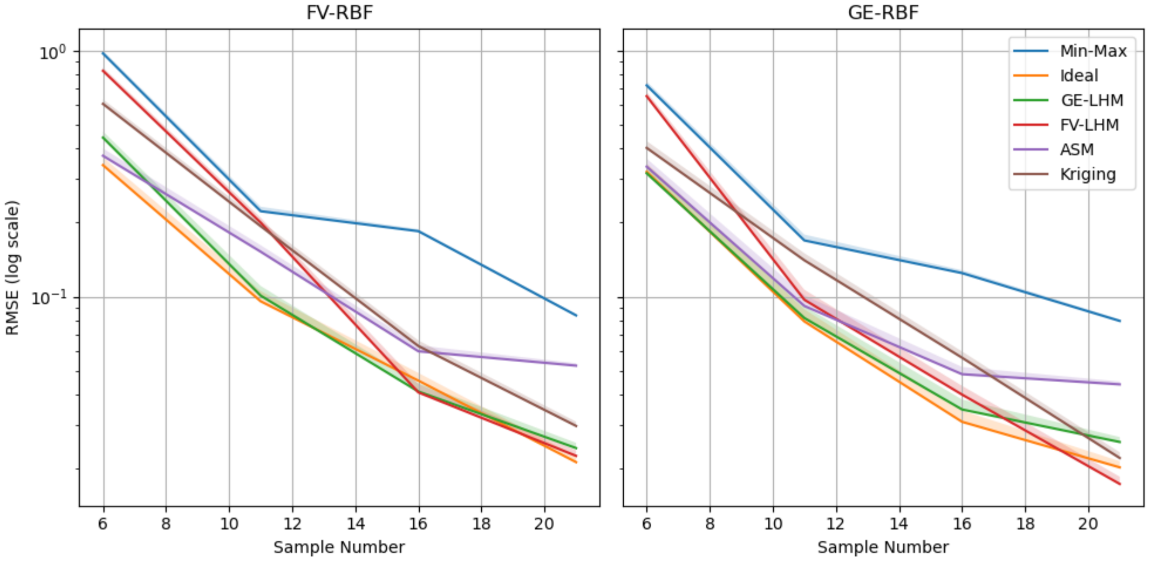


Figure 4.12. RMSE results for the FV-RBF (left) and the GE-RBF (right) on the 2-dimensional test problem.

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 complete 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 (4.13) and (4.14) present the results for the 4 and 8-dimensional problems respectively.

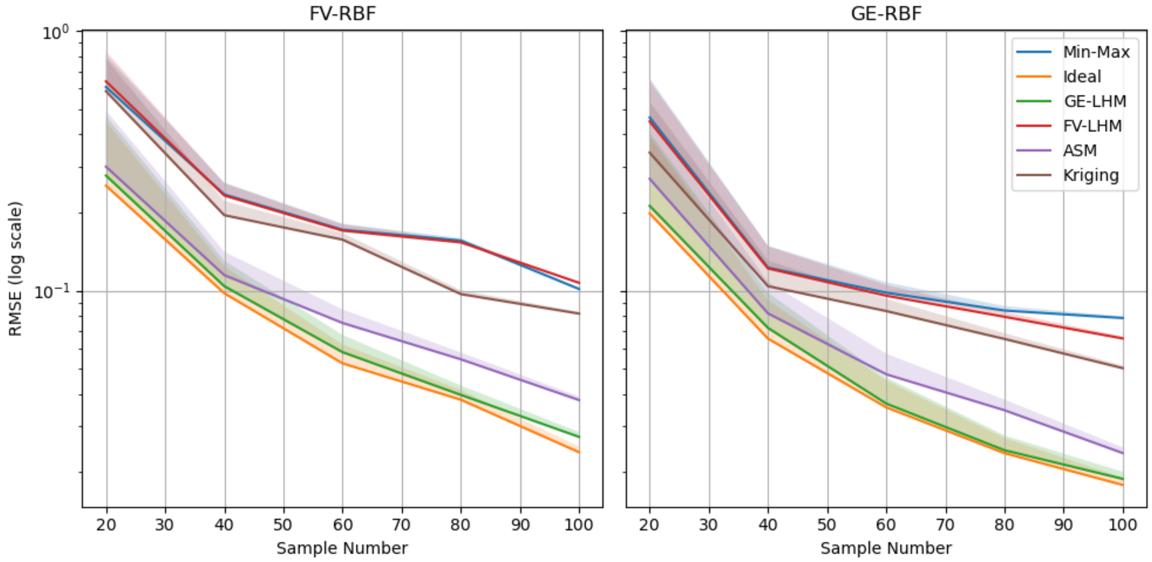


Figure 4.13. Log RMSE results for the FV-RBF (left) and the GE-RBF (right) on the 4-dimensional test problem.

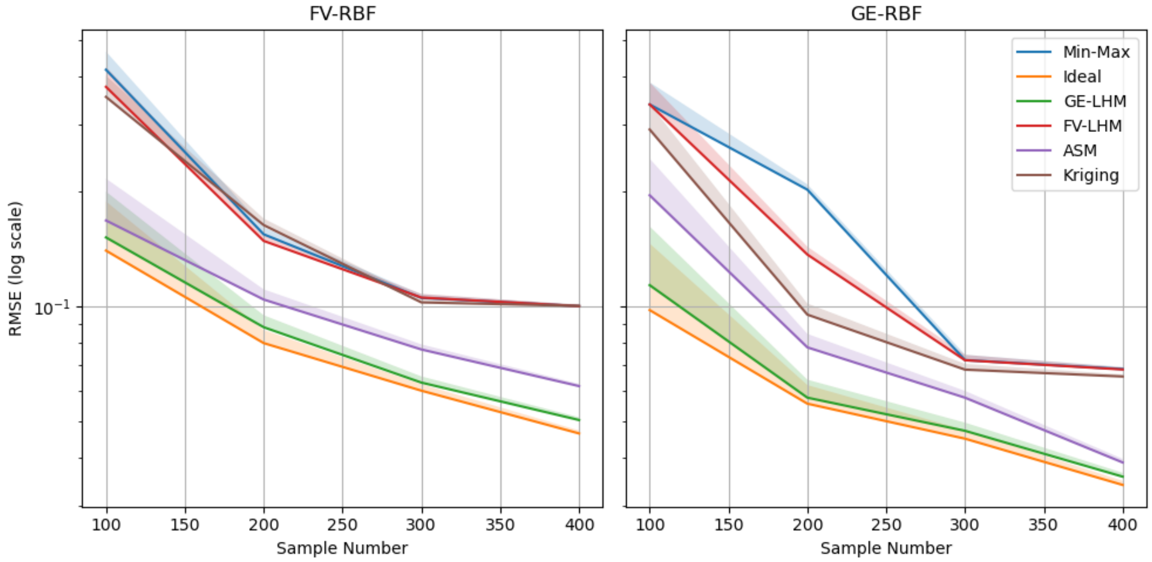


Figure 4.14. Log RMSE results for the FV-RBF (left) and the GE-RBF (Right) on the 8-dimensional test problem.

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 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 Figure (4.13) and (4.14) 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 coordinate system is a far more information dense task, that grows with the dimensionality of 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 the problem means that they 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 (4.15).

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.

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 grows 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.

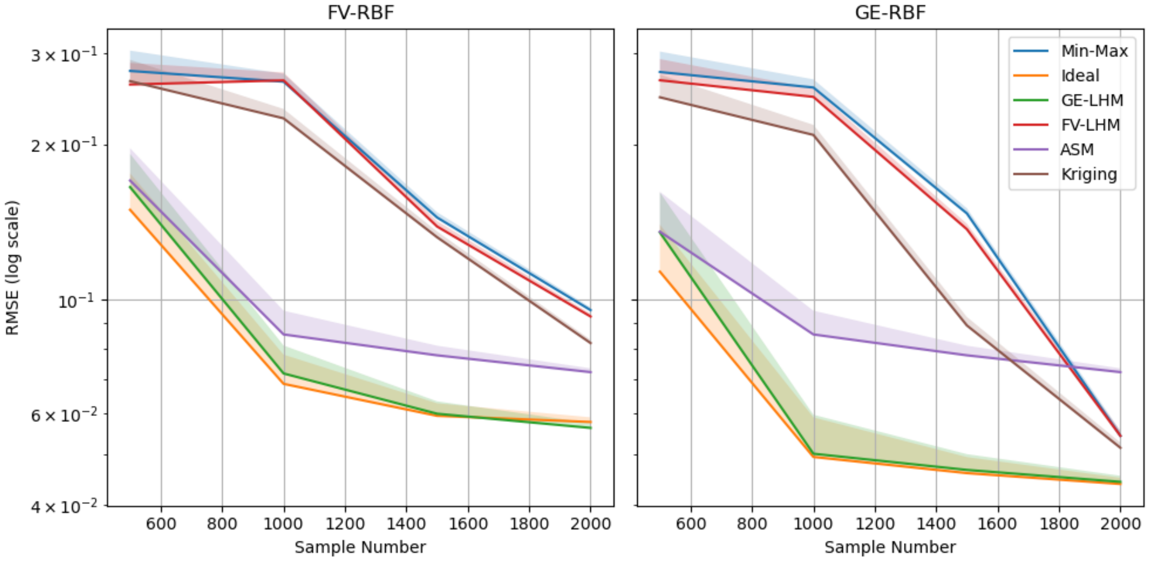


Figure 4.15. Log RMSE results for the FV-RBF (left) and the GE-RBF (Right) on the 16-dimensional test problem

An additional feature of the results in Figure (4.15) 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.

4.8 Non-Decomposable Functions

The developed transformation technique assumes that the underlying function is decomposable, meaning there exists a single linear transformation that will recast the problem into a coordinate system where the variables in the underlining function are uncoupled. To investigate the performance of the method on a non-decomposable problem the well known $n-$dimensional Rosenbrok function in the domain $[-1,1]^n \to \mathbb{R}$ is used. This problem is expressed in Equation (4.44) and the 2-dimensional problem is shown in Figure (4.17).

$$f(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100 \cdot (x_{i+1} - x_i^2)^2 + (1 - x_i)^2\right] \tag{4.44}$$

The GE-RBF model is then constructed for various sample numbers at various dimensionalities 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 (4.17)

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-

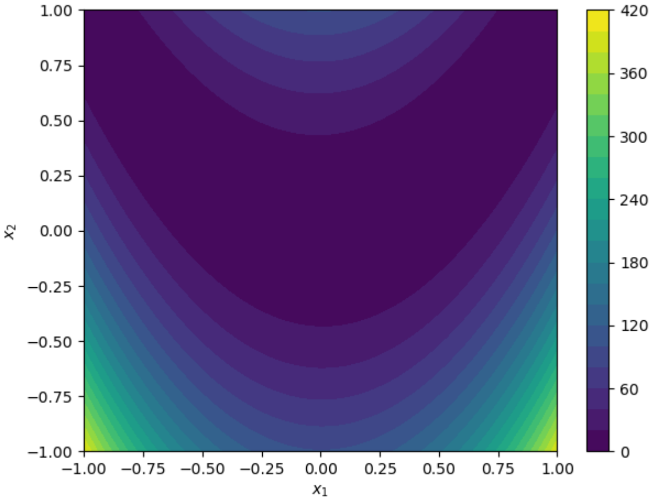


Figure 4.16. Contour plot of the 2 dimensional Rosenbrock function

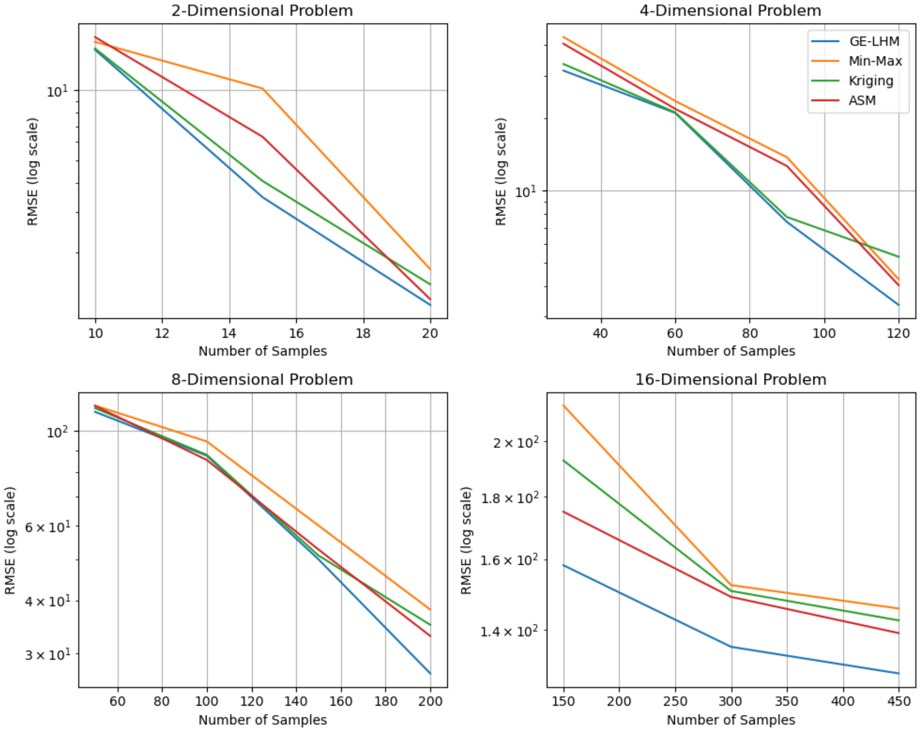


Figure 4.17. Results for the Rosenbrok Function

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

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

4.9 Chapter Conclusion

The work presented in this chapter demonstrates that the coordinate system in which common radial 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 $\hat{\mathbf{C}}$ 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.

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.

Therefore, the developed transformation scheme in this chapter can now be tested on the load path optimisation problem. Specifically, the ability of the RBF surrogate models, with the addition of the coordinate transformation scheme, to approximate the load path of a snap-through structure as a function of both spatial and temporal variables is investigated in the next chapter.

Chapter 5 Spatio-Temporal and Network Surrogate Models

5.1 Chapter Abstract

This chapter compares the performance of spatio-temporal surrogate models (STSMs) and network surrogate models (NSMs) when a system's response varies over time (or pseudo-time) and this response must be approximated. A surrogate model is used to approximate the response of computationally expensive spatial and temporal fields resulting from some computational mechanics simulations. Within a design context, a surrogate takes a vector of design variables that describe a current design and returns an approximation of the design's response through a pseudo-time variable.

To compare various radial basis function (RBF) surrogate modeling approaches, the prediction of a load-displacement path of a snap-through structure is used as an example numerical problem. This work specifically considers the scenario where analytical sensitivities are available directly from the computational mechanics' solver and therefore gradient enhanced surrogates are constructed. In addition, the gradients are used to perform a domain transformation preprocessing step to construct surrogate models in a more isotropic domain, which is conducive to RBFs. This work demonstrates that although the gradient-based domain transformation scheme offers a significant improvement to the performance of the spatio-temporal surrogate models (STSMs), the network surrogate model (NSM) is far more robust. This research offers explanations for the improved performance of NSMs over STSMs and recommends future research to improve the performance of STSMs.

5.2 Introduction

This chapter investigates the use and suitability of Radial Basis Functions (RBF) surrogate models to predict a system's response through some pseudo-time variable. This type of problem is often present in computationally expensive Computation Fluid Dynamics (CFD) simulations, crash-worthiness simulations, or pressure control simulations, where a designer wishes to predict the change in a design's behaviour without recomputing a computationally expensive simulation. Therefore, the surrogate model is intended to replace the computationally expensive simulation by accepting a vector of variables that describe a design and offer a computationally inexpensive approximation of the design's response through a pseudo-time variable. In this problem, the surrogate model needs to return a continuous approximation as a function of the pseudo-time variable, and not simply a single scalar value as is often the case with surrogate models. Hence, the surrogate needs to return a response vector, which is often referred to as a vector-valued surrogate model [61].

In this work, the scenario where the analytical gradients are available is considered. There are many examples in literature [19, 20, 22, 23] where detailed procedures to calculate the design sensitivities for functions that are computed using the Finite Element Method (FEM), Finite Volume Method (FVM) for structural mechanics and Computational Fluid Dynamics (CFD) are offered. Some finite element packages even have adjoint sensitivities implemented, for example, Calculix [44]. This gradient information can be calculated with respect to many different design variables for a wide range of problems [19, 22, 45–47].

As per usual practice, these analytical sensitivities are used directly in the RBF surface approximation, i.e., to construct Gradient Enhanced RBF models (GE-RBF) [53, 62]. In addition, these sensitivities are also used to perform a domain transformation preprocessing step that aims to find a transformation that results in a near isotropic surface to be approximated. This domain transformation step was previously developed by the author [34] to accommodate the application of RBF kernels that are isotropic on the approximation of non-isotropic surfaces often resulting in engineering problems [9, 35, 38, 39]. This is discussed in more detail later in the chapter in section (5.3).

There are two main options for constructing a surrogate model that can predict behaviour through a pseudo-time variable. Firstly, the design variables and the pseudo-time variable can be uncoupled and a network of smaller surrogate models can be constructed and trained at m predetermined locations in the pseudo-time variable. In order to sample the approximated behaviour, each RBF model is then sampled separately and some interpolation scheme is used to find the result at locations other than the m pseudo-time points. In this work simple linear interpolation is implemented. This surrogate model approach is referred to as the network surrogate model (NSM) [63].

The second option is to couple the design and pseudo-time variables and construct one surrogate model over the shape and pseudo-time variables. Such a surrogate model is often referred to as a spatio-temporal surrogate model (STSM) [64–66]. RBFs are a popular choice for the construction of STSMs, and is used in this study. Hence, from here onwards STSM will imply an RBF-constructed STSM. This option has the benefit of being easier to implement as only one model needs to be trained, whereas, for the NSM, m models need to be trained. However, as will be demonstrated in this chapter, if the limitations of the RBF model are not properly understood and addressed the STSM may offer inferior predictive performance when compared to the NSM.

To assess the surrogate models in this work, the models are used to predict the load-displacement paths of snap-through structures as a function of various shape parameters. As the load paths of the structures are highly non-linear, the finite element method (FEM) simulation needs to be completed with the Arc Length Control (ALC) method [13]. The ALC method introduces a pseudo-time variable, commonly referred to as the arc length, to allow the calculation of highly non-linear force-displacement curves.

The analytical sensitivity procedure used in this research to compute sensitivities has been completed previously by the author [21]. Specifically, the assumed stress element in conjunction with the ALC method is used to simulate the behaviour of snap-through structures.

The layout of this chapter is then as follows. Firstly, a brief mathematical description of RBF surrogate models is offered in Section (5.3), where the use of gradient information, both directly in the model itself and as a preprocessing step, is also detailed. This section also describes the NSM and STSMs in more detail. In Section (5.4) a more detailed description of snap-through structures' behaviour as well as the required ALC solution strategy is described. From the description, in Section (5.4) the numerical problem of predicting the load path of these structures is presented in Section (5.5). The numerical results are detailed in Section (5.6), where the models are used to predict the load-displacement curves of the selected snap-through mechanism. This is followed by observations and conclusions in Section (5.7).

5.3 RBF Surrogate Models

Surrogate model construction can broadly be separated into the following steps:

1. Domain sampling,
2. Preprocessing,
3. Model selection or construction,
4. Model Prediction.

Each of these steps is a research topic by itself, where various methods or algorithms have been developed to produce optimum results. In this research, each of the steps is completed with the most standard or common method, as the goal of the chapter is to isolate the effects of the preprocessing step.

Radial Basis Function surrogate models refer to a group of models that use a linear summation of basis functions that usually depend on some non-linear distance measure between two points. Popular options as basis functions include

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

where the variables $\mathbf{x}$, ε, $\mathbf{c}$ are the sampled point in design space, the centre of the basis function, and the shape parameter respectively. A widely used basis function is the Gaussian function [6,7,53], which is the basis function used in this research. The RBF surrogate is expressed as a linear combination of k basis functions

$$f_{\text{RBF}} = \sum_{i=1}^{K} w_i \phi(\mathbf{x},\mathbf{c}_i,\varepsilon). \tag{5.1}$$

This can be expressed as a system of equations

$$\mathbf{f} = \mathbf{\Phi}(\mathbf{x},\mathbf{c},\varepsilon)\mathbf{w}, \tag{5.2}$$

where the goal of fitting the surrogate model refers to finding the optimum values of the weight vector $\mathbf{w}$. The matrix $\mathbf{\Phi}$ is expressed as

$$\mathbf{\Phi} = \begin{bmatrix} \phi(\mathbf{x}_1,\mathbf{c}_1,\varepsilon) & \phi(\mathbf{x}_1,\mathbf{c}_2,\varepsilon) & \dots & \phi(\mathbf{x}_1,\mathbf{c}_k,\varepsilon) \\ \phi(\mathbf{x}_2,\mathbf{c}_1,\varepsilon) & \phi(\mathbf{x}_2,\mathbf{c}_2,\varepsilon) & \dots & \phi(\mathbf{x}_2,\mathbf{c}_k,\varepsilon) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(\mathbf{x}_p,\mathbf{c}_1,\varepsilon) & \phi(\mathbf{x}_p,\mathbf{c}_2,\varepsilon) & \dots & \phi(\mathbf{x}_p,\mathbf{c}_k,\varepsilon) \end{bmatrix}. \tag{5.3}$$

The size of matrix $\mathbf{\Phi}$ is, therefore, $p \times k$, where p is the number of samples in the design space and k is the number of centres or basis functions used in the model. The methods for determining the locations of the samples in the construction space in discussed in Section (5.3.3).

The remaining parameters of the surrogate include the number and locations of the centres $\mathbf{c}$ 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 [53]. For this choice the matrix $\mathbf{\Phi}$ becomes square and the weight vector can be solved directly from Equation (5.2). This is the option that is implemented in this research.

The shape parameter can either be determined with some heuristic [67], or with a k-fold cross-validation or Leave-Out-One cross-validation (LOOCV) approach [6, 53]. 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.

5.3.1 Direct Gradient Enhancement

The so-called Gradient Enhanced RBF (GE-RBF) surrogate model is implemented in this research [62]. The usual function-value-only RBF model can be expanded to include gradient information in its construction [53]. This can be done by first taking the gradient of the chosen basis function

$$\frac{d\phi(\mathbf{x},\mathbf{c},\varepsilon)}{d\mathbf{x}}^{T} = -2\varepsilon\phi(\mathbf{x},\mathbf{c},\varepsilon)(\mathbf{x}-\mathbf{c}), \tag{5.4}$$

where Equation (5.4) returns a column vector of the gradients of the RBF basis functions [53].

A new system of equations can then be created from the gradient information at each sampled point for the p samples of the RBF surrogate model [53]

$$\begin{bmatrix} \frac{df_1}{d\mathbf{x}} \\ \frac{df_2}{d\mathbf{x}} \\ \vdots \\ \frac{df_p}{d\mathbf{x}} \end{bmatrix} = \begin{bmatrix} \frac{d\phi(\mathbf{x}_1,\mathbf{c}_1,\varepsilon)}{d\mathbf{x}} & \frac{d\phi(\mathbf{x}_1,\mathbf{c}_2,\varepsilon)}{d\mathbf{x}} & \cdots & \frac{d\phi(\mathbf{x}_1,\mathbf{c}_k,\varepsilon)}{d\mathbf{x}} \\ \frac{d\phi(\mathbf{x}_2,\mathbf{c}_1,\varepsilon)}{d\mathbf{x}} & \frac{d\phi(\mathbf{x}_2,\mathbf{c}_2,\varepsilon)}{d\mathbf{x}} & \cdots & \frac{d\phi(\mathbf{x}_2,\mathbf{c}_k,\varepsilon)}{d\mathbf{x}} \\ \vdots & \vdots & \vdots & \\ \frac{d\phi(\mathbf{x}_p,\mathbf{c}_1,\varepsilon)}{d\mathbf{x}} & \frac{d\phi(\mathbf{x}_p,\mathbf{c}_2,\varepsilon)}{d\mathbf{x}} & \cdots & \frac{d\phi(\mathbf{x}_p,\mathbf{c}_k,\varepsilon)}{d\mathbf{x}} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix}. \tag{5.5}$$

This system can then be written as

$$\mathbf{f}_{fo} = \mathbf{\Phi}_{fo}\mathbf{w}_{fo}. \tag{5.6}$$

The subscript fo denotes that first-order information is used in the system. The gradient information can then be combined with the original function-value-only system, Equation (5.2), to create a new system of equations

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_{fo} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} \\ \mathbf{\Phi}_{fo} \end{bmatrix} \mathbf{w}_{GE}. \tag{5.7}$$

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. Equation (5.7) can then be expressed as

$$\mathbf{f}_{GE} = \mathbf{\Phi}_{GE}\mathbf{w}_{GE}. \tag{5.8}$$

As the weight vector, $\mathbf{w}_{GE}$, is the same size, specifically $k \times 1$ in both the function and GE models, the system is, therefore, over-determined [53]. The difference between the function and GE models is therefore that the GE models regress through both the zero-order and gradient information [53]. The $\mathbf{w}_{GE}$ is found using the least squares formulation,

$$\mathbf{\Phi}_{GE}^{T}\mathbf{f}_{GE} = \mathbf{\Phi}_{GE}^{T}\mathbf{\Phi}_{GE}\mathbf{w}_{GE}. \tag{5.9}$$

5.3.2 gradient-based Domain Transformation

Typically, the domain over which the RBF surrogate model is constructed is the result of a simple min-max scaling of the design domain as a preprocessing step [6]. This is done by

$$x_s = \frac{x - x_{min}}{x_{max} - x_{min}}, \tag{5.10}$$

for each design variable. Here x_s is the scaled construction domain of the surrogate model, x_{min} is the minimum value of the design variable and x_{max} is the maximum value of the design variable.

On the other hand, previous work by the author demonstrated that making use of the gradient information to approximate the underlying curvature of the function can greatly improve the predictive performance of the RBF model [34]. This is due to the fact that RBF models often make the implicit assumption that the underlying function is isotropic which is often not a suitable assumption in many practical engineering problems. Work to address the isotropic problem includes using Neural Networks, Gaussian Mixture Models, or Sub-space methods [68, 69]. In this work, the author select to handle the isotropic problem using their developed preprocessing procedure.

The proposed method developed by the author makes use of the gradient information to approximate the underlying curvature and recast the problem into a domain where the isotropic assumption is more appropriate by completing a full transformation, i.e. a rotation and scaling, of the domain as a preprocessing step.

The method can be broken into the following steps:

1. For each centre point **c**:
    (a) Find the $n+1$ closest points to the centre in the sampled points,
    (b) complete a Symmetric Rank 1 approximation of the local curvature or Hessian [53],
    (c) find the eigenvalues and eigenvectors of the local Hessian,
    (d) recalculate the local Hessian with the absolute value of the eigenvalues.
2. Find the average local Hessian from the collection of absolute Hessians.
3. Calculate the eigenvalues and eigenvectors of the average Hessian.
4. Rotate the domain using the eigenvectors and scale with the eigenvalues of the average Hessian.

This method was shown to recover the ideal reference frame and scaling, for a benchmark problem that is decomposable in a rotated frame [34].

5.3.3 Sampling Methods

Typically the locations of the samples for the construction of the RBF surrogate model are determined with some Latin Hyper-Cube Sampling (LHS) method [7, 53]. The standard LHS method is shown in Figure (5.1) for an increasing number of samples in two-dimensions.

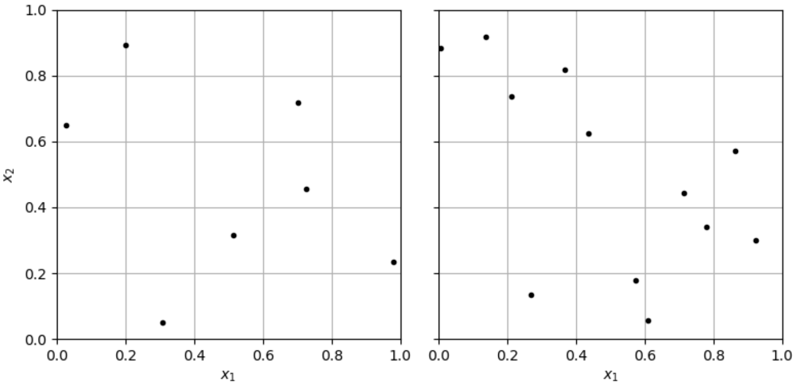


Figure 5.1. An example of the LHS method in a two Dimensional problem with 7 and 12 samples respectively.

The standard LHS method can be improved upon by adding additional constraints to the method such as maximising the minimum distances between nearest neighbours or incorporating Halton sequences [6]. In this research the sampling strategy is limited to the simplest version of LHS as the goal of the chapter is to isolate the effects appropriate domain transformation have on NSM and STSM.

In addition to the LHS method, in this work, the sampling of the pseudo-time variable is completed during some numerical simulation using an iterative solver. This means that the designer has little or no control over the locations, or number in the case of an adaptive solver, of the samples in this domain. This creates a non-uniform sampling scheme as demonstrated in Figure (5.2), where design problems with a pseudo-time variable are sampled.

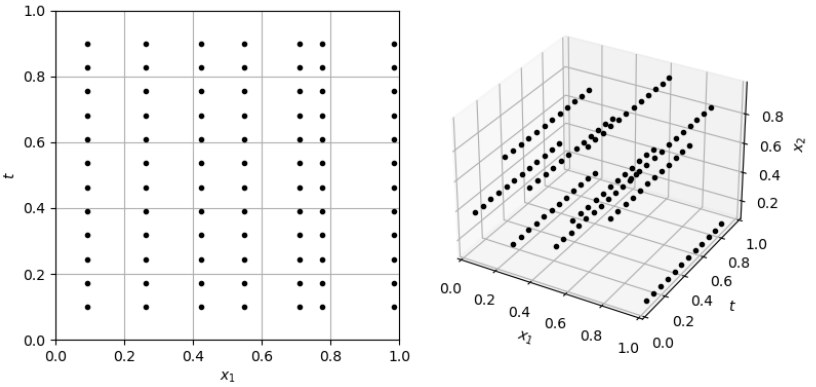


Figure 5.2. Locations of samples in a one and two design variable problem respectively with a pseudo-time variable t.

5.3.4 Network (NSMs) and spatio-temporal surrogate models (STSMs)

The surrogate models in this work need to return a continuous approximation of a result through a pseudo-time variable. The first method is to uncouple the design and pseudo-time variables and construct multiple smaller RBF models at predetermined locations in the pseudo-time variable. This is represented visually in Figure (5.3).

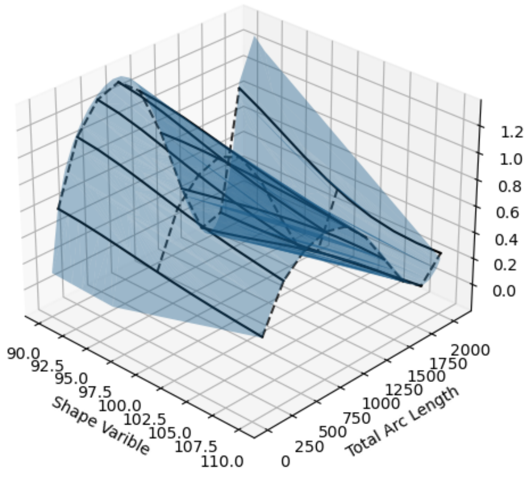


Figure 5.3. An Example of the NSM approximating an underlying function in blue. The solid black lines through the design variable domain indicate the constructed RBF models, while the dashed black lines through the pseudo-time domain indicate the interpolated final approximated curve.

The designer needs to determine the number of smaller RBF models as well as their locations in the pseudo-time domain. These decisions are problem specific. As more locations in the pseudo-time variable are selected to have an RBF model, the more flexible the final approximations of the behaviour will be, but, the more computationally expensive the training of the surrogate model becomes.

On the other hand, STSM couple the design and pseudo-time variables into one large surrogate model. This method is represented visually in Figure (5.4) on the same underlying function as for the NSM in Figure (5.3).

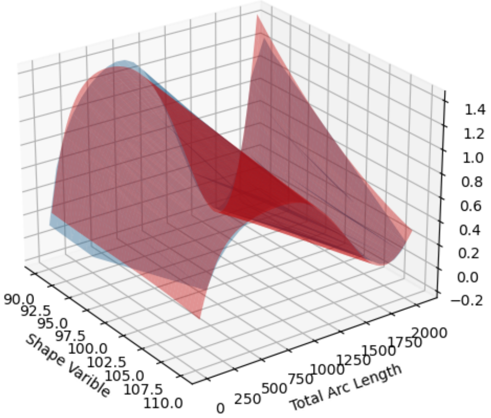


Figure 5.4. Example of an STSM in red approximating the underlying function in blue.

Another key difference, that is highlighted in this research, between the two models is how they make use of gradient information during their construction. The NSM is only constructed in the design variable domain and therefore cannot make use of the gradient information with respect to the pseudo-time variable. Once all the network RBFs are evaluated at a specific point in pseudo-time, it would be possible to construct some interpolation scheme that uses the gradient information with respect to the pseudo-time variable. In this chapter the linear interpolation in the pseudo-time variable does not use the available gradient information in this direction. On the other hand, the STSM is constructed in both domains, and can therefore make use of all the gradient information present in the problem.

5.4 Simulating Snap-Through Behaviour

Snap-Through structures are compliant mechanisms that exhibit highly non-linear load-deflection paths. This means that typical linear finite element solutions strategies are not equipped to fully simulate the structures' behaviour. Instead, a combination of a non-linear solver with the ALC method needs to be implemented in order to accurately predict the structures' load-displacement path. In typical non-linear FEM analysis, the governing residual equation is expressed as

$$\mathbf{R} = \mathbf{K}(\mathbf{u})\mathbf{u} - \lambda \mathbf{F}, \tag{5.11}$$

where $\mathbf{K}$, $\mathbf{u}$, λ, and $\mathbf{F}$ denote the stiffness matrix, the nodal displacement vector, the load parameter, and the load vector respectively. In the load control scheme, the goal of some iterative non-linear solver is to find the nodal displacement vector at some load parameter that reduces the governing residual, Equation (5.11), to $\mathbf{0}$ within some desired tolerance.

This iterative solver typically takes the form of Newton's method, where the gradient of Equation (5.11) with respect to the nodal displacement vector is computed. A problem with this solution strategy arises when the load path exhibits a limit point, which is present in snap-through problems, as Newton's method cannot fully trace the curve past this point.

The ALC algorithm proposes a solution to this problem by adding an additional constraint equation,

$$L^2 = \blacktriangle\mathbf{u}^{\mathrm{T}}\blacktriangle\mathbf{u} + \psi^2\blacktriangle\lambda^2\mathbf{F}^{\mathrm{T}}\mathbf{F}. \tag{5.12}$$

Here, L denotes the prescribed arc length, $\blacktriangle\mathbf{u}$ is the total displacement vector update for the current load step, $\blacktriangle\lambda$ is the total load parameter increment for the current load step, and ψ is some non-dimensional scale factor. For a more in-depth description of the ALC algorithm, the reader is referred to literature [13].

The important characteristic of the ALC algorithm that influences this research is that the load and displacement results are now functions of the introduced arc length variable, as can be seen in Equation (5.12). This means that in this research the arc length domain can be treated as a pseudo-time variable when the surrogate models are constructed.

5.4.1 Design Sensitivities

The FEM simulations in this research are completed using the assumed stress element [12] in combination with the ALC method. The assumed stress element is implemented as often snap-through behaviour occurs in bending-dominated problems and the assumed stress element returns accurate results in bending problems. Therefore, as the analytical sensitivities are required for this research, the author implement a procedure developed in previous research where the same solution strategy was selected [21]. This sensitivity procedure returns the sensitivity information with respect to both the shape variables as well as the arc length variable.

5.5 Numerical Example

The example problem for this research is to predict the non-linear load path of the so-called Deep Semi-Circular Arch [10]. The load-displacement path as well as the deformation overlay is depicted in Figure (5.5).

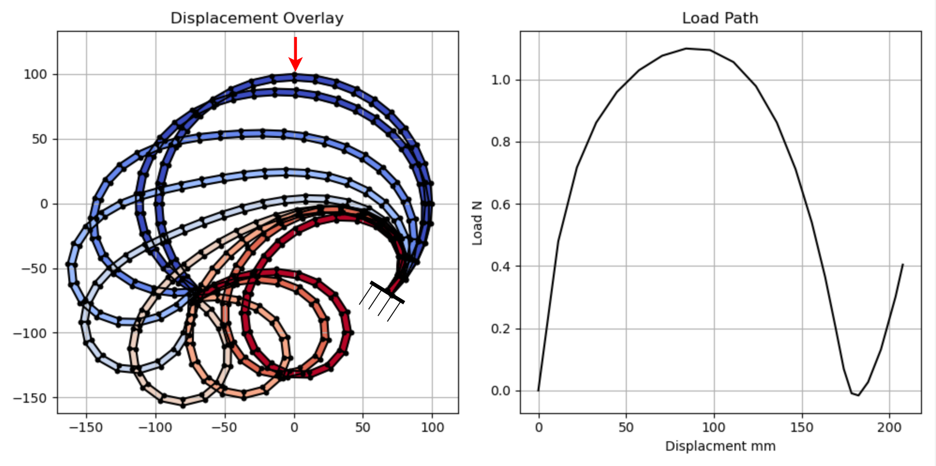


Figure 5.5. The deformation of the Deep Semi-Circular Arch that is pinned at the left and clamped on right edge under a single point load.

This structure is parameterised such that a vector of design variables describes the radii of the structure at equidistant points along the circumference. A cubic spline is then fitted through these radii and the overall shape of a trial design is found. This is shown in Figure (5.6).

This parametrisation scheme is selected as the number of design variables can be systematically increased to investigate the performance of the surrogate models as a function of the dimensionality

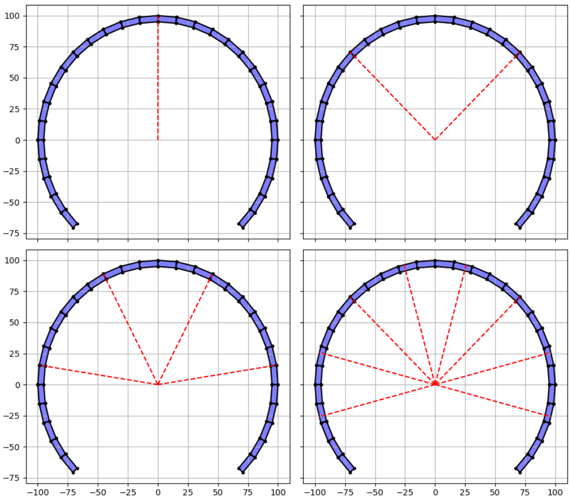


Figure 5.6. The design variables of the Deep Semi-Circular Arch depicted in red with the design problem increasing in dimensionality.

of the problem. The radii are sampled between 90mm and 110mm and the accumulated arc length is limited to 2000 with the default arc length increment set to 100.

5.6 Results

The results of six different gradient-enhanced surrogate models are found. These include the domain-transformed versions for both the NSMs and STSMs, as well as their standard min-max implementations. NSMs include versions where the shape parameter of the surrogate models in the network is allowed to vary and where the parameter is kept constant for all the models in the network. By keeping the shape parameter constant the NSM model has the same number of tunable parameters as the STSM. The number of tunable parameters for NSM and STSM is shown in Table (5.1). The models are constructed on an equal number of sampled points and are evaluated by comparing the predicted load-displacement paths to actual results.

Table 5.1. The number of parameters that must be tuned in the NSM where the shape parameter is kept constant, the shape parameter is allowed to vary, and the STSM. The number of parameters is expressed as a function of the number of centres K and the number of RBF models L in the NSM.

Parameters	NSM Constant Shape Parameter	NSM Varying Shape Parameter	STSM
Weights	K	K	K
Shape Parameters	1	L	1

Initially, a simple univariate shape variable problem is completed using 7 designs sampled using LHS. Figure (5.7) compares the predicted load-displacement curves with the actual response, for four random test designs.

The models are then assessed further by calculating the Root Mean Square Error (RMSE) for 75 load-displacement paths for randomly sampled designs. The RMSE is calculated by comparing the approximated load (λ)and displacement (u) results of the model to the new 75 sampled load-displacement paths. The models are evaluated at the arc length positions found by the solver for each of the new load-displacement paths. This implies that the models must predict at new design variable

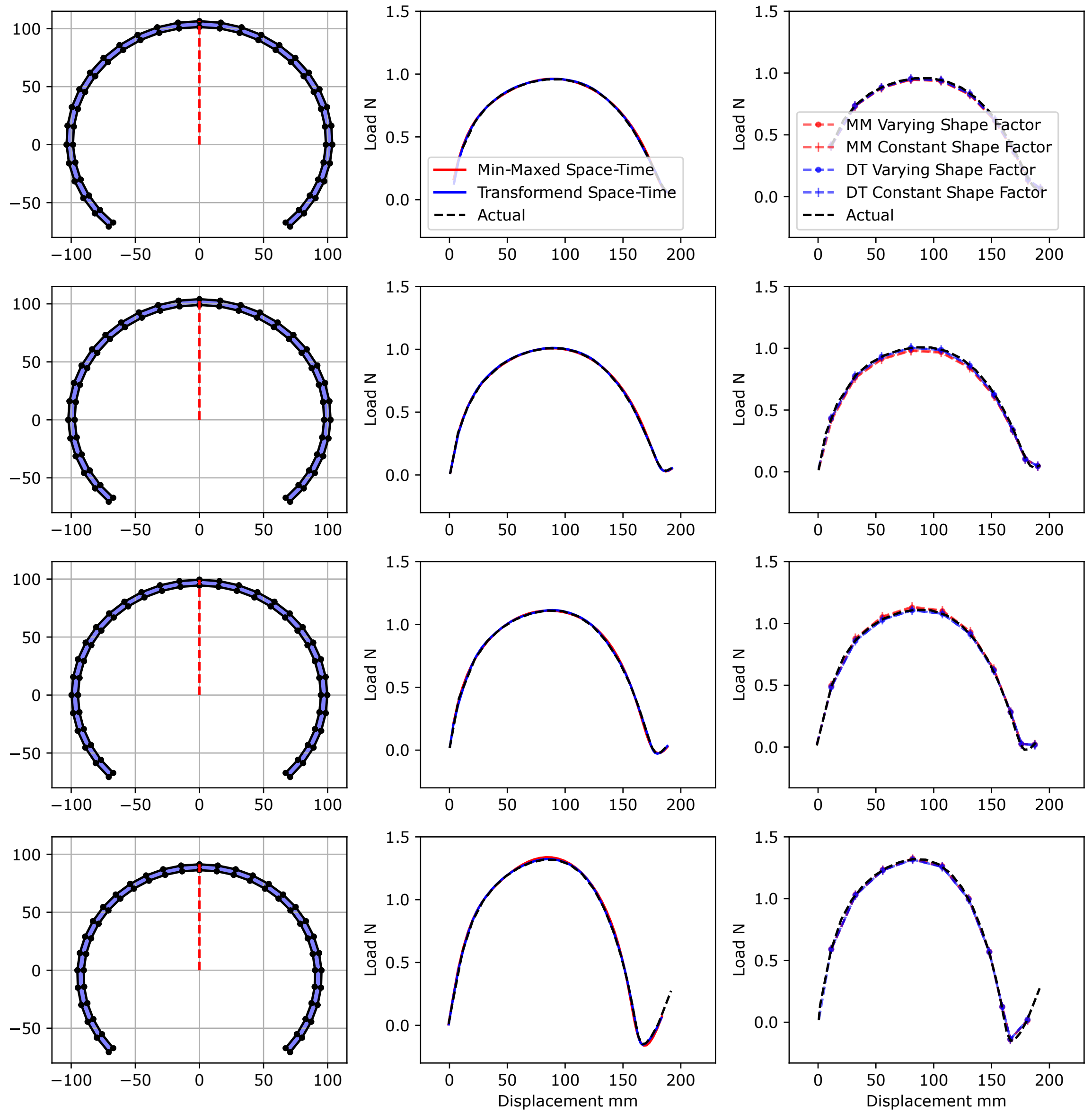


Figure 5.7. One shape variable problem results for four randomly generated designs. The simulated structures are shown of the left, followed by the STSM results, and then the NSM results with the labels MM and DT indicating min-maxed and full domain transformation respectively.

locations as well as new arc length or pseudo-time locations. This RMSE for a single load-displacement curve is expressed as

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i}^{n}(\lambda_i^{approx} - \lambda_i^{target})^2 + \frac{1}{n}\sum_{i}^{n}(u_i^{approx} - u_i^{target})^2}, \tag{5.13}$$

where n is the number of arc length steps and the values of λ and u are normalised using the maximum values from the training samples.

The average RMSE results are shown in Figure (5.8) in the form of a box and whisker plot.

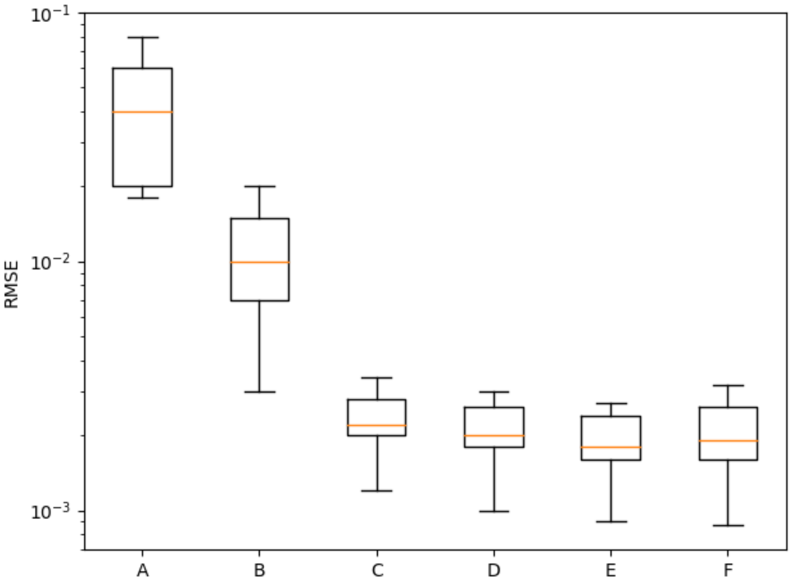


Figure 5.8. Average RMSE results for the one shape variable problem. A) min-max STSM, B) transformed STSM, C) min-max NSM with a constant shape factor, D) min-max NSM with a varying shape factor, E) transformed NSM with a constant shape factor, and F) transformed NSM with a varying shape factor

.

Clearly, the NSM outperforms the STSM but performing the gradient-based domain transformation offers a significant improvement to the predictive performance of the STSM. There is also very little improvement (if any) in the NSM when the shape parameter is allowed to vary compared to when the shape parameter is kept constant. Hence, the improved NSM fits are not the result of additional shape parameter flexibility, but rather due to ability of the NSM to complete different domain transformations as a function of the pseudo-time.

The dimensionality of the problem is now increased to two shape variables and the number of LHS samples is increased to 12. The results of the model are again overlaid with the simulated results in Figures (5.9), and the average RMSE results are displayed in Figure (5.10).

The results of the two-variable shape variable problem are similar to the univariate shape variable problem, with the NSMs outperforming the STSMs. In addition, the domain transformation improves both the NSMs and STSMs. The benefit of an appropriately transformed domain is evident for the two-variable problem. The difference in performance between the min-maxed models and the transformed models clearly increased , while there is a negligible difference between the varying shape parameters models compared to the constant shape parameter models.

Lastly, the problem dimension and the LHS sample number is increased to four shape variables and 25 samples respectively. The results are presented in Figures (5.12) and (5.11).

This problem demonstrates that the benefit of appropriately transforming the construction domain increases with the problem dimension. The min-max STSMs offer extremely poor predictions of the underlying function, while the transformed model remained robust.

The NSMs still outperform the STSMs, but there is a clear improvement at higher problem dimensions when the transformed domain is employed as compared to the min-max scaling. On the other hand, the

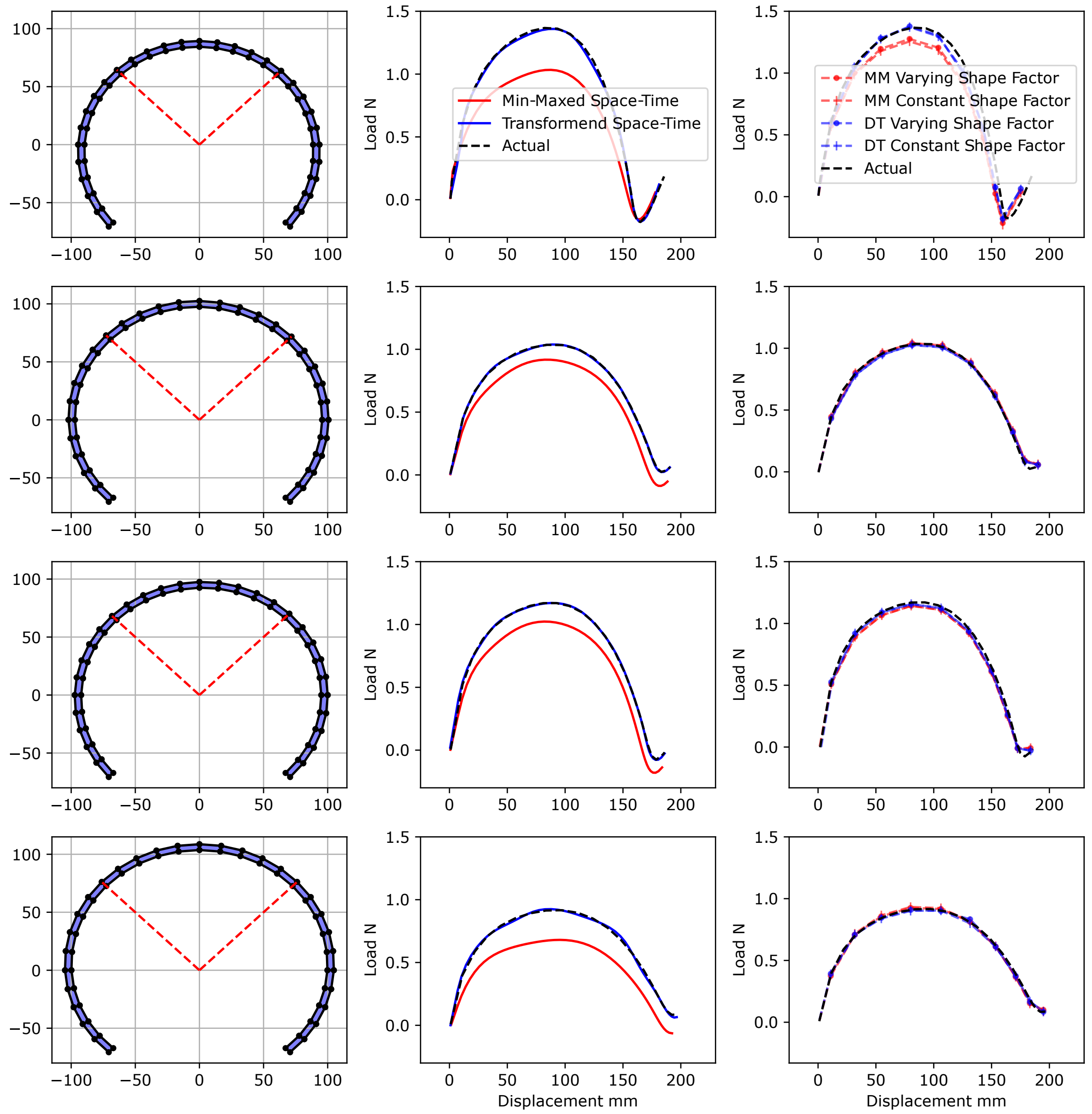


Figure 5.9. Two shape variables problem results for four randomly generated designs. The simulated structures are shown of the left, followed by the STSM results, and then the NSM results with the labels MM and DT indicating min-maxed and full domain transformation respectively.

varying shape parameter offers very little improvement to the model over the constant shape parameter models, despite this increase in flexibility.

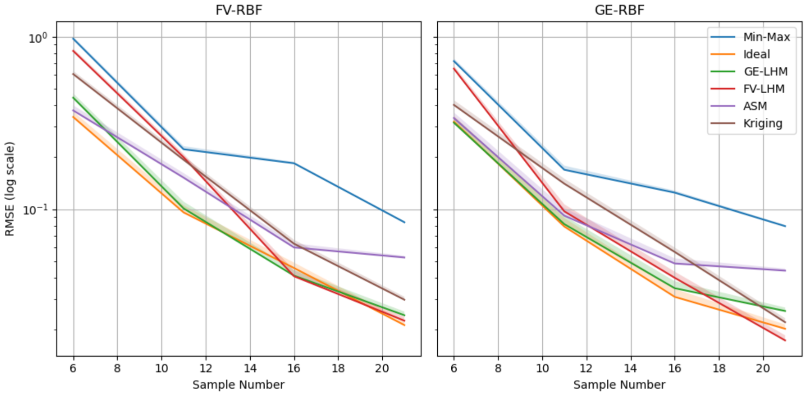


Figure 5.10. Average RMSE results for the two shape variable problem. A) min-max STSM, B) transformed STSM, C) min-max NSM with a constant shape factor, D) min-max NSM with a varying shape factor, E) transformed NSM with a constant shape factor, and F) transformed NSM with a varying shape factor

.

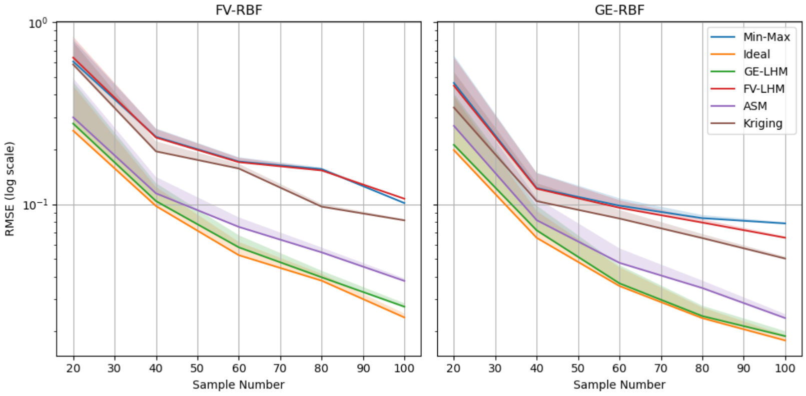


Figure 5.11. Average RMSE results for the four shape variable problem. A) min-max STSM, B) transformed STSM, C) min-max NSM with a constant shape factor, D) min-max NSM with a varying shape factor, E) transformed NSM with a constant shape factor, and F) transformed NSM with a varying shape factor

.

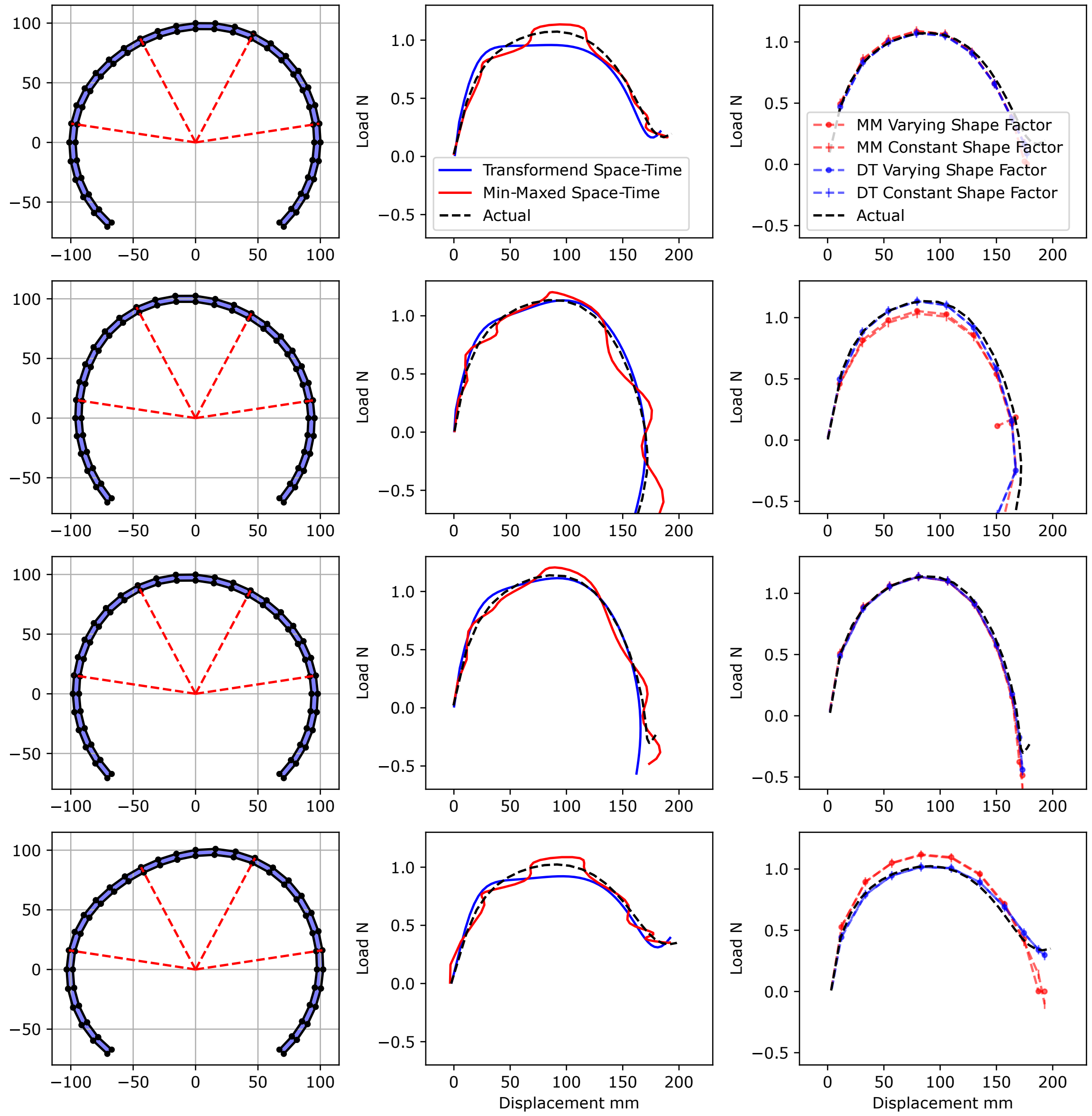


Figure 5.12. Four shape variable problem results for four randomly generated designs. The simulated structures are shown of the left, followed by the STSM results, and then the NSM results with the labels MM and DT indicated min-maxed and full domain transformation respectively.

5.7 Chapter Conclusions

This chapter demonstrates that constructing surrogate models to predict the spatio-temporal behaviour of a system requires careful consideration as a naive implementation may be unable to capture the inherent complexities present in the spatio-temporal response. This study compares two strategies to construct spatio-temporal surrogates, namely, spatio-temporal surrogate models (STSMs) and network surrogate models (NSMs). From this, two main findings regarding the construction of surrogate models are observed.

Firstly, merely including gradient information in the construction of the surrogate models is not sufficient to construct high-quality models. Instead, the gradients should be employed to transform the domain to a more isotropic domain (the inherent assumption of RBFs). This enables the construction of accurate surrogate models for higher dimensional problems.

Secondly, NSMs significantly outperform the STSMs. The results from the constant and varying shape parameter models demonstrate that this increase in performance is not due to the ability of the NSM to fit different shape parameters, but rather it is due to two underlying characteristics of the problem:

1. the locations of the centres of the spatio-temporal RBF model have a non-uniform structure to them, i.e. the pseudo-time variable is sampled more densely than the shape variables, and
2. a single linear transformation might not be sufficient to transform the domain to be isotropic. The shape variables may impact the results differently at different stages in the pseudo-time variable domain and therefore a nonlinear transformation scheme that is a function of the pseudo-time variable is required.

Therefore, from the results in this chapter, it is clear that before the spatio-temporal can be implemented to complete the shape optimisation problem the anisotropic locations of the samples in the full spatio-temporal domain needs to be addressed. This is completed in the following chapter by redistributing the kernels throughout the full spatio-temporal domain. The original shape optimisation problem is then solved using these vastly improved surrogate models.

Chapter 6 Redistributing the Kernel Centres of Spatio-Temporal Surrogate Models

6.1 Chapter Abstract

This chapter expands on Bouwer et al. [70]'s work to improve Spatio-Temporal surrogate models' performance. Specifically, the case of non-uniform sampling is considered when the spatial variables are sampled with a different numerical strategy than the temporal variable. Poor surrogate accuracy for the non-uniform sampling case is addressed by redistributing the centres of the Radial Basis Function (RBF) model throughout the spatio-temporal domain.

The benefit of separating the centre locations from the sample locations is demonstrated on a numerical test problem, the same problem used in Bouwer et al. [34], and a practical engineering problem where the non-linear load path of the Deep Semi-Circular Arch (DSCA) [10, 21, 70] is subject to a shape optimisation problem.

The chapter concludes that the anisotropic behaviour in both the underlying function and the sampling locations must be addressed before the spatio-temporal surrogate models are comparable in performance to the network surrogate model approach.

6.2 Introduction

This chapter extends the investigation completed by Bouwer et al. [70] into the use and suitability of Radial Basis Functions (RBF) surrogate models to predict a system's response that varies as a function of some pseudo-time variable. In the authors' previous work [70], it was shown that using spatio-temporal surrogate models (STSM) is often inferior to using a network of surrogate models (NSM) placed at predetermined locations in the pseudo-time variable. It was also demonstrated that using a novel domain transformation scheme [34] developed by the authors offered improvement to both STSMs and NSMs. The transformation scheme uses gradient information to estimate a more isotropic design coordinate system, thereby improving the predictive performance of RBF surrogate models.

The process of surrogate modelling can be separated into four main steps,

1. Data collection,
2. Pre-processing of the data,
3. Surrogate construction,
4. Post-processing.

These steps and some common strategies employed in these steps, are depicted in Figure (6.1).

Previous research by the authors [34, 70] focused on the pre-processing of the collected data to reduce model bias. The work completed in this chapter focuses on improving the model's training, specifically

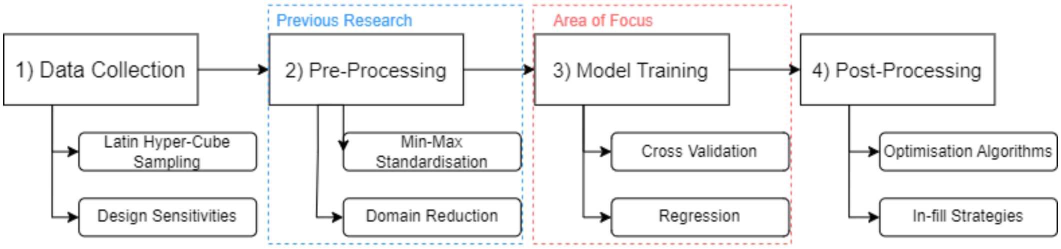


Figure 6.1. Four main steps of surrogate construction, and associated common strategies.

in the case where spatio-temporal surrogate models are implemented by redistributing the centres throughout the spatio-temporal domain.

The difference between spatio-temporal problems and more standard full spatial problems is that for the spatio-temporal problem, there is an additional variable is sampled differently from the spatial variables. Typically, the data collection step is completed using some variation of Latin Hyper-Cube sampling to determine the locations of data points in the design domain. In spatio-temporal problems, however, the time variable is sampled during the computational simulation, which is typical with some iterative solver. This difference in sampling strategies is demonstrated in Figures (6.2) and (6.3).

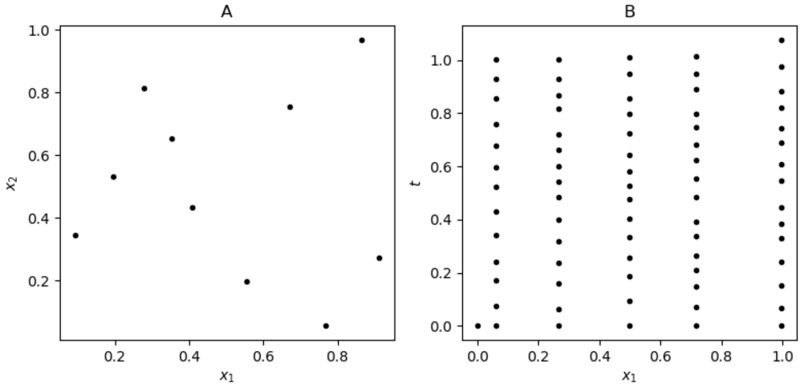


Figure 6.2. The difference in sampling in a 2-dimensional problem for A) the full spatial problem and B) the spatio-temporal problem.

The main characteristic to note which is important for the work completed in this chapter is the non-symmetric nature of the sampling in the spatio-temporal problem compared to the full spatial problem. This non-symmetry negatively impacts the STSM's performance in much the same way that an anisotropic underlying function is poorly modelled by RBF models [9, 34, 35, 38, 39]. In other words, the isotropic radial basis function used in these models is ill-suited to the anisotropic nature of the data locations in the design space. This work will demonstrate that the anisotropic nature of the underlying function and the sampling strategy in spatio-temporal problems must be addressed before the STSMs are comparable to the NSMs.

The layout of this chapter is then as follows. Firstly, Section (6.3) discusses surrogate-based optimisation and the concepts needed to complete the developed research. A brief mathematical description of RBF surrogate models is offered in Section (6.3.3). This section also describes the NSM and STSM in more detail. From these descriptions two numerical example problems are completed where the benefit of redistributing the centres of the RBF model is demonstrated. Firstly, in Section (6.4) a carefully

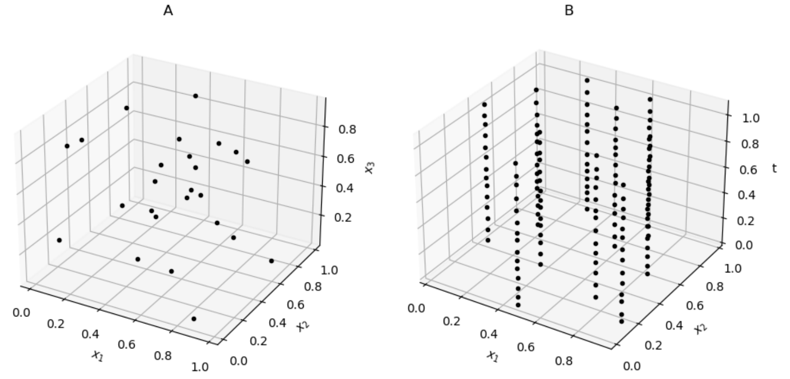


Figure 6.3. The difference in sampling in a 3-dimensional problem for A) the full spatial problem and B) the spatio-temporal problem.

crafted test problem [34] is solved. Secondly, the Deep Semi-Circular Arch (DSCA) [10, 70] is used, where the highly non-linear load path of a snap-through structure is subjected to a shape optimisation problem. Lastly, based on the results of these two problems, conclusions and recommendations are offered in Section (6.6).

6.3 Related Work

The type of problems that will typically make use of spatio-temporal models include Computation Fluid Dynamics (CFD) simulations, crash-worthiness simulations, or pressure-controlled simulations, where a designer wishes to predict the change in a design's response as a time-like variable change without recomputing a computationally expensive simulation. The surrogate model must accept a vector of design variables and return a continuous approximation as a function of some pseudo-time variable and not simply a single scalar value, as is often the case with surrogate models.

Therefore, the main research areas depicted in Figure (6.1) are discussed in this section in the context of predicting a design's response as some pseudo-time variable changes.

6.3.1 Data Collection

Typically in surrogate modelling the design domain is sampled using some Latin Hyper-Cube sampling strategy [6, 9]. This strategy is depicted in Figures (6.2)A and (6.3)A.This strategy can add space-filling criteria to the sample locations, such as maximising the average minimum distance between the locations, which is beneficial to surrogate performance [6]. In this research the standard variation of the method, without additional criteria, is implemented for both the sample locations and the redistributed centre locations.

It is also possible to sample gradient information during the data collection phase. In this research, these gradient vectors are used to complete the Local Hessian Method (LHM) coordinate system transformation developed by the authors [34]. The gradients are included in constructing the spatio-temporal and network surrogate models to create Gradient Enhanced Radial Basis Functions (GE-RBF) [53, 62]. Many papers [19, 20, 22, 23] offer procedures to obtain the analytical gradient vectors for functions that are computed using the Finite Element Method (FEM) or Finite Volume Method (FVM) for structural mechanics and Computational Fluid Dynamics (CFD), respectively. Some finite element packages even have adjoint sensitivities implemented, for example, Calculix [44]. These gradient vectors can be calculated for different design variables for various problems [19, 22, 45–47].

6.3.2 Pre-Processing

Typically, the domain over which the RBF surrogate model is constructed results of a simple min-max scaling of the design domain as a pre-processing step [6]. This is done by computing a scaled version of each variable from

$$x_s = \frac{x - x_{\min}}{x_{\max} - x_{\min}}. \tag{6.1}$$

Here x_s is the scaled construction domain of the surrogate model, $x_{\min}$ is the minimum value of the design variable and $x_{\max}$ is the maximum value of the design variable.

In this work the surrogate models will be constructed in the coordinate system found by LHM [34], using the sampled gradient information. This transformed coordinate system attempts to recast the problem such that the underlying function is more isotropic, and therefore, the model bias of the RBF model is decreased. In the case of the NSM, each of the m models in the network have their own linear transformation, while the STSM model only has one global linear transformation.

6.3.3 Model Training

Two main options exist for constructing a surrogate model to predict behaviour through a pseudo-time variable. Firstly, the design and pseudo-time variables can be uncoupled, and a network of smaller surrogate models can be constructed and trained at m predetermined locations in the pseudo-time variable. If, as with the Arc Length Control method, the implemented numerical solver can adapt the step size through the pseudo-time variable, the results will most likely not be available at the m locations of the network RBF models. Therefore, some interpolation of the function values and the gradient vectors found during the simulation needs to be completed with respect to the pseudo-time variable such that the results are located at the same locations as the m models. In this work, this interpolation is also completed using RBF models. To sample the approximated behaviour, each RBF model is then sampled separately. Then a RBF model is used to interpolate these results to return a continuous approximated response curve. This surrogate model approach is referred to as the network surrogate model (NSM) [63].

The second option is to couple the shape design and the pseudo-time variables and construct one surrogate model over the shape and pseudo-time variables. Such a surrogate model is often referred to as a space-time or spatio-temporal surrogate model (STSM) [64–66]. This option has the benefit of being easier to implement as only one model needs to be trained, whereas, for the NSM, m models need to be trained. Unlike the NSM, this model can also directly use the available gradient in the pseudo-time direction as this variable is included in the model.

Figures (6.4) and (6.5) graphically show the NSM and the STSM respectively.

RBF surrogate models refer to a group of models that use a linear combination of basis functions that depend on a distance measure between two points. Popular options as basis functions include

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

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

$$f_{\text{RBF}} = \sum_{i=1}^{k} w_i \phi_i(\mathbf{x},\mathbf{c}_i,\varepsilon), \tag{6.2}$$

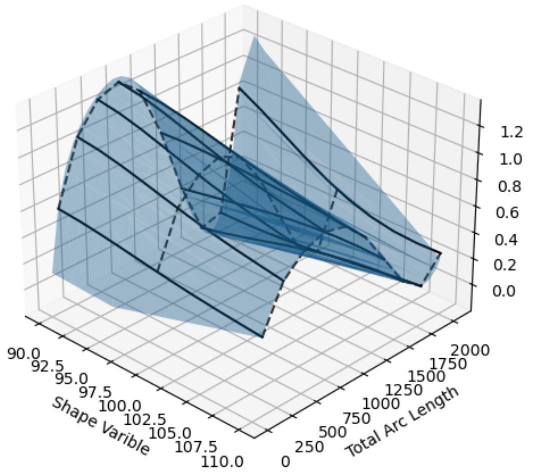


Figure 6.4. An Example of the NSM approximating an underlying function in blue. The solid black lines through the design variable domain indicate the constructed RBF models, while the dashed black lines through the pseudo-time domain indicate the interpolated final approximated curve.

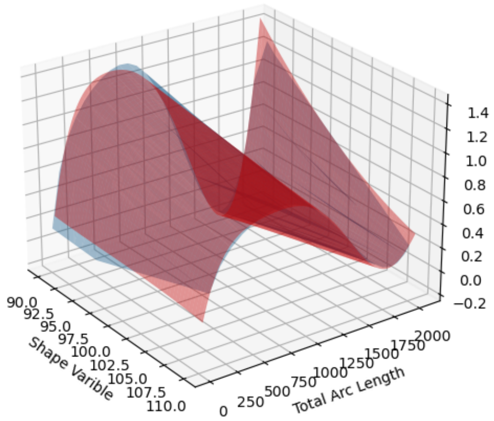


Figure 6.5. Example of an STSM in red approximating the underlying function in blue.

which can be written as

$$\mathbf{f} = \boldsymbol{\Phi}(\mathbf{x}, \mathbf{c}, \varepsilon)\mathbf{w}, \tag{6.3}$$

where $\boldsymbol{\Phi}$ is a $k \times p$ matrix where p is the number of samples and k the number of centres or basis functions. This matrix is then expressed as

$$\boldsymbol{\Phi} = \begin{bmatrix} \phi(\mathbf{x}_1, \mathbf{c}_1, \varepsilon) & \phi(\mathbf{x}_1, \mathbf{c}_2, \varepsilon) & \dots & \phi(\mathbf{x}_1, \mathbf{c}_k, \varepsilon) \\ \phi(\mathbf{x}_2, \mathbf{c}_1, \varepsilon) & \phi(\mathbf{x}_2, \mathbf{c}_2, \varepsilon) & \dots & \phi(\mathbf{x}_2, \mathbf{c}_k, \varepsilon) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(\mathbf{x}_p, \mathbf{c}_1, \varepsilon) & \phi(\mathbf{x}_p, \mathbf{c}_2, \varepsilon) & \dots & \phi(\mathbf{x}_p, \mathbf{c}_k, \varepsilon) \end{bmatrix}_{k \times p} . \tag{6.4}$$

The remaining parameters of the surrogate model include the number and locations of the centres $\mathbf{c}$ 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 then to either position the centres at the location of the sampled points or to redistribute the centres in the design domain. This research will demonstrate the benefit of redistributing the centres in the case of the STSM. This redistribution of the centres is completed using the same numerical strategy used to place the spatial samples, namely, the Latin Hyper-Cube method. When $p = k$ the matrix $\mathbf{\Phi}$ becomes square and the weight vector can be solved directly from the linear system in Equation (6.3).

6.3.3.1 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, it may fit more to the noise in the data than to the underlying function. Secondly, a full interpolation matrix in higher dimensional or densely sampled problems become prohibitively large to solve. At the same time, regression-based model can still offer useful results at a more reasonable computational cost. Therefore, this section offers regression-based derivations for the discussed surrogate models.

To implement the GE-RBF model the gradient of the Gaussian basis function is computed from

$$\frac{d\phi(\mathbf{x},\mathbf{c},\varepsilon)}{d\mathbf{x}} = -2\varepsilon\phi(\mathbf{x},\mathbf{c},\varepsilon)(\mathbf{x}-\mathbf{c}), \tag{6.5}$$

where Equation (6.5) 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:

$$\begin{bmatrix} \frac{df_1}{d\mathbf{x}} \\ \frac{df_2}{d\mathbf{x}} \\ \vdots \\ \frac{df_p}{d\mathbf{x}} \end{bmatrix} = \begin{bmatrix} \frac{d\phi(\mathbf{x}_1,\mathbf{c}_1,\varepsilon)}{d\mathbf{x}} & \frac{d\phi(\mathbf{x}_1,\mathbf{c}_2,\varepsilon)}{d\mathbf{x}} & \cdots & \frac{d\phi(\mathbf{x}_1,\mathbf{c}_k,\varepsilon)}{d\mathbf{x}} \\ \frac{d\phi(\mathbf{x}_2,\mathbf{c}_1,\varepsilon)}{d\mathbf{x}} & \frac{d\phi(\mathbf{x}_2,\mathbf{c}_2,\varepsilon)}{d\mathbf{x}} & \cdots & \frac{d\phi(\mathbf{x}_2,\mathbf{c}_k,\varepsilon)}{d\mathbf{x}} \\ \vdots & \vdots & \vdots & \\ \frac{d\phi(\mathbf{x}_p,\mathbf{c}_1,\varepsilon)}{d\mathbf{x}} & \frac{d\phi(\mathbf{x}_p,\mathbf{c}_2,\varepsilon)}{d\mathbf{x}} & \cdots & \frac{d\phi(\mathbf{x}_p,\mathbf{c}_k,\varepsilon)}{d\mathbf{x}} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_k \end{bmatrix}. \tag{6.6}$$

The linear system in Equation (6.6) is written as

$$\nabla\mathbf{f} = \mathbf{\Phi}_{fo}\mathbf{w}_{fo}. \tag{6.7}$$

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

$$\begin{bmatrix} \mathbf{f} \\ \nabla\mathbf{f} \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} \\ \mathbf{\Phi}_{fo} \end{bmatrix} \mathbf{w}_{GE}. \tag{6.8}$$

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 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, $\mathbf{w}_{GE}$, 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

$$\boldsymbol{\Phi}^T \boldsymbol{f}_{GE} = \boldsymbol{\Phi}^T \boldsymbol{\Phi} \boldsymbol{w}_{GE}. \tag{6.9}$$

6.4 Numerical Test Problem

The test function from the authors' previous work [34] is re-used here. This test function is deliberately created as a decomposable function

$$f(\mathbf{x}) = f_1(x_1) + f_2(x_2) + \cdots + f_n(x_n), \tag{6.10}$$

defined in the domain $x \in [0,1]$ where the final variable x_n is now considered a pseudo-time variable. The coordinate system is then rotated using a random rotation matrix $\mathbf{R}$ created from

$$\mathbf{R} = \text{expm}(\pi(\mathbf{A} - \mathbf{A}^\intercal)), \tag{6.11}$$

where $\mathbf{A}$ 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 $(\mathbf{A} - \mathbf{A}^\intercal)$ results in an orthogonal matrix [48]. This rotation of a decomposable function results in a transformed function that is now coupled, which is more representative of a practical engineering problem. Therefore, the results of the different pre-processing procedures, min-max scaling and the LHM transformation scheme, can be compared on a coupled function that can be uncoupled with a linear coordinate system transformation.

The test function is chosen to have the form

$$f(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{N} A_i \sin(F_i x_i), \tag{6.12}$$

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

$$A_i = -2 \exp \frac{-(2i-N)^2}{N} + 3, \tag{6.13}$$

$$F_i = \frac{3\pi}{2 + 2\exp \frac{-20i+N}{2}} + \frac{\pi}{2}. \tag{6.14}$$

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

In this research, the case where gradient information is available is discussed. Therefore, the gradients of the n-dimensional test function are needed. The gradient of Equation (6.12) is simply

$$\frac{\partial F_i}{\partial x_i} = A_i F_i \cos(F_i x_i), \tag{6.15}$$

and the gradient vector in the transformed coordinate systems is found using the procedure described by Bouwer et al. [34].

6.4.1 RMSE Results

The RMSE results are found using

$$\text{RMSE} = \sqrt{\frac{\sum_i^N (V_T^i - V_P^i)^2}{N}}, \tag{6.16}$$

where V_T is the target value and V_P is the predicted value from the model. The test set uses 10000 randomly placed points in the full spatio-temporal domain.

Eight different models are tested, four different NSMs with 15 models equally spaced throughout the pseudo-time domain, and four different STSMs. These models, and their abbreviations, include

- NSM with centres at the sample locations and min-max pre-processing (NSM).
- NSM with redistributed centres and min-max pre-processing (R-NSM).
- NSM with centres at the sample locations and the LHM pre-processing (NSM LHM).
- NSM with redistributed centres and the LHM pre-processing (R-NSM LHM).
- STSM with centres at the sample locations and min-max pre-processing (STSM).
- STSM with redistributed centres and min-max pre-processing (R-STSM).
- STSM with centres at the sample locations and the LHM pre-processing (STSM LHM).
- STSM with redistributed centres and the LHM pre-processing (R-STSM LHM).

To account for the variation in the randomness in both the sample locations and the redistributed centre locations, each model is constructed 50 times and the RMSE for each case is recorded.

The models are then tested on the 2-, 3-, 5-, and 9-dimensional problems. The last dimension is sampled as a pseudo-time variable with random step sizes between 0.05 and 0.1 to simulate a practical engineering problem. The results for the 2-dimensional problem are presented in Figures (6.6), (6.7) and (6.8) where 5 spatial samples are used. Figures (6.6) and (6.7) show curves for 6 randomly sampled spatial variables for the different NSM and STSM models overlaid with the target curve.

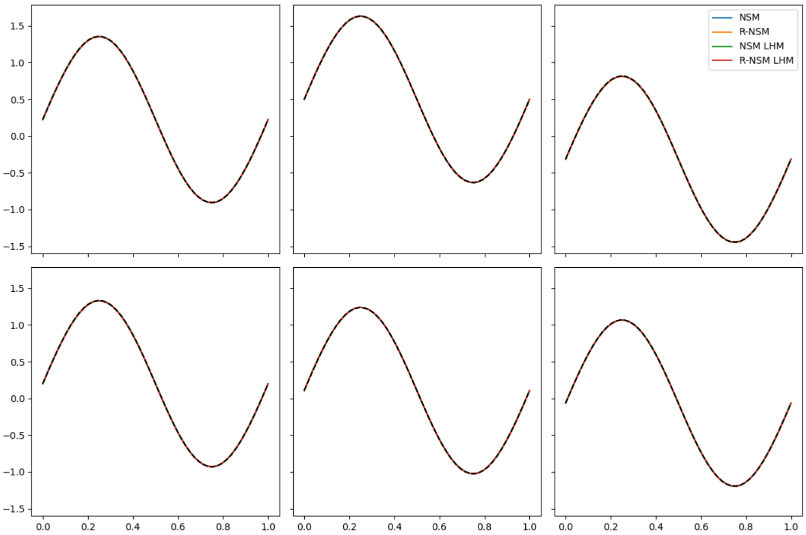


Figure 6.6. NSM Results for 2 variables using 5 samples.

At this low problem dimension, Figures (6.6) and (6.7) demonstrate that all the different model options already offer accurate approximations of the underlying function. Figure (6.8) shows the box and whisker plots of the RMSE results.

As shown in the random curve results (Figures (6.6) and (6.7)), all the models already have low RMSE results. The STSMs do outperform the NSMs at this problem dimension, mostly likely due to the accuracy loss when the results are interpolated to the network model locations in the pseudo-time variable. However, even at this high level of accuracy it is still clear that redistributing the centres and using the LHM pre-processing step noticeably improve the accuracy of the STSMs. In the case of the NSMs, redistributing the centres does not improve the model's accuracy, while the LHM transformation offers a small improvement.

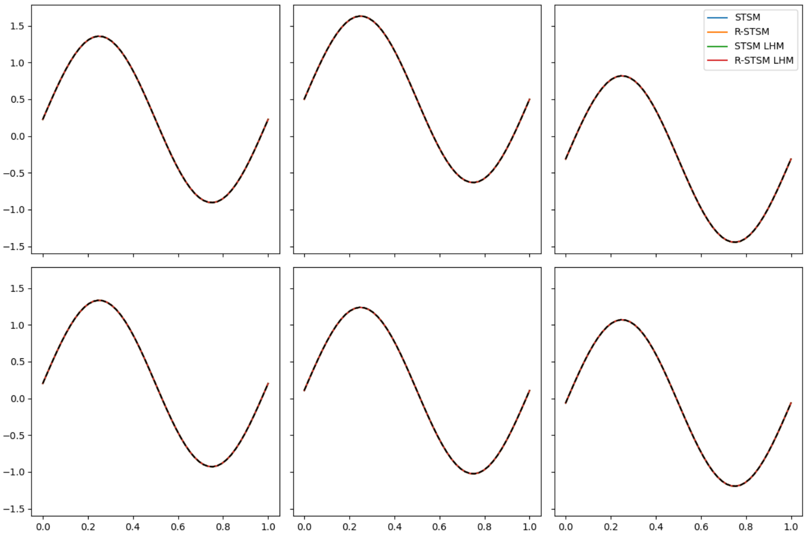


Figure 6.7. STSM Results for 2 variables using 5 samples.

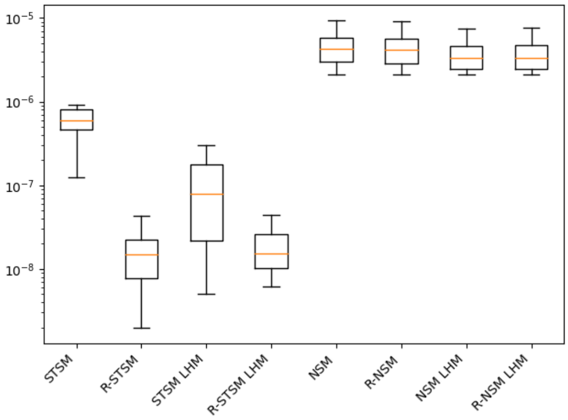


Figure 6.8. RMSE Results for all models for 2 variables and 5 samples.

The number of spatial dimensions is now increased to 2, resulting in a problem dimension of 3. The results are generated using 12 spatial samples. The six random curves are shown in Figures (6.9) and (6.10).

At this problem dimension the difference between standard min-max scaling and LHM pre-processing is more obvious. In the case of the NSMs, the redistributing of the centres does not offer any additional performance, and only once the LHM transformation scheme is implemented is there a notable improvement in accuracy regardless of the position of the centres.

For the STSMs the min-max scaling model with centres at the sample locations offers the worst approximation of the underlying function. At this problem dimension redistributing the centres or implementing the LHM drastically improves the models performance. These results are then further shown in the RMSE results in Figure (6.11).

The RMSE results for the three-dimensional problem show the benefit of the LHM preprocessing transformation for both the NSMs and STSMs. For the NSMs, redistributing centres do not offer any

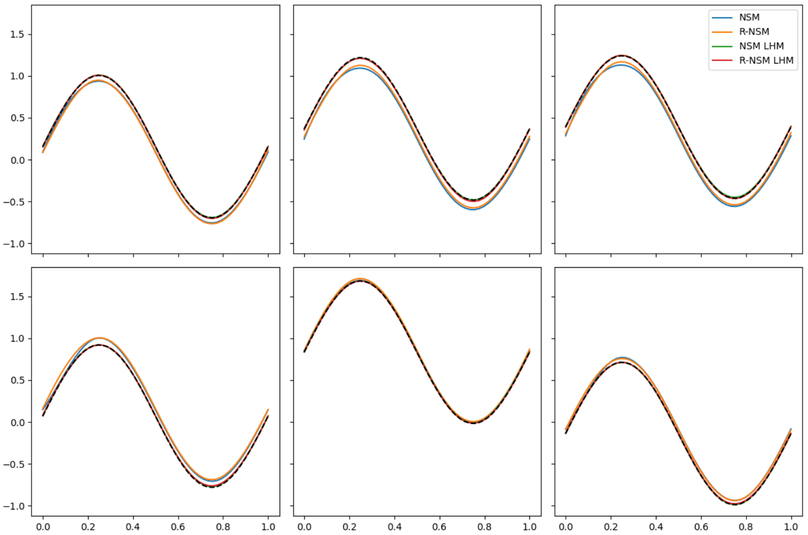


Figure 6.9. NSM Results for 3 variables using 12 samples.

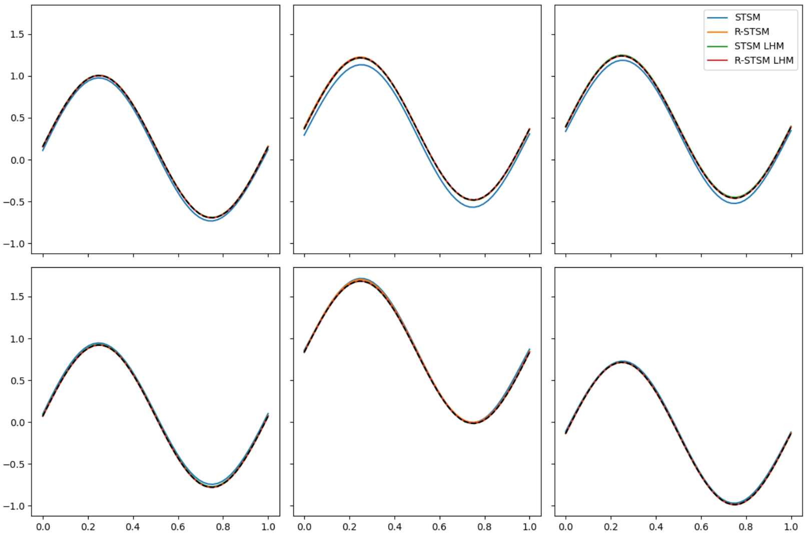


Figure 6.10. STSM Results for 3 variables using 12 samples.

improvement to the accuracy of the model, regardless of the pre-processing procedure. This is because, in the NSMs, there is no non-symmetry present in the sample locations, as only the locations of the samples in the spatial domain are used.

The STSMs see a massive improvement in accuracy, from 10^{-3} to 10^{-7}, when both the LHM pre-processing and the redistributing of the centre locations are implemented. This demonstrates that by addressing the anisotropy in both the underlining function and the sample locations, it is possible to gain a large improvement in performance without requiring additional samples to be generated.

The same results are repeated for the five and nine-dimensional problem using 35 and 135 spatial samples respectively. The six random sampled curves are repeated in Figures (6.12) and (6.13) for the five-dimensional problem, and in Figures (6.15) and (6.16) for the nine-dimensional problem.

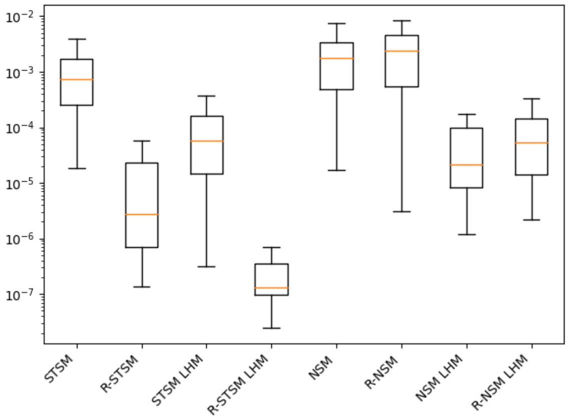


Figure 6.11. RMSE Results for all models for 3 variables using 12 samples.

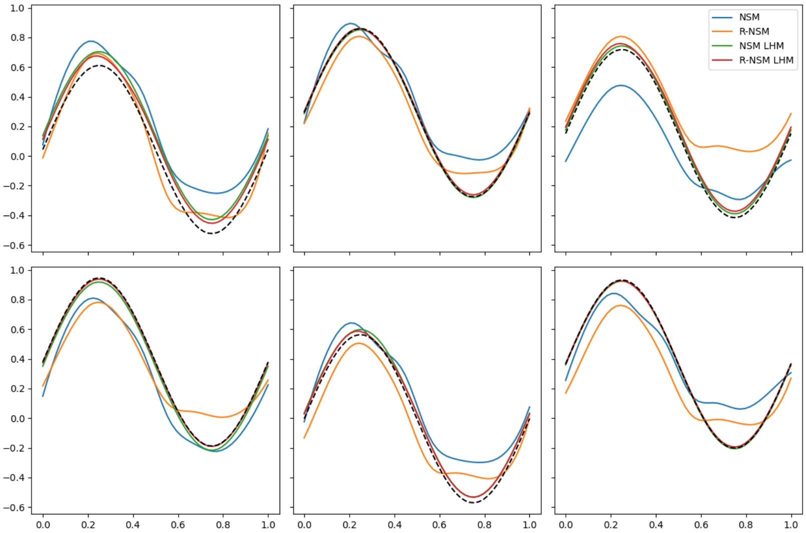


Figure 6.12. NSM Results for 5 variables using 35 samples.

The same trend in the results at the lower dimensions continues and grows at higher dimensions. Firstly, addressing the anisotropy in the underlying function provides a notable improvement to both the NSMs and the STSMs, regardless of the centre locations. The second result is that the model is further improved by redistributing the centres in the STSMs, while in the NSMs, there is no notable difference in RMSE results. This is expected as the sample locations are anisotropic only in the full spatio-temporal domain. The NSMs that are trained using only the locations of the samples in the spatial domain are not impacted by the difference in sampling density between the spatial and temporal variables.

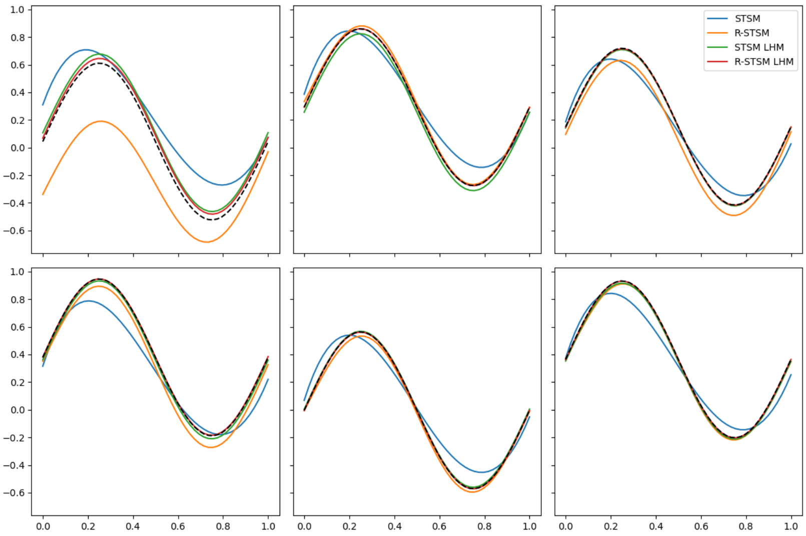


Figure 6.13. STSM Results for 5 variables using 35 samples.

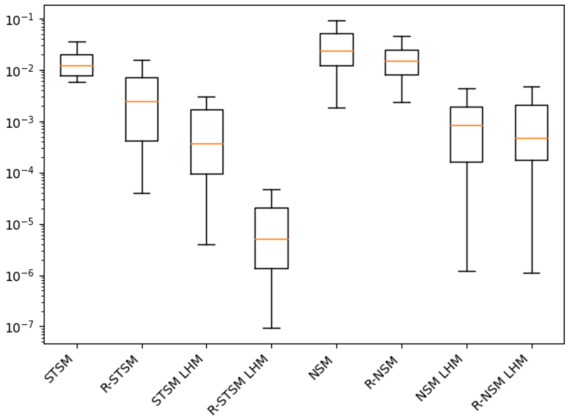


Figure 6.14. RMSE Results for all models for 5 variables using 35 samples.

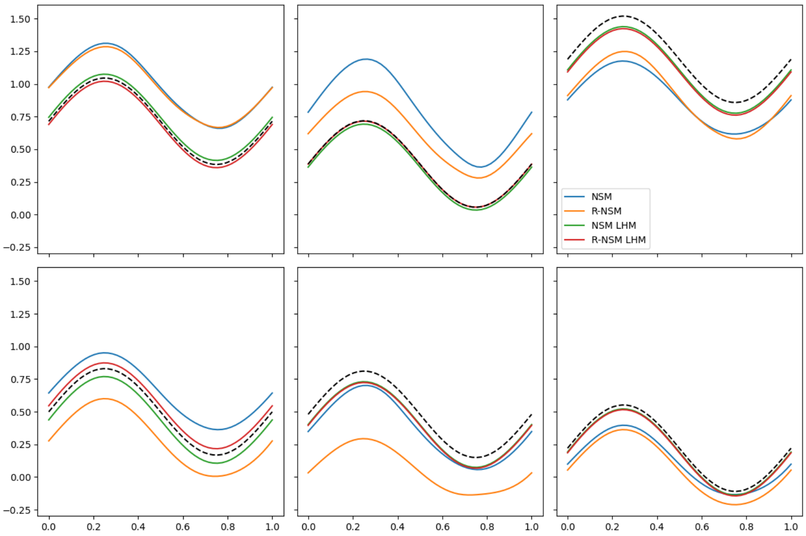


Figure 6.15. NSM Results for 9 variables using 135 samples.

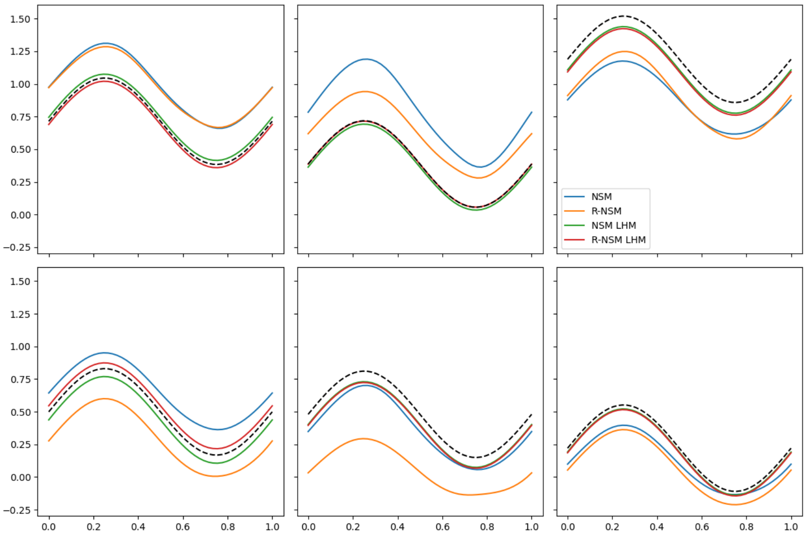


Figure 6.16. STSM Results for 9 variables using 135 samples.

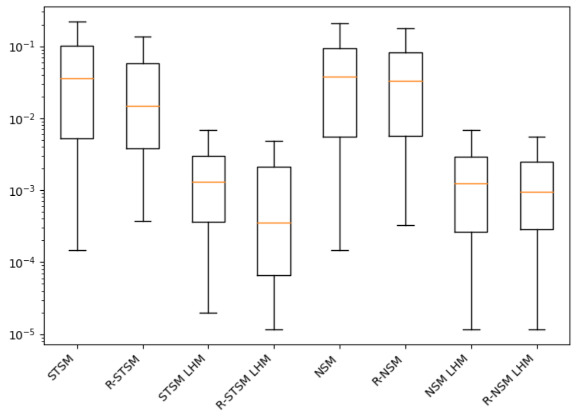


Figure 6.17. RMSE Results for all models for 9 variables using 135 samples.

6.5 Practical Example

The practical engineering example problem for this research is to find the shape of a snap-through structure that will exhibit a user-specified non-linear load path. This example structure will be the so-called Deep Semi-Circular Arch [10, 70]. The load-displacement path as well as the deformation overlay are depicted in Figure (6.18).

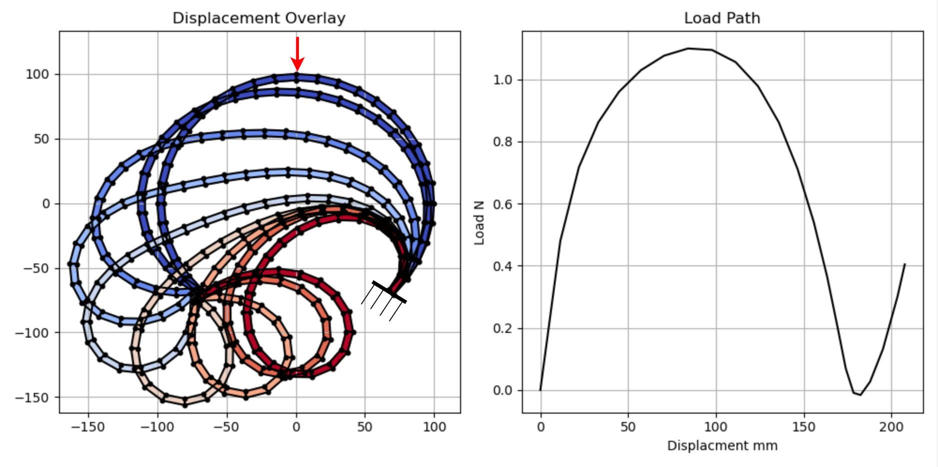


Figure 6.18. The deformation of the Deep Semi-Circular Arch that is pinned at the left and clamped on the right edge under a single-point load.

This structure is parameterised such that a vector of design variables describes the radii of the structure at equidistant points along the circumference. A cubic spline is then fitted through these radii, and the overall shape of a trial design is found. This is shown in Figure (6.19).

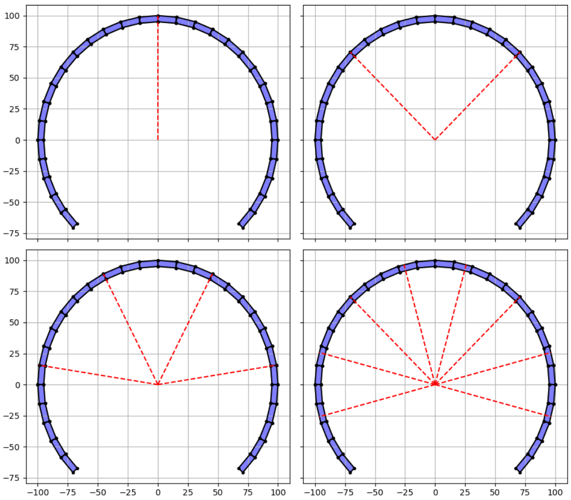


Figure 6.19. The design variables of the Deep Semi-Circular Arch depicted in red as the problem dimension increases.

To simulate the structure's behaviour, the finite element method (FEM) is coupled to the arc length control algorithm to fully trace the non-linear load path. A detailed description of this algorithm is found in the literature [13–15, 21], while the sensitivity procedure to sample the gradient vectors with respect to the shape variables is detailed in the authors' previous work [21]. The structures are simulated until an accumulated arc length of 2000 is reached, using 30 assumed stress elements [24] due to its accuracy in bending problems.

The author's previous work [21], it is noted that the DSCA is prone to bifurcation in its load path when the structure is nearly symmetrical. Therefore, the design problem is adapted to one where the most left radii between 90 and 110mm, while the remaining radii are between 80 and 90mm. The left and right extremes are fixed at 100mm. These modifications avoid bifurcating load paths.

The objective function of the problem is the RMSE metric given by

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i}^{n}(\lambda_i^{approx} - \lambda_i^{target})^2 + \frac{1}{n}\sum_{i}^{n}(u_i^{approx} - u_i^{target})^2}, \tag{6.17}$$

where the terms λ and u are the normalised load and displacement values, normalised using the maximum load and displacement values from the target curve respectively. The target curve is sampled from the exact centre of the domain, such that the global optimum is known and the performance of the different optimisation techniques can be compared. Each subsequent trial curve during the optimisation process is evaluated by interpolating the load and displacement values to the appropriate accumulated arc length position [21].

The performance of the 8 different surrogate models used to replace the FEM simulation from Section (6.4) is compared to the performance of two different optimisers used directly on the FEM simulation. The optimisers are the Sequential Least Squares Programming (SLSQP) method and the gradient-only sequential Spherical Approximation method (GOSSA) [28]. The SLSQP method is a well known optimiser and has been used on a wide variety of numerical optimization problems [30]. The GOSSA algorithm is a gradient-only method that is robust in handling the discontinuities present in this design problem [16–18,21]. These discontinuities arise from the need to interpolate the load and displacement values to standard accumulated arc length positions. The SLSQP algorithm is also implemented to solve the surrogate-based optimisation problem.

To begin the two-shape variable problem is solved. The target curve for this problem is shown in Figure (6.20). The 8 different surrogate models are all trained on the same 6 spatial samples. Figure (6.21) depicts four load paths randomly sampled from the design domain overlaid with the predicted load-displacement curves of the various surrogate models.

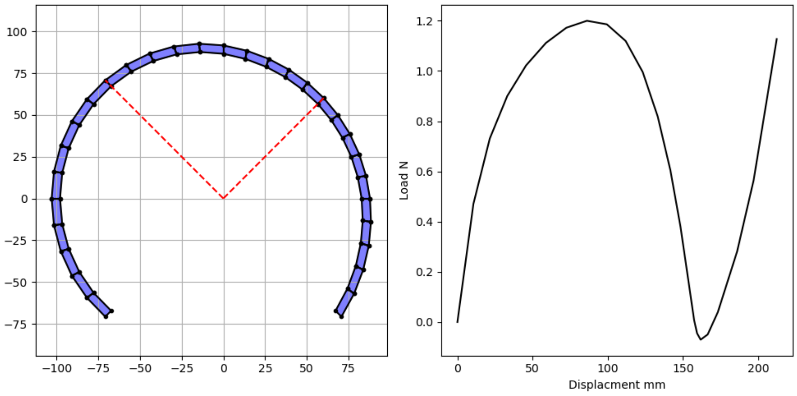


Figure 6.20. Target shape and load path for the 2 variable DSCAs shape optimisation problem.

Even at this low problem dimension of 2 shape variables, Figure (6.21) demonstrates the poor performance of STSMs if the anisotropic behaviour of the full spatio-temporal domain and the sample

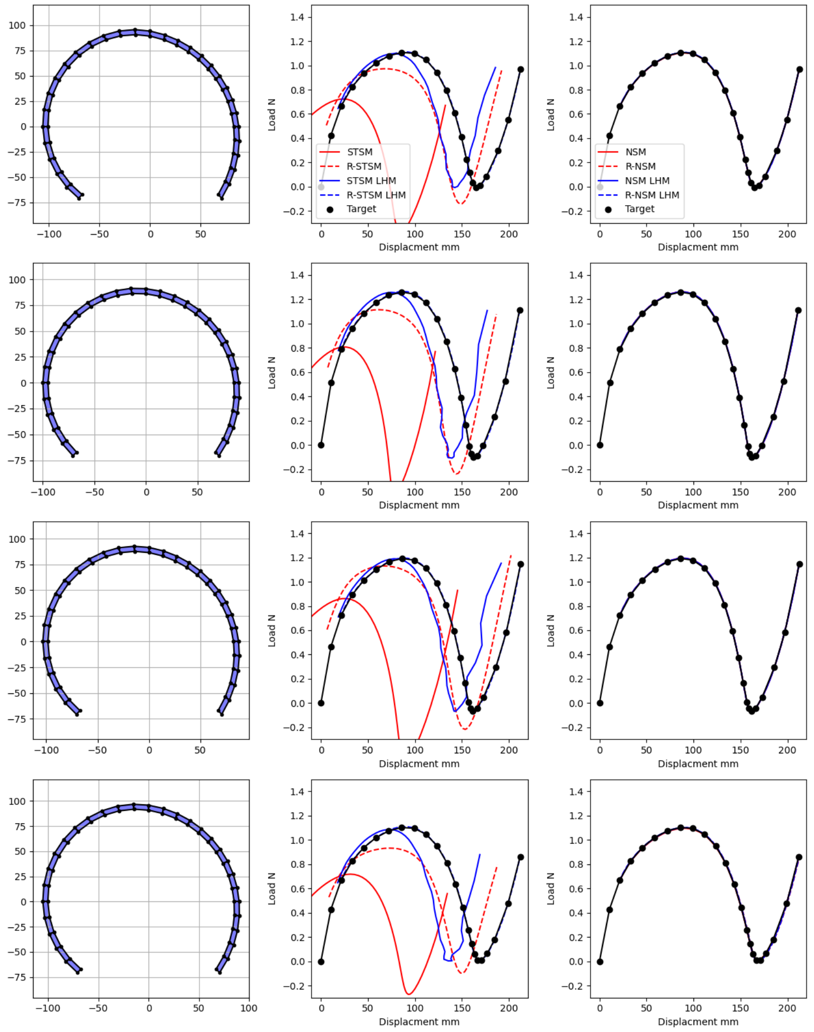


Figure 6.21. Predicted Load-Displacement Paths for the 2 variable DSCA problem using 4 randomly sampled DSCAs using 6 spatial samples.

locations are not adequately addressed. The models not addressing these anisotropic characteristics offer a poor approximation of the underlining function.

The final optimised load path of the different surrogate-based optimisation strategies are shown in Figures (6.22) and (6.23) for the STSMs and NSMs respectively. In contrast the best result for SLSQP and GOSSA for the simulation in the loop strategy using 10 random starting positions is shown in Figure (6.24). Each approximated curve is labelled with the error measure e defined by

$$e = ||\mathbf{x}_t - \mathbf{x}_{opt}||_2^2, \tag{6.18}$$

where the 2-norm between $\mathbf{x}_t$, the known target variable vector, and $\mathbf{x}_{opt}$, the returned optimum variable vector, is calculated.

The SLSQP and GOSSA algorithms required 7 and 22 simulations to find their optima respectively. The result of the SLSQP algorithm again demonstrates the negative impact of the discontinuities present in the objective function as the algorithm terminates prematurely after incorrectly interpreting a discontinuity as a local minimum [21]. The surrogate-based optimisation techniques in the case of NSMs required only 6 samples to find the global optimum within a tolerance of 10^{-6}. In contrast, the

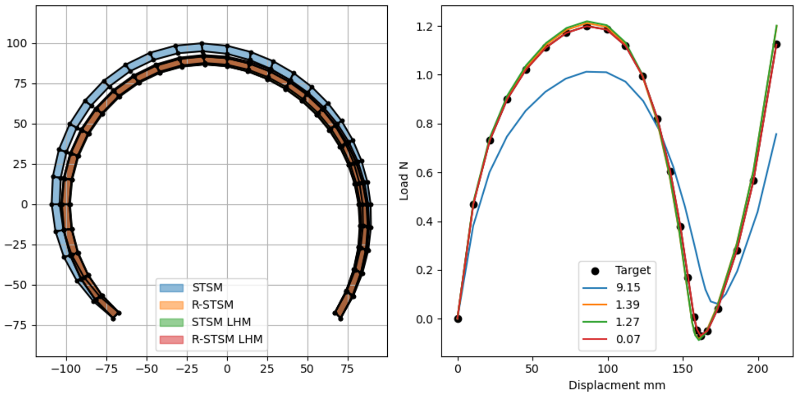


Figure 6.22. The shape of the final 2 variable optimised DSCAs and their load paths using surrogate-based optimisation provided by different STSMs.

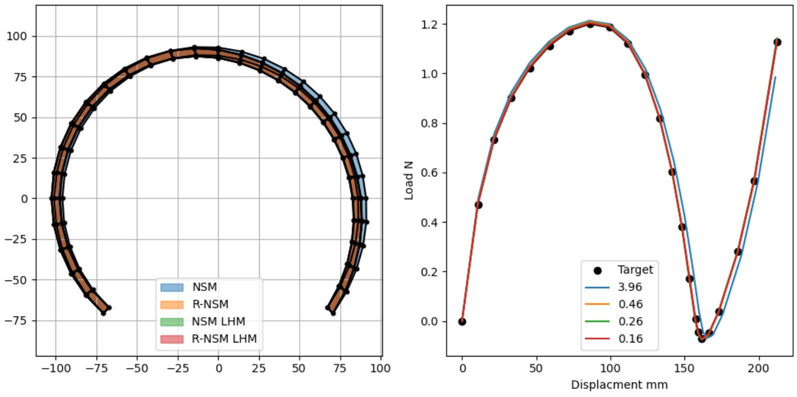


Figure 6.23. The shape of the final 2 variable optimised DSCAs and their load paths using surrogate-based optimisation provided by the different NSMs.

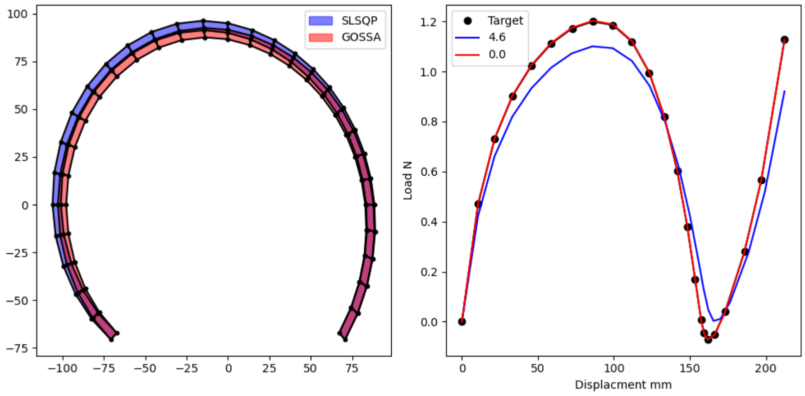


Figure 6.24. The shape of the final 2 variable optimised DSCAs and their load paths provided by the SLSQP and GOSSA algorithms using simulation in the loop.

case of the STSM, only the transformed model with distributed centres found the global minima using the same 6 spatial samples.

To further investigate the performance of the STSMs, the results along the path that the SLSQP algorithm took through the design domain, using the surrogate models, are compared to the result if the FEM simulation was used instead. Specifically, at 3 random locations along the optimisation path the objective function value, the true function value f and the approximated value f_m, and the gradient vectors are stored. Three metrics are then calculated to assess the performance of the surrogate models, namely,

1. The ratio between the function values,
2. the ratio between the magnitudes of the gradient vectors,
3. the dot product between the normalised gradient vectors.

Using these metrics, the accuracy of the surrogate models can be demonstrated with respect to both function value information, the ratio between function values, and gradient information, the magnitude of the gradient vectors, as the direction of the gradient vectors. Ideally, the dot product between the vectors will be 1, meaning the directions are identical. In the worst case, it will be -1 when they point in opposite directions. These results are shown in Table (6.1) for the STSMs used in the two-shape variable problem.

Table 6.1. A table showing the ratio of the objective function, the ratio of the gradient vector magnitude, and the dot product of the gradient vectors between the actual value f and the approximated value f_m from the STSMs at 3 different locations along the optimization path in the design space for the 2 variable problem.

	Location 1			**Location 2**			**Location 3**		
Model	$\frac{f}{f_m}$	$\lvert\frac{df}{dx}\rvert / \lvert\frac{df_m}{dx}\rvert$	$\frac{df}{dx} \cdot \frac{df_m}{dx}$	$\frac{f}{f_m}$	$\lvert\frac{df}{dx}\rvert / \lvert\frac{df_m}{dx}\rvert$	$\frac{df}{dx} \cdot \frac{df_m}{dx}$	$\frac{f}{f_m}$	$\lvert\frac{df}{dx}\rvert / \lvert\frac{df_m}{dx}\rvert$	$\frac{df}{dx} \cdot \frac{df_m}{dx}$
STSM	0.05	2.08	-0.60	0.049	0.09	0.14	0.14	19.13	-0.63
R-STSM	0.16	1.62	0.86	0.11	1.29	0.10	0.10	0.83	0.07
STSM LHM	0.21	0.99	0.83	0.08	3.99	0.06	0.06	0.40	0.70
R-STSM LHM	**0.88**	**1.01**	**0.97**	**0.89**	**0.92**	**0.98**	**0.98**	**0.99**	**1.00**

The STSM models only offer accurate predictions of gradient and function information when the R-STSM LHM version is implemented. The other models offer poor predictions of the function value and can predict gradient vectors that point away from the actual gradient vector, undermining the implemented optimisation algorithm.

The results are then repeated for the four variable shape optimisation problem where the new target curve is shown in Figure (6.25).

The 8 different surrogate models are all trained on the same 12 spatial samples. Figure (6.26) again depicts four load paths randomly sampled from the design domain overlaid with the predicted load-displacement curves of the various surrogate models.

The same result for the 2 variable problem is repeated here in four dimensions. Specifically, the redistributed and transformed model is the only STSM that accurately approximates the load-displacement path.

As with the 2 variable problem the final optimised load path of the different surrogate-based optimisation strategies are shown in Figures (6.27) and (6.28) for the surrogate-based optimisation approach using the STSMs and NSMs respectively, while the best results for SLSQP and GOSSA implemented in the simulation in the loop strategy using 10 random starting positions are shown in Figure (6.29).

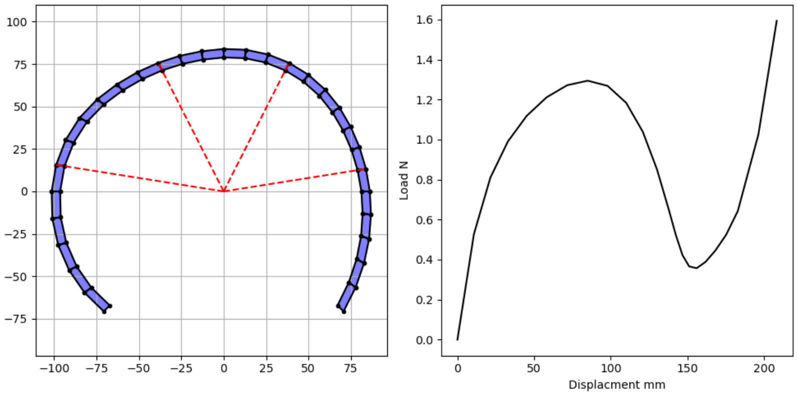


Figure 6.25. Target shape and load path for the 4 variable DSCAs shape optimisation problem.

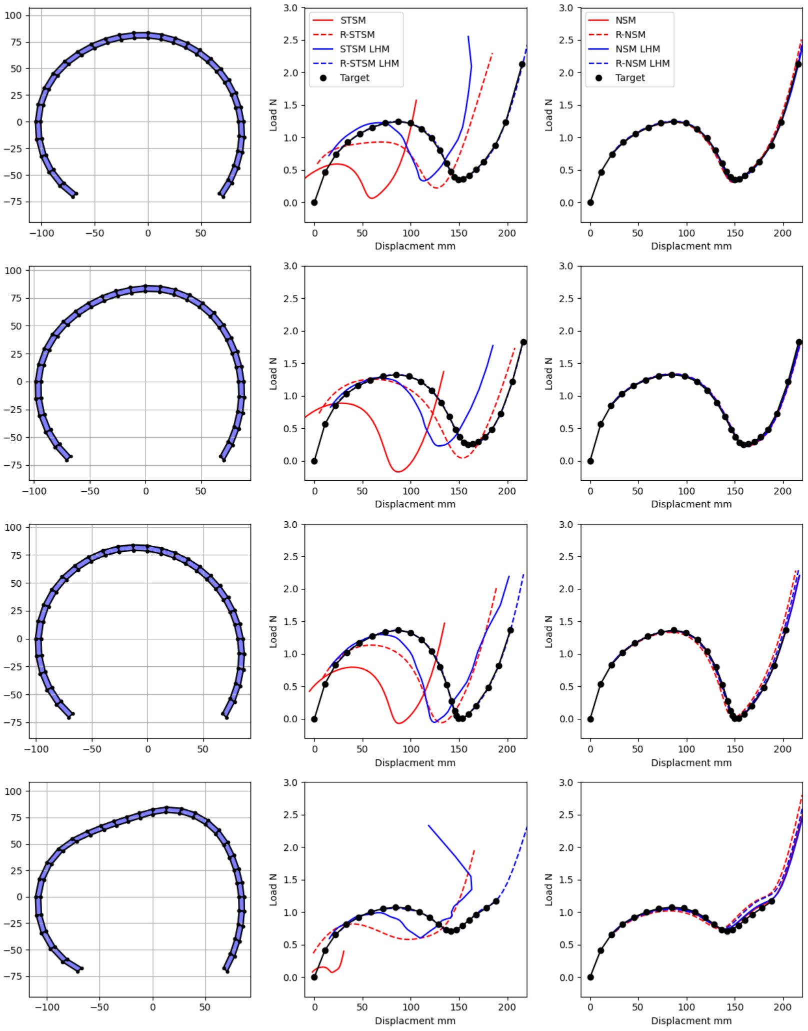


Figure 6.26. Predicted Load-Displacement Paths for the 4 variable DSCA problem for four randomly sampled DSCAs.

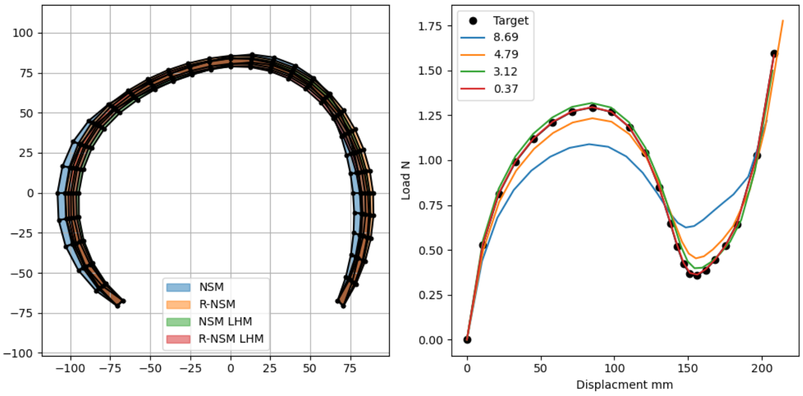


Figure 6.27. The shape of the final 4 variable optimised DSCAs and their load paths using surrogate-based optimisation provided by the different STSMs.

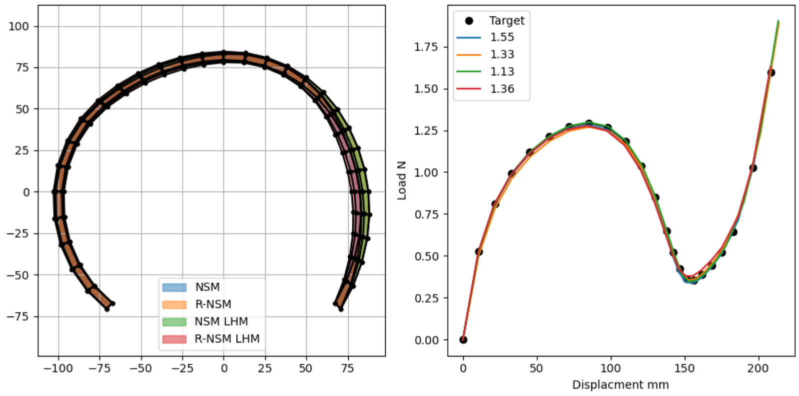


Figure 6.28. The shape of the final 4 variable optimised DSCAs and their load paths using surrogate-based optimisation provided by the different NSMs.

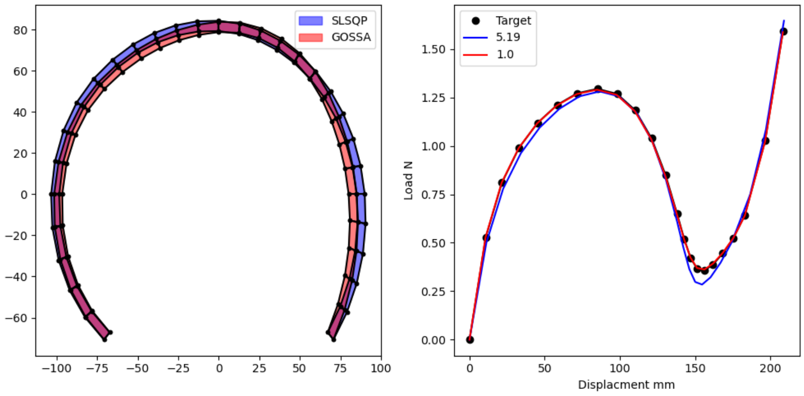


Figure 6.29. The shape of the final 4 variable optimised DSCAs and their load paths provided by the SLSQP and GOSSA algorithms using simulation in the loop.

The R-STSM LHM model once again found the global optimum while the other STSMs struggled to represent the underlying function accurately. The NSMs remain robust, and all the implementations find the global minima used to generate the target load-displacement path in Figure (6.25).

In the case where the FEM simulation is placed directly in the optimisation loop (Figure (6.29)) the SLSQP algorithm struggles to bypass the discontinuities in the objective function terminating after 37 simulations. The GOSSA algorithm remains robust enough to find the global optimum after 42 simulations. These results mean that the surrogate-based optimisers required 70% fewer simulations to find the global optimum, but *only* when the data is appropriately transformed to an isotropic coordinate system and the centres are redistributed in the case of the STSMs.

To further assess the STSMs the same metrics in Table (6.1) are repeated in Table (6.2). The R-STSM LHM model far outperforms the other models, offering a better approximation of the function and gradient information throughout the optimisation path.

Table 6.2. A table showing the ratio of the objective function, the ratio of the gradient vector magnitude, and the dot product of the gradient vectors between the actual value f and the approximated value f_m from the STSMs at 3 different locations along the optimization path in the design space for the 4 variable problem.

	Location 1			**Location 2**			**Location 3**		
Model	$\frac{f}{f_m}$	$\lvert\frac{df}{dx}\rvert / \lvert\frac{df_m}{dx}\rvert$	$\frac{df}{dx} \cdot \frac{df_m}{dx}$	$\frac{f}{f_m}$	$\lvert\frac{df}{dx}\rvert / \lvert\frac{df_m}{dx}\rvert$	$\frac{df}{dx} \cdot \frac{df_m}{dx}$	$\frac{f}{f_m}$	$\lvert\frac{df}{dx}\rvert / \lvert\frac{df_m}{dx}\rvert$	$\frac{df}{dx} \cdot \frac{df_m}{dx}$
STSM	0.67	0.98	-0.56	0.21	4.91	-0.75	0.17	0.16	0.78
R-STSM	0.34	1.95	-0.07	0.35	4.31	0.72	0.07	0.88	0.26
STSM LHM	0.05	0.77	-0.08	0.04	0.21	0.24	0.15	1.51	-0.85
R-STSM LHM	**1.10**	**1.02**	**0.99**	**0.99**	**1.38**	**0.90**	**0.88**	**0.97**	**0.99**

Lastly, the 8-shape variable optimisation problem is completed using 25 spatial samples. The new target curve is shown in Figure (6.30) and the four randomly sampled load paths are depicted in Figure (6.31). For the 8 variable problem highly complex and non-linear paths are possible in this shape optimisation problem.

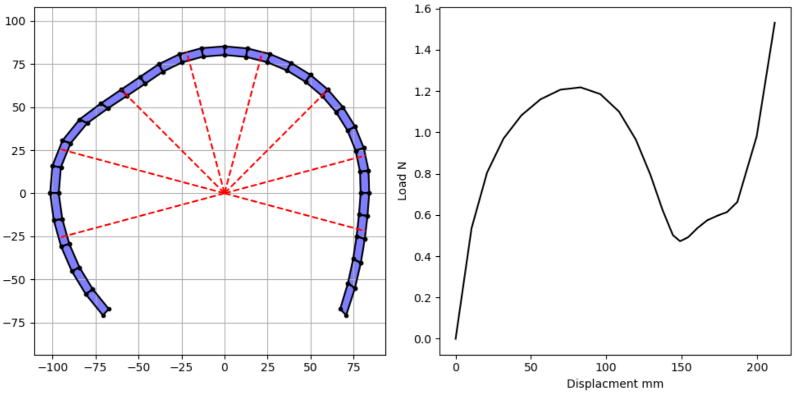


Figure 6.30. Target shape and load path for the 8 variable DSCAs shape optimisation problem.

The R-STSM LHM option remains the STSM that best approximates the underlying function of all the various STSM implementations. As with the lower dimensional problems, all the NSMs offer reasonable approximations of the load paths as they, by construction, do not have to accommodate the anisotropy present in the sample locations in the full spatio-temporal domain. Instead, as they are only constructed in the spatial domain, the RBF kernel is better suited to how the samples are distributed.

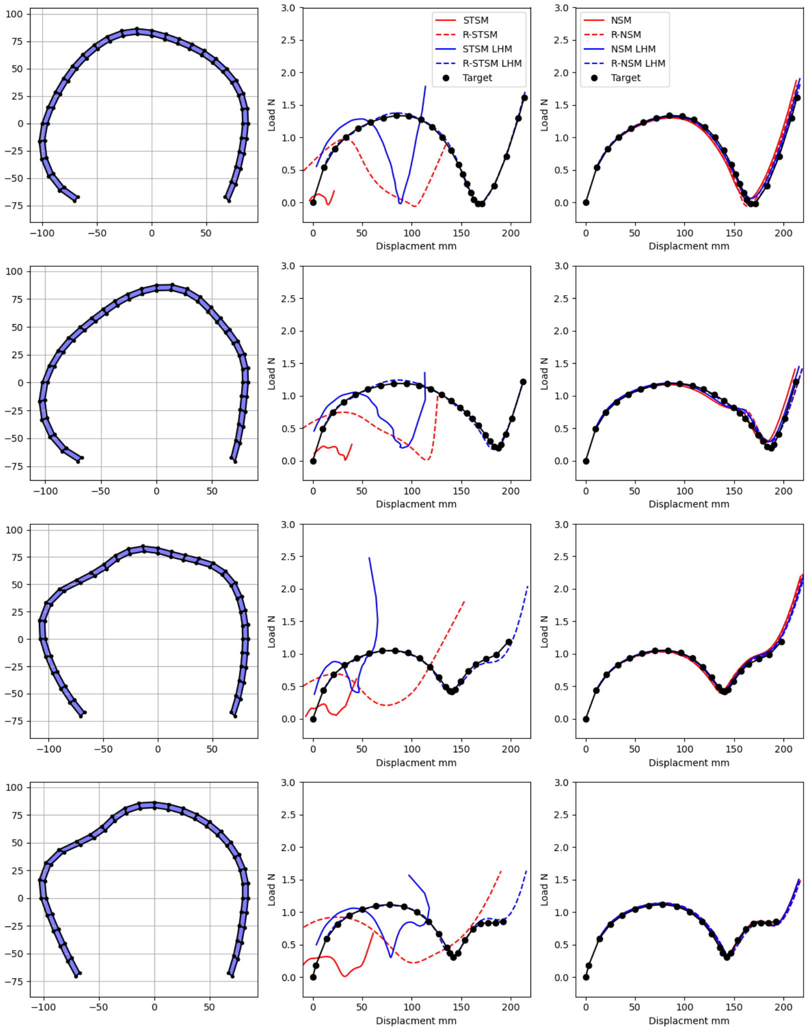


Figure 6.31. Predicted Load-Displacement Paths for the 8 variable DSCA problem.

The results from the implemented optimisation strategies are shown below. Figures (6.32) and (6.33) for the surrogate-based optimisation approach using the STSMs and NSMs respectively, while the best results for SLSQP and GOSSA implemented in the simulation in the loop strategy using 10 random starting positions are shown in Figure (6.34).

From the results for the 2-, 4-, and 8-shape variable problems, the benefit of appropriately transforming the domain and redistributing the centres grows as the problem dimension increases. For both the STSMs and the NSMs, only the models constructed with the LHM pre-processing procedure can find the global optimum. In contrast, the other min-max scaled models cannot represent the underlying function accurately enough to complete meaningful shape optimisation.

The same discontinuities present in the lower dimensional problems are aggravated in higher dimensional problems. This can be seen from the poor optimum result returned from the SLSQP algorithm, were it terminated after only 21 simulations. The GOSSA algorithms still finds the global optimum, although it required 84 simulations. This means that appropriately constructed surrogate models, in this problem, required approximately 70% less computational resources to find the global minimum.

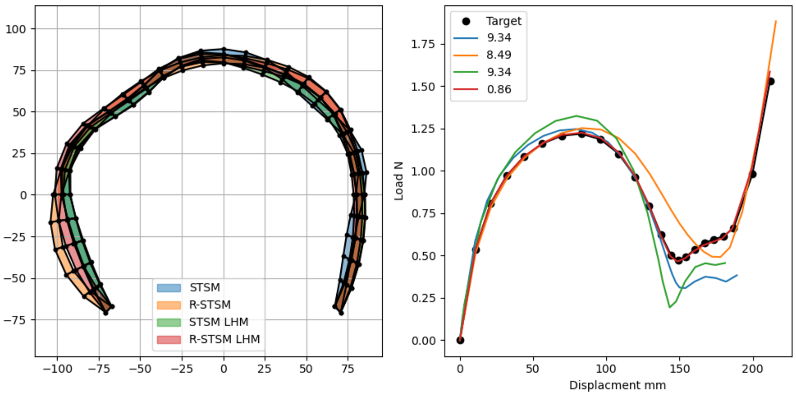


Figure 6.32. The shape of the final 8 variable optimised DSCAs and their load paths using surrogate-based optimisation provided by the different STSMs.

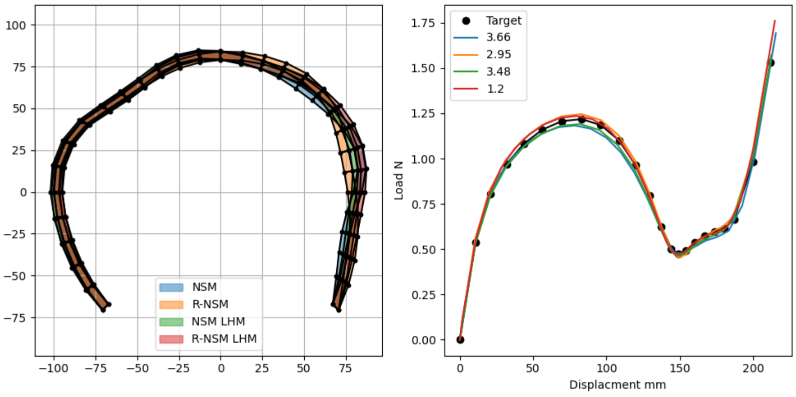


Figure 6.33. The shape of the final 8 variable optimised DSCAs and their load paths using surrogate-based optimisation provided by the different NSMs.

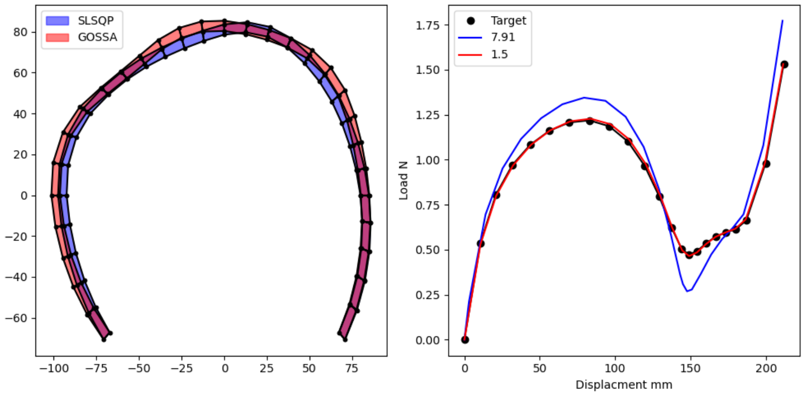


Figure 6.34. The shape of the final 8 variable optimised DSCAs and their load paths provided by the SLSQP and GOSSA algorithms using simulation in the loop.

The metrics in Table (6.3) further highlight the superior accuracy of the R-STSM LHM over the other STSMs. Therefore, by appropriately using gradient information and the redistribution of centres, STSMs become a viable and efficient strategy to solve an optimisation problem where a desired user-specified response through some pseudo-time is being designed.

Table 6.3. A table showing the ratio of the objective function, the ratio of the gradient vector magnitude, and the dot product of the gradient vectors between the actual value and the approximated value from the STSMs at 3 different locations along the optimization path in the design space for the 8 variable problem.

	Location 1			Location 2			Location 3		
Model	$\frac{f}{f_m}$	$\lvert\frac{df}{dx}\rvert / \lvert\frac{df_m}{dx}\rvert$	$\frac{df}{dx} \cdot \frac{df_m}{dx}$	$\frac{f}{f_m}$	$\lvert\frac{df}{dx}\rvert / \lvert\frac{df_m}{dx}\rvert$	$\frac{df}{dx} \cdot \frac{df_m}{dx}$	$\frac{f}{f_m}$	$\lvert\frac{df}{dx}\rvert / \lvert\frac{df_m}{dx}\rvert$	$\frac{df}{dx} \cdot \frac{df_m}{dx}$
STSM	0.052	2.33	-0.01	0.01	0.07	0.29	0.86	3.27	-0.22
R-STSM	0.072	2.64	0.32	0.06	3.04	-0.04	0.06	2.04	0.13
STSM LHM	0.02	0.43	-0.32	0.04	0.53	0.63	0.18	1.31	0.11
R-STSM LHM	**0.98**	**1.20**	**0.72**	**0.94**	**0.85**	**0.78**	**1.00**	**1.03**	**0.99**

6.6 Chapter Conclusion

The work presented in this chapter demonstrates that if appropriate pre-processing steps are taken, spatio-temporal surrogate models become a viable and efficient surrogate-based optimisation technique. These pre-processing steps must address anisotropic characteristics in both the underlying function and the sample locations in the full spatio-temporal domain.

Although the NSMs address the anisotropic sample locations, the numerical problem in Section (6.4) demonstrates that these models still benefit from implementing the LHM transformation scheme. These models do require the interpolation of the data, meaning both function information and gradient vector information need to be available at the required m locations in the pseudo-time domain, and then the construction and training of all m RBF models needs to be completed separately. The designer must, therefore, decide on the location and number of the m models, and these heuristics will be problem-specific. Therefore, STSM is a more convenient model to implement when compared to the NSM, as the STSM requires fewer heuristics to be tuned.

A numerical test problem and an engineering shape optimisation problem used in this chapter show that the STSM can be an efficient and powerful optimisation tool, only if necessary steps are taken to address the isotropic assumption in the RBF kernel. These steps need to address the anisotropy in function behaviour as spatial and temporal variables are evolved, as well as the anisotropy in the sample locations in the full spatio-temporal domain. In this work these steps are to firstly, perform a coordinate system transformation based off gradient information, and then secondly, to redistribute the centres of the isotropic kernel throughout the full spatio-temporal domain.

Chapter 7 Conclusion and Future Work

This chapter briefly highlights the main findings and contributions of this thesis. It then offers some insight into the possible future applications of the work as well as possible future research avenues.

The objective of the research documented in this thesis is to develop methodology that can complete meaningful and efficient shape optimisation of the highly non-linear load paths present in snap-through compliant mechanisms. The developed methodology is assessed and motivated in the context of generality, robustness, and computational efficiency. These criteria ensure that the work in this study is capable of simulating and designing a wide range of complex non-linear load paths, that it will consistently locate the optimum solutions in the design domain, and that computational resources are used as efficiently as possible.

To simulate the highly non-linear behaviour present in this design problem, the implementation of the arc length control algorithm is required. This solver is capable of tracing complex load paths with multiple limit and equilibrium points, allowing for the development of a robust and general optimisation procedure. To improve the computational efficiency of this algorithm is allowed to make adjustment of solution step sizes as a function of the complexity of the load path during the simulation. The by-product of this adaptive step is that any crafted objective function that quantifies the discrepancy between a target and trail load path will experience unavoidable discontinuities. These discontinuities occur as the number and the locations of the solution points along the load path will differ at different locations in the design domain.

These discontinuities mean that popular optimisation algorithms can terminate prematurely by misinterpreting the sudden discontinuous increases in the objective function as local minima. Therefore, to bypass these discontinuities and find true optimum designs, the implementation of non-negative gradient projection points with gradient-only optimisers is required. The results of gradient-only techniques show that complex interpolation strategies to eliminate the discontinuities are an unnecessary complication to the design problem.

To efficiently implement the gradient-only optimisers, an analytical shape sensitivity procedure is developed. This analytical sensitivity procedure means that computationally expensive numerical methods to obtain gradient information, such as forward difference or complex step, are not needed. The computational cost of these methods grow with the dimensionality of the design problem, where as the developed analytical procedure remains relatively insensitive to the number of design variables allowing for higher dimensional problems to be completed. At this stage in the research the shape optimisation framework can complete the design of highly non-linear load paths for a wide range of possible load paths, and consistently find the optimum solutions in the design space.

The next step is to improve the efficiency of the proposed optimisation framework by eliminating the sequential manner in which the numerical simulations are completed. Instead, surrogate models can be constructed with results found by running the simulations in parallel, and then replacing the expensive numerical simulation with the computationally inexpensive surrogate model. The focus now shifts to ensuring these models can accurately approximate the load path as a function of the design space, as the accuracy of the models directly impact the quality of the returned optimum design. Two modelling strategies are considered, namely, spatio-temporal models and network models.

Two sources of inaccuracy in the models are addressed. The first is the highly anisotropic data manifold present in the design problem. This anisotropy arises from the wide range of possible shape or spatial variables and their distinct impact on the load path, as well as the distinct difference in behaviour of the load path as a function of either the shape variables or the arc length variable. This anisotropy is detrimental to the accuracy of surrogate models constructed with isotropic kernels commonly used in research. This anisotropy is addressed with the development of a novel coordinate system transformation scheme that attempts to find an isotropic reference frame for the problem. This transformation scheme makes use of sampled gradient information to make local estimates of curvature and from these local estimates complete a single global linear transformation. The method is shown to greatly improve the accuracy of models, and the benefit to accuracy grows with the dimensionality of the problem.

The second source of inaccuracy is the anisotropy present in the distinct methods used to sample the shape and arc length variables. In the case of spatio-temporal models, the anisotropy is alleviated by redistributing the centre or kernel locations throughout the full spatio-temporal domain. The network models avoid this source of anisotropy by only constructing the models as a function of the spatial variables at predetermined locations in the temporal domain.

By implementing these models on the original shape optimisation problem, the benefit of the developed methods is demonstrated. The results demonstrate that only once both sources of anisotropy, functional and sampling, are addressed are the spatio-temporal models a useful addition to the shape optimisation problem. The newly developed surrogate modelling strategy requires far fewer computational resources to find optimum solutions compared to the initial simulation in the loop strategy.

Therefore, this thesis develops a general, efficient and robust methodology to complete the shape optimisation of highly non-linear load paths. In the author's opinion, techniques and conclusions with wide-reaching applications and consequences are developed during the development of this methodology.

Firstly, in both the direct and surrogate-based strategies, it is shown that gradient vectors are useful and necessary to complete practical and meaningful optimisation. Specifically, to bypass discontinuities in the case of direct optimisation and to complete a powerful coordinate system transformation scheme for isotropic kernel-based surrogate models. Therefore, the developed analytical sensitivity procedure is most certainly required in any numerical simulation that uses the arc length control method if any design or optimisation is to be completed.

Second, the developed transformation scheme demonstrates that gradient information is a greatly beneficial addition to isotropic kernel-based surrogate models if they are used to estimate an isotropic coordinate system and not simply naively included directly into the model. The author believes this will allow for the construction of accurate surrogate models in more applications than simply shape-optimisation problems such as environmental modelling, robotics and control systems, biomedical applications, or any problem where complex relationships need to be modelled and optimised efficiently. The developed procedure is a general method that requires no tuning of heuristics and is applicable in any problem where obtaining gradient information is possible.

Lastly, the ability of the transformation technique to improve the accuracy of the models grows with the dimensionality of the problem. As the number of variables in the problem grows, the likelihood of the underlying function being anisotropic and the severity of the anisotropy increase. In the author's opinion, this leaves the potential for creating far higher-dimensional models than what is currently used in literature. This is a possibility as the poor performance of the high dimensional models may have been incorrectly attributed to the sparsity of information in higher dimensions, i.e. the curse of dimensionality, but may have been due to the severe anisotropic functions inherently present in high dimensional problems.

The completed research opens many avenues of possible research opportunities, such as addressing the bifurcation present in the load paths of snap-through structures, implementing the developed methods on full three-dimensional problems, higher-dimensional problems, and continuing the research into surrogate models.

In the author's opinion, the most impactful area of research will be the continued development of the foundational transformation scheme. The current transformation technique completes a single linear transformation after finding an average of many transformations and is most impactful on decomposable problems where a linear transformation is all that is required to find isotropy.

The first option to expand the transformation scheme to non-decomposable functions is by constructing many local models, each with it's own transformed coordinate system. An example of this methodology is completed in [71], where every time the model is sampled a collection of nearest neighbours are used to construct a new model. This methodology is typically implemented in the cases where the problem has a large dataset or is highly dimensional and, therefore, computational memory may be a bottle neck. If the assumption is made that a non-decomposable function can be represented as a decomposable function for a sufficiently small domain, the developed transformation scheme can find a local isotropic coordinate system using $n+1$ nearest neighbours for each new local model.

The other, more complex option is to find a single non-linear transformation of the entire coordinate system that fits all the found local transformations. This non-linear version of the transformation scheme will be completely general and applicable to all decomposable and non-decomposable problems in which an isotropic kernel-based surrogate model is constructed. However, finding this single transformation may be very computationally expensive, and the benefit of needing fewer samples to construct accurate models may be offset by the computational cost of finding a fully non-linear transformation scheme.

References

[1] A. Bhattacharyya, C. Conlan-Smith, and K. A. James, “Design of a Bi-stable Airfoil with Tailored Snap-through Response Using Topology Optimization,” *CAD Computer Aided Design*, vol. 108, pp. 42–55, 2019. [Online]. Available: https://doi.org/10.1016/j.cad.2018.11.001

[2] H. Deng, L. Cheng, X. Liang, D. Hayduke, and A. C. To, “Topology optimization for energy dissipation design of lattice structures through snap-through behavior,” *Computer Methods in Applied Mechanics and Engineering*, vol. 358, p. 112641, 2020. [Online]. Available: https://doi.org/10.1016/j.cma.2019.112641

[3] T. Sekimoto and H. Noguchi, “Homologous Topology Optimization in Large Displacement and Buckling Problems,” *JSME International Journal*, vol. 44, no. 4, pp. 616 – 622, 2001.

[4] T. E. Bruns, O. Sigmund, and D. A. Tortorelli, “Numerical methods for the topology optimization of structures that exhibit snap-through,” *International Journal for Numerical Methods in Engineering*, vol. 55, no. 10, pp. 1215–1237, 2002.

[5] T. E. Bruns and O. Sigmund, “Toward the topology design of mechanisms that exhibit snap-through behavior,” *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 36-38, pp. 3973–4000, 2004.

[6] C. D. Vu Khac Ky, “Surrogate-based methods for black-box optimization,” *International Transactions in Operational Research*, vol. 24, no. 3, pp. 393–424, 2019.

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

[8] K. Cheng, Z. Lu, C. Ling, and S. Zhou, “Surrogate-assisted global sensitivity analysis: an overview,” *Structural and Multidisciplinary Optimization*, vol. 61, no. 3, pp. 1187–1213, 2020.

[9] F. A. Viana, C. Gogu, and T. Goel, “Surrogate modeling: tricks that endured the test of time and some recent developments,” *Structural and Multidisciplinary Optimization*, vol. 64, no. 5, pp. 2881–2908, 2021.

[10] D. DaDeppo and E. Schmidt, “Instability of clamped-hinged circular arches subjected to a point load,” *Transactions of the American Society of Mechanical Engineers, Journal of Applied Mechanics*, pp. 894—896, Dec. 1975.

[11] I. Leahu-Aluas and F. Abed-Meraim, “A proposed set of popular limit-point buckling benchmark problems,” *Structural Engineering and Mechanics*, vol. 38, no. 6, pp. 767–802, 2011.

[12] E. Riks, “The application of Newton’s method to the problem of elastic stability,” *Journal of Applied Mechanics, Transactions ASME*, vol. 39, no. 4, pp. 1060–1065, 1972.

[13] M. Ritto-Corrêa and D. Camotim, “On the arc-length and other quadratic control methods: Established, less known and new implementation procedures,” *Computers and Structures*, vol. 86, no. 11-12, pp. 1353–1368, 2008.

[14] M. A. Crisfield, “A fast incremental/iterative solution procedure that handles “snap-through”,” *Computers & Structures*, vol. 13, no. 1, pp. 55–62, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0045794981901085

[15] M. Bashir-ahmed and S. U. Xiao-zu, “Arc-length technique for nonlinear finite element analysis,” *Journal of Zhejiang University SCIENCE*, vol. 5, no. 5, pp. 618–628, 2004.

[16] D. Wilke, J. Snyman, S. Kok, and A. Groenwold, “Gradient-only approaches to avoid spurious local minima in unconstrained optimization,” *Optimization and Engineering*, vol. 14, 06 2011.

[17] D. Wilke, “Approaches to accommodate remeshing in shape optimization,” Ph.D. dissertation, University of Pretoria, 08 2010.

[18] D. Wilke, S. Kok, and A. Groenwold, “Relaxed error control in shape optimization that utilizes remeshing,” *International Journal for Numerical Methods in Engineering*, vol. 94, pp. 273–289, 04 2013.

[19] T. Hisada, “Recent Progress in Nonlinear FEM-Based Sensitivity Analysis,” *JSME International Journal*, vol. 38, no. 3, pp. 430 – 433, 1995.

[20] J. Parente and L. E. Vaz, “On evaluation of shape sensitivities of non-linear critical loads,” *International Journal for Numerical Methods in Engineering*, vol. 56, no. 6, pp. 809–846, 2003.

[21] J. M. Bouwer, S. Kok, and D. N. Wilke, “Challenges and solutions to arc-length controlled structural shape design problems,” *Mechanics Based Design of Structures and Machines*, pp. 1–32, jul 2021. [Online]. Available: https://doi.org/10.1080/15397734.2021.1950549

[22] Y. S. Ryu, M. Haririan, C. C. Wu, and J. S. Arora, “Structural design sensitivity analysis of nonlinear response,” *Computers and Structures*, vol. 21, no. 1-2, pp. 245–255, 1985.

[23] N. Olhoff and E. Lund, “Finite Element Based Engineering Design Sensitivity Analysis and Optimization,” Ph.D. dissertation, Aalborg University, 1995.

[24] T. H. Pian and K. Sumihara, “Rational approach for assumed stress finite elements,” *International Journal for Numerical Methods in Engineering*, vol. 20, no. 9, pp. 1685–1695, 1984.

[25] D. N. Wilke, Kok, Schalk, and A. A. Groenwold, “The application of gradient-only optimization methods for problems discretized using non-constant methods,” *Structural and Multidisciplinary Optimization*, vol. 40, pp. 433–451, 2010.

[26] D. Wilke, “Modified subgradient methods for remeshing based structural shape optimization,” in *Proceedings of the 13th International Conference on Civil, Structural and Environmental Engineering Computing*, 2011.

[27] S. Kok and D. N. Wilke, “Optimizing snap-through structures by using gradient-only algorithms,” in *11th World Congress on Structural and Multidisciplinary Optimisation, Sydney, Australia*, 2015, pp. 7–12.

[28] J. A. Snyman and D. N. Wilke, *Practical Mathematical Optimization*, 2nd ed. Springer, 2018.

[29] D. Wilke, “Structural shape optimization using shor’s r-algorithm,” in *Third International Conference on Engineering Optimization*, 2012.

[30] D. Kraft, “A software package for sequential quadratic programming,” DLR German Aerospace Center – Institute for Flight Mechanics, Koln, Germany., Tech. Rep., 1988.

[31] D. N. Wilke and S. Kok, “Numerical sensitivity computation for discontinuous gradient-only optimization problems using the complex-step method,” *Proceedings of the Tenth World Congress on Computational Mechanics (WCCM 2012)*, vol. 1, pp. 3665–3676, 2014.

[32] H. B. Keller, “Constructive methods for bifurcation and nonlinear eigenvalue problems,” in *Computing Methods in Applied Sciences and Engineering, 1977, I*, R. Glowinski, J. L. Lions, and I. Laboria, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1979, pp. 241–251.

[33] W. C. Rheinboldt, “Numerical methods for a class of finite dimensional bifurcation problems,” *SIAM Journal on Numerical Analysis*, vol. 15, no. 1, pp. 1–11, 1978. [Online]. Available: https://doi.org/10.1137/0715001

[34] J. Bouwer, D. N. Wilke, and S. Kok, “A novel and fully automated coordinate system transformation scheme for near optimal surrogate construction,” *Computer Methods in Applied Mechanics and Engineering*, vol. 419, p. 116648, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045782523007715

[35] M. A. Bouhlel and J. R. Martins, “Gradient-enhanced kriging for high-dimensional problems,” *Engineering with Computers*, vol. 35, no. 1, pp. 157–173, 2019.

[36] D. J. Toal, N. W. Bressloff, and A. J. Keane, "Kriging hyperparameter tuning strategies," *AIAA Journal*, vol. 46, no. 5, pp. 1240–1252, 2008.

[37] M. A. Bouhlel, N. Bartoli, A. Otsmane, and 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*, vol. 2016, 2016.

[38] M. Urquhart, E. Ljungskog, and S. Sebben, "Surrogate-based optimisation using adaptively scaled radial basis functions," *Applied Soft Computing Journal*, vol. 88, p. 106050, 2020. [Online]. Available: https://doi.org/10.1016/j.asoc.2019.106050

[39] D. R. Jones, "A Taxonomy of Global Optimization Methods Based on Response Surfaces," *Journal of Global Optimization*, vol. 21, no. 4, pp. 345–383, 2001.

[40] S. Ulaganathan, I. Couckuyt, F. Ferranti, E. Laermans, and T. Dhaene, "Performance study of multi-fidelity gradient enhanced kriging," *Structural and Multidisciplinary Optimization*, vol. 51, pp. 1017–1033, 2015.

[41] I. C. Kampolis, E. I. Karangelos, and K. C. Giannakoglou, "Gradient-assisted radial basis function networks: Theory and applications," *Applied Mathematical Modelling*, vol. 28, no. 2, pp. 197–209, 2004.

[42] L. Laurent, R. Le Riche, B. Soulier, and P. A. Boucard, "An Overview of Gradient-Enhanced Metamodels with Applications," *Archives of Computational Methods in Engineering*, vol. 26, no. 1, pp. 61–106, 2019.

[43] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 1979.

[44] G. Dhondt and K. Wittig, "Calculix," 1988.

[45] V. Komkov, K. K. Choi, E. J. Haug, and F.-d. S. Systems, "Design sensitivity analysis of structural systems," *Mathematics in Science and Engineering*, vol. 177, no. C, pp. 1–82, 1986.

[46] D. Balagangadhar and S. Roy, "Design sensitivity analysis and optimization of steady fluid-thermal systems," *Computer Methods in Applied Mechanics and Engineering*, vol. 190, no. 42, pp. 5465–5479, 2001.

[47] J. C. Newman, A. C. Taylor, R. W. Barnwell, P. A. Newman, and G. J.-W. Hou, "Overview of Sensitivity Analysis and Shape Optimization for Complex Aerodynamic Configurations," *Journal of Aircraft*, vol. 36, no. 1, pp. 87–96, 1999. [Online]. Available: https://doi.org/10.2514/2.2416

[48] J. A. Snyman and D. N. Wilke, *Practical Mathematical Optimization*, 2nd ed. Springer, 2005.

[49] J. Laurenceau and P. Sagaut, "Building efficient response surfaces of aerodynamic functions with kriging and cokriging," *AIAA Journal*, vol. 46, no. 2, pp. 498–507, 2008.

[50] J. Laurenceau and 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.

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

[52] Y. Zhang, C. Gong, H. Fang, H. Su, C. Li, and A. D. Ronch, "An efficient space division–based width optimization method for rbf network using fuzzy clustering algorithms," *Structural and Multidisciplinary Optimization*, vol. 60, pp. 461–480, 2019.

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

[54] P. G. Constantine, E. Dow, and Q. Wang, "Active subspace methods in theory and practice: Applications to kriging surfaces," *SIAM Journal on Scientific Computing*, vol. 36, pp. A1500–A1524, 2014.

[55] N. Namura, K. Shimoyama, and S. Obayashi, “Kriging surrogate model with coordinate transformation based on likelihood and gradient,” *Journal of Global Optimization*, vol. 68, pp. 827–849, 8 2017.

[56] J. Li, J. Cai, and K. Qu, “Surrogate-based aerodynamic shape optimization with the active subspace method,” *Structural and Multidisciplinary Optimization*, vol. 59, pp. 403–419, 2 2019.

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

[58] Z. Li, J. Zhu, C. C. Foo, and C. H. Yap, “A robust dual-membrane dielectric elastomer actuator for large volume fluid pumping via snap-through,” *Applied Physics Letters*, vol. 111, no. 21, 2017. [Online]. Available: http://dx.doi.org/10.1063/1.5005982

[59] E. Liski, K. Nordhausen, H. Oja, and A. Ruiz-Gazen, “Averaging orthogonal projectors,” 2012.

[60] Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio, “Identifying and attacking the saddle point problem in high-dimensional non-convex optimization,” *Advances in Neural Information Processing Systems*, vol. 4, no. January, pp. 2933–2941, 2014.

[61] L. Nuñez, R. G. Regis, and K. Varela, “Accelerated random search for constrained global optimization assisted by radial basis function surrogates,” *Journal of Computational and Applied Mathematics*, vol. 340, pp. 276–295, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377042718300888

[62] L. Laurent, R. Le Riche, B. Soulier, and P.-A. Boucard, “An overview of gradient-enhanced metamodels with applications,” *Archives of Computational Methods in Engineering*, vol. 26, 07 2017.

[63] P. Bogaert, “Comparison of kriging techniques in a space-time context,” *Mathematical Geology*, vol. 28, no. 1, pp. 73–86, 1996. [Online]. Available: https://doi.org/10.1007/BF02273524

[64] F. L. Gao, Y. C. Bai, C. Lin, and I. Y. Kim, “A time-space Kriging-based sequential metamodeling approach for multi-objective crashworthiness optimization,” *Applied Mathematical Modelling*, vol. 69, pp. 378–404, 2019.

[65] J. Jang, J. M. Lee, S. G. Cho, S. Kim, J. M. Kim, J. P. Hong, and T. H. Lee, “Space-time kriging surrogate model to consider uncertainty of time interval of torque curve for electric power steering motor,” *IEEE Transactions on Magnetics*, vol. 54, no. 3, pp. 8–11, 2018.

[66] M. R. Kandroodi, B. N. Araabi, M. M. Bassiri, and M. N. Ahmadabadi, “Estimation of Depth and Length of Defects from Magnetic Flux Leakage Measurements: Verification with Simulations, Experiments, and Pigging Data,” *IEEE Transactions on Magnetics*, vol. 53, no. 3, 2017.

[67] D. Correia and D. N. Wilke, “Purposeful cross-validation: a novel cross-validation strategy for improved surrogate optimizability,” *Engineering Optimization*, vol. 53, no. 9, pp. 1558–1573, 2021. [Online]. Available: https://doi.org/10.1080/0305215X.2020.1807017

[68] D. A. White, “Multiscale topology optimization using neural network surrogate models,” *Computer Methods in Applied Mechanics and Engineering*, vol. 346, pp. 1118–1135, 2019.

[69] N. Navaneeth and S. Chakraborty, “Surrogate assisted active subspace and active subspace assisted surrogate—A new paradigm for high dimensional structural reliability analysis,” *Computer Methods in Applied Mechanics and Engineering*, vol. 389, p. 114374, 2022. [Online]. Available: https://doi.org/10.1016/j.cma.2021.114374

[70] J. M. Bouwer, D. N. Wilke, and S. Kok, “Spatio-temporal gradient enhanced surrogate modeling strategies,” *Mathematical and Computational Applications*, vol. 28, no. 2, 2023. [Online]. Available: https://www.mdpi.com/2297-8747/28/2/57

[71] B. Meyer, B. Harwood, and T. Drummond, “Nearest neighbour radial basis function solvers for deep neural networks,” 05 2017.