GAUSSIAN PROCESS PROXY MODELS FOR POROUS MEDIA FLOWS

by

Md Shamsuzzaman

Submitted in partial fulfilment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
January 2016

DEDICATION

To my parents and loving family.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Despite the recent data explosion, many optimization and decision making processes are constrained with detailed and/or expensive simulation runs. Therefore, original simulation models are replaced with a light-weight model that captures the underlying input-output relationship plausibly nonlinear and complex, known as *proxy*. The focus of this study is to develop a Gaussian Process based proxy model and apply it to porous media flows. Porous media flows are ubiquitous and, the study of groundwater virus transport and petroleum reservoir flows, are crucial for health and economic impacts. Experimental design plays an important role in extracting the most information from limited data. Optimization is always required for selecting the best design. On the other hand 'covariance function' or 'kernel' is the most important ingredient for constructing a Gaussian process model. In this study, all these elements are carefully considered and new methods or modifications to the existing method(s) are proposed. A new and efficient method for experimental design is proposed. Modification to an existing heuristic optimization technique is also implemented to improve its performance. An algorithm is proposed to produce composite kernels, which can select the optimal kernel given some base kernels and the initial design. Eventually, the proposed methods are applied to different case studies. Parameters of 1-dimensional groundwater virus transport problem and the porosity of a small-size industrial reservoir are estimated successfully using a limited number of simulation runs. In the optimal well placement problem, despite limited data, optimal well locations were calculated for two production wells of a real reservoir. The locations predicted using proxy models were found to be reasonably close to the original well sites of the reservoir, validating the proposed proxy models.

LIST OF ABBREVIATIONS USED

GP	Gaussian Process
DoE	Design of Experiment
AE	Audze-Eglais criterion
BHM	Borehole model
BLUP	Best linear unbiased predictor
CDF	Cumulative distribution function
DC	Distance correlation
EI	Expected improvement
LHD	Latin hypercube design
LMA	Levenberg-Marquardt algorithm
NOLHD	Nearly orthogonal Latin hypercube design
OLHD	Orthogonal Latin hypercube design
OK	Ordinary kriging
PDE	Partial differential equation
PDF	Probability density function
MLE	Maximum likelihood estimation
MM	Maximin
MC	Maximum absolute pairwise correlation
RMSE	Root mean square error
SA	Simulated annealing
GAMS	General Algebraic Modeling System
PSO	Particle swarm optimization
QPSO	Quantum-behaved particle swarm optimization
GA	Genetic algorithm

ACKNOWLEDGEMENTS

I would like to express my utmost gratitude and respect to my supervisor, Dr. Mysore G. Satish, for his guidance, advice and continuous encouragement to carry out this thesis. Without his kind support this thesis could not have been completed. The financial support provided by him through Natural Science and Engineering Research Council (NSERC) is highly appreciated.

I am also grateful to my supervisory committee members, Dr. Mo El-Hawary, Dr. Lei Liu and Dr. Dmitry Garagash, for their insightful suggestions and encouragements during the entire thesis work.

Thanks go to Arjun KC who helped in many ways and my colleagues in room D240 for their mutual support and encouragements.

I appreciate Schlumberger® for providing training in Petrel/RE and Ocean Plugin.

My special thanks go to my mother and siblings for their encouragement and emotional support. Last, but not the least, this work could not have been completed without the continuous support and love from my wife, Fahmida, my son, Sakif, and my daughter, Fariha. I dedicate this thesis to my wife and kids, and my mother in Bangladesh for their unconditional love and support.

CHAPTER 1 INTRODUCTION

1.1 BACKGROUND

With the explosion of availability of data, the use of 'Big Data' is becoming increasingly popular. One reason for the data explosion is that the users became data producers using the Internet. Moreover, cheaper sensors and advances in telemetry resulted in huge bursts of data in engineering, earth sciences, and medical/health sciences, etc., Ranganathan (2011); Grolinger et al. (2016); Zhou & Yang (2016); Clark et al. (2016). However, scarcity of data is still a niche for many scientific and engineering applications. For example, very few monitoring wells are generally available for groundwater flow studies, Kim (2015) and similarly, a limited number of wells are drilled for oil and gas reservoirs compared to the extent of a field due to economic reasons or other constraints. As drilling costs of oil and gas wells vary from 5.5 – 9.5 million dollars depending on location, Ziomek-Moroz (2012), a well is drilled only when it is economically viable. As a result, measured properties of an aquifer/reservoir, like porosity and permeability, are available for a very tiny fraction of the whole field. Virus transport in groundwater and petroleum reservoir flows however, are important fields of porous media flows due to their impacts on health and economy. On the other hand, a simulation is an indispensable tool for many engineering and scientific projects, including groundwater flow, Seethaa et al. (2015) and petroleum reservoirs, Zaydullin et al. (2014), where these properties are an integral part for the modeling. Missing data in the model are filled using inference and are validated by comparing the known production data and the simulated results. This process is known as parameter estimation in groundwater flow modeling while the same is known as 'history matching' in the reservoir engineering discipline. In any case, parameter estimation is an inverse process, Oliver et al. (2008); Xavier et al. (2013) and is hard to solve as different values of the

model parameters may be consistent with the available data (non-uniqueness) and it might require the exploration of a huge parameter space, potentially multimodal (finding a needle in a haystack). Usually a large number of simulation runs are carried out with known parameter values and the simulated results are compared with the known production/measured data, mostly corrupted with noise. The difference between the observed and simulated results is optimized (minimized) in terms of some distant metric. Unfortunately, optimization needs thousands of simulation runs. Moreover, the objective functions are multimodal in general and can be deceptive. Detailed porous media flow simulation is expensive in terms of time and computing resources. Therefore, the calculation of objective function for optimization using direct simulation becomes prohibitive for most cases. To circumvent the problem of using expensive simulation models, a model of the simulation model can be used instead Forrester et al. (2008). The resulting model developed using the data produced by the original simulation model is known as a surrogate or proxy model.

1.2 OBJECTIVE

The objective of this thesis is to develop an efficient proxy model for porous media flow simulation. The rationale for developing a surrogate is to replace an expensive simulation model with a cheaper model that can be used for further applications, like, optimization and decision making. There are different ways to develop a proxy model. For example, polynomial functions, radial basis functions, artificial neural networks (ANN), Gaussian Process (GP), and support vector machines (SVM) to name a few, Cranganu et al. (2015). In this thesis, the GP regression methods are used. The GP models have been extensively used for diverse fields of applications. It has fewer numbers of hyper-parameters to tune compared to ANN and inherent capacity to incorporate uncertainty with the measured parameters, a huge difference compared to many

popular alternatives. Although, it is considered as a linear model, the kernel can capture any underlying process, if selected properly. Therefore, in most cases, modeling nonlinearity is a matter of selecting a nonlinear kernel. However, selecting an appropriatekernel for the GP model is not a trivial issue. With the proper selection of a kernel and the judicious selection of experimental data points, an efficient Gaussian Process-based proxy model can be developed using a limited number of simulation runs. This can replace the use of an expensive simulator for further analysis and experimentation.

In the following section, the organization of the thesis is outlined.

1.3 ORGANIZATION OF THE THESIS

The organization of this thesis is shown in Figure 1.1. This chapter outlines the overall rationale, goal and objective of the thesis. This section in particular establishes the connections between different chapters. In Chapter 2, the design of experiment is discussed and a new method is presented with a nearly orthogonal space-filling design. As our objective is to create a proxy model from the data obtained by running a limited number of simulations, selection of proper parameter values for the simulation is a crucial factor. Selecting parameters efficiently is part of the research field known as 'Design of Experiment'. Similarly, parameters of the GP model, also known as hyper-parameters, are to be selected using some optimization algorithm. In most cases, these hyper-parameters are multimodal and it is difficult to find a global optimum. Therefore, in Chapter 3, an improved optimization heuristic algorithm is proposed based on existing quantum-behaved particle swarm optimization (QPSO). Krigging or GP models are introduced in Chapter 4 and the feasibility of using GP models as a proxy is carried out in the context of parameter estimation for a groundwater virus transport problem. The encouraging results from Chapter 4 led us to study GP models more deeply in Chapter 5. An algorithm was proposed to identify

proper kernel to capture the underlying process of the problem to be modeled using GP. Parameter estimation of a groundwater virus transport problem is carried out again with the new composite kernel. This approach is also applied for history matching of a reservoir problem. Further, the problem of optimal well placements in North Triumph gas reservoir was also solved using the proposed GP proxy model. In all cases, the results are promising. Detailed background information available in the literature for Design of Experiments and optimization using QPSO, is described in chapters 2 and 3 respectively, and in Chapters 4 and 5 for the GP Modeling. Conclusions from this study are presented in Chapter 6.

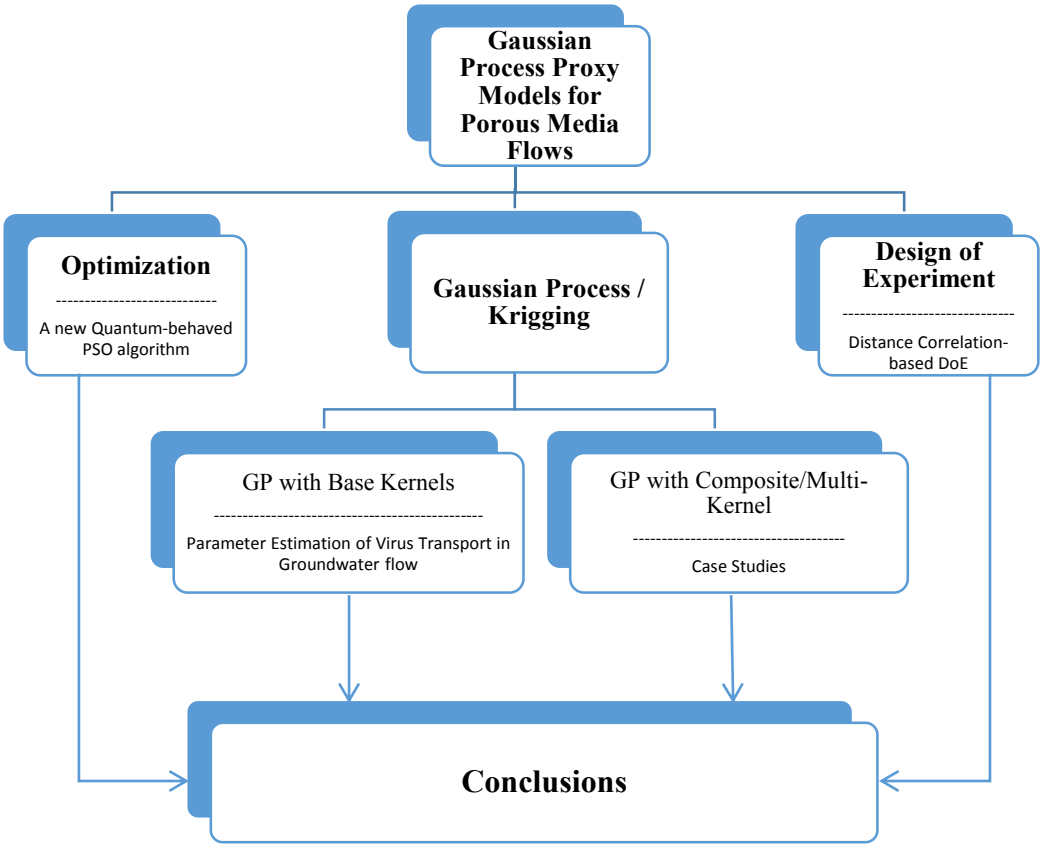

Figure 1.1: Organization of the Thesis

1.4 CONTRIBUTION TO NEW KNOWLEDGE

This study contributes finding a better proxy model for expensive simulators. Following outcomes contribute to improve GP-based proxy modeling.

- Distance correlation as a new measure for creating design of experiment (DoE). As distance correlation can unveil correlation between variables beyond linear relations, the proposed method can produce better DoE.

- Proposed perturbation and re-initialization as a remedy for one swarm-based algorithm and found it to be an effective tool.
- To our knowledge, Gaussian Process based proxy has not been used for parameter estimation in groundwater virus transport problem. Our work shows that it can be used as an effective tool.
- Combining different covariance functions can be a versatile tool for modeling different phenomenon effectively. An algorithm is developed to combine different basis kernels and is successfully applied to porous media flow problems.

CHAPTER 2 EFFICIENT DESIGN OF EXPEIRMENT BASED ON DISTANCE CORRELATION

In many real-world applications of optimization, the required model (function) evaluations are determined by expensive and time-consuming physical experiments or numerical procedures. Within this context, the objective of experimental design is to obtain meaningful information based on a strongly limited number of experiments or function evaluations. In order to generate informative designs, space-filling and orthogonality are widely considered to be essential. In this chapter, we review several design performance metrics that address these criteria, and the distance correlation-based metric is proposed for achieving improved experimental design. Three closely related randomized sampling schemes are proposed to generate nearly orthogonal designs with good space-filling properties in real time. The effectiveness of our approach is demonstrated by numerical examples and an illustrative borehole model case study.

2.1 INTRODUCTION

Experimental studies, whether based on actual physical experiments or on computer-assisted analysis, are frequently constrained by safety, money, resources and time. To give a few examples, it is evidently impossible to carry out full-scale experiments related to dangerous, hazardous or catastrophic events such as the failure of an industrial plant, nuclear reactor or spacecraft – even if, unfortunately, such accidents have occurred and do occur. Another, less dramatic example is the development of new engineering design alternatives that can be time-consuming and/or expensive to create and to test: hence, only a limited number of designs can be considered. Frequently, computer-based systems simulation tools are developed and used in such cases. Notice, however, that even sufficiently detailed simulation models may require substantial computational time. For example, a reservoir simulation model – used in the oil and gas industry to support decision making – may require hours to run, for each decision alternative considered.

In similar circumstances to the ones outlined above, a surrogate or proxy mathematical model is developed that approximately maps the input-output relationships of the underlying real-world processes.

The aim of the Design of Experiments (DoE) discipline is to develop efficient strategies to explore the design space on the basis of a strongly limited number of experiments. For discussions and case studies regarding DoE and/or resource-constrained optimization, consult Barthelemy and Haftka (1993), den Hertog and Stehouwer (2002), Simpson et al. (2004), Rikards and Auzins (2004), Biles et al. (2007), Horváth et al. (2007), Kleijnen (2009), Pintér and Horváth (2012), and references therein.

Specifically, we will consider here the problem of estimating a real-valued function $f(\boldsymbol{x})$: $\mathbb{R}^k \rightarrow \mathbb{R}$; here f is the outcome of some experiment that depends on the k-component real vector (input) $\boldsymbol{x}$. Function f will be assessed (approximated) on the basis of selected input variable settings called design points. We shall assume that n design points will be chosen simultaneously. Then the corresponding DoE is specified by a $n \times k$ size design matrix X. Each column of X represents the settings for a given component of x (called also a factor), and each row of X specifies an experiment defined by the complete set of input variable values chosen for evaluation.

Let us point out that advanced nonlinear – specifically, global – optimization methodology is often required to create good designs, as well as to handle the resulting proxy decision problems correctly since the unknown response surface of a single modeled function $f(x)$ may well be multimodal. Various criteria for selection of a good design is discussed in section 2.2. Without going into technical details at this point regarding the subject of global optimization, we only

mention a few topical books by Horst and Pardalos (1995), Pintér (1996), and Pardalos and Romeijn (2002).

It is to be noted that DoE methods targeted for use in computer experiments differ significantly from the classical DoE methods as the output from the former method is deterministic while the later method assumes the presence of random error in the output. Therefore, classical DoE methods seek to minimize the variance of the parameters and generally place the design points at the extremes of the experimental region. On the other hand, DoE for computer experiments are mainly concerned with reducing the bias in the estimated model or metamodel. Moreover, concepts of blocking and replication, extensively used in classical DoE, are irrelevant for the DoE for computer experiments. Some of the efficient classical DoE include 2^k Factorial Design, Fractional Factorial Design (Mukerjee & Wu, 2006), Central Composite Design (Demirel & Kayan, 2012), Response Surface Methodology (Kleijnen J. P., 2015), Taguchi method (Wolf, Henes, Bogdanski, Lutz, & Krämer, 2013), etc. Modern DoE includes but not limited to Monte Carlo Sampling (Kroese & Chan, 2013), Orthogonal Arrays (Ma, Fang, & Liski, 2000), Latin Hypercube Design etc.

In this study, we will specifically deal with Latin hypercube designs (LHDs). In the two-dimensional unit interval $[0,1]^2$ an $n \times n$ grid of equally sized square cells with n chosen sample point positions is a Latin square design if there is exactly one sample point in each row and in each column. The sample points can be simply chosen as the centers of the selected grid cells. LHD is the direct generalization of this sampling concept to arbitrary dimension k. Following the notation used by Husslage et al. (2011), a k-dimensional LHD is a set of n design points $x_i = (x_{i1}, x_{i2}, \cdots, x_{ik}) \in \{0, \cdots, n-1\}^k$ such that all x_{ij} are distinct for each dimension $j = 1, \ldots, k$. Originally proposed by McKay et al. (1979), LHD based sampling plans are considered to be one

of the most efficient DoE strategies – when used properly. The principal virtue of LHDs is to offer a flexible sampling plan which, by construction, is non-overlapping, since a LHD in each dimension $j = 1, \ldots, k$ gives exactly n distinct sample values. All LHDs will not be automatically suitable designs, however. To give an extreme example, if we take a completely diagonal design (in which all k components are equidistantly sampled along the main diagonal of the design space) then this will give a very poor sample for two reasons. First of all, the design factors are perfectly correlated; therefore we will not be able to distinguish among the effects of the design factors. Second, a large portion of the experimental region will not be explored at all, especially when we are "far" from the main diagonal. Hence, a proxy model based on such a design cannot be expected to be acceptable (i.e., to do well) over the entire design space. Many other LHD instances could be inferior choices for similar reasons.

To eliminate evidently poor LHDs, various performance criteria have been introduced to reward sampling plans which give good coverage of the design space and also offer low correlation among the design points selected. These criteria will be expressed respectively by the space-filling and orthogonality properties of a given LHD. Designs that spread points evenly throughout the experimental region is known as space-filling designs while orthogonal designs offer uncorrelated input values that help to independently assess effects of individual inputs on the response. Both properties are widely considered to be key characteristics of good designs. Unfortunately, good space-filling properties do not necessarily imply good orthogonality and vice versa. For example, two instances of 5×5 LHDs are shown in Figure 2.1. By simple inspection, it can be conjectured that the first design seems inferior to the second, in terms of the orthogonality criterion. However, an orthogonal design can provide a stepping stone towards space-filling designs, and space-filling designs can be used to find nearly orthogonal designs.

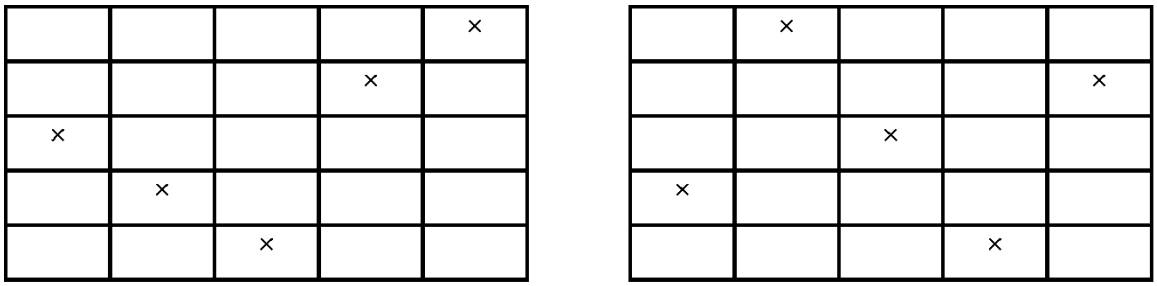

Figure 2.1: Two LHD examples

To find high-quality designs – or the "best possible" LHD – is far from trivial. For a given pair of k and n, the set of all possible LHDs is finite, of course. However, the cardinality of the entire LHD design option set quickly becomes astronomical with increasing k and n, since – for (by assumption) interchangeable input parameters – this set contains $\frac{n!^{k-1}}{(k-1)!}$ essentially different designs: consult Pronzato and Müller (2011). To illustrate this point, if we take just $n = 20$ and $k = 2$ (a rather small DoE instance) then we should consider $20! = 2{,}432{,}902{,}008{,}176{,}640{,}000 > 2.43 \times 10^{18}$ LHD choices. Therefore, instead of finding the best possible design based on a suitable performance criterion (or criteria), in many cases an optimized design is used that is expected and sometimes proven to perform well. Due to the outlined difficulties associated with finding the best possible LHD, nearly orthogonal designs (NOLHDs) can be used with optimized space-filling and orthogonality properties.

The construction of space-filling LHDs has been discussed extensively in van Dam et al. (2007), Grosso et al. (2009), van Dam et al. (2009), Pintér and Horváth (2012). Ankenman et al. (2010) analyze the issue of correlation among design points in the context of simulation metamodeling. Orthogonal Latin hypercube designs (OLHDs) are discussed in Ye (1998), Steinberg and Lin (2006), Cioppa and Lucas (2007), Nguyen (2008), Joseph and Hung (2008).

Following the introduction, this chapter is organized as follows: Several relevant performance metrics are reviewed in Section 2.2 including the proposed metric. In Section 2.3, three algorithms are introduced to find good designs. Section 2.4 describes our numerical experiments and the analysis of the results obtained. An illustrative case study using Borehole Model follows Section 2.5.

2.2 DoE Performance Metrics

A number of apparently non-equivalent design performance criteria have been proposed in the DoE literature. Three of the most frequently used metrics are reviewed in Sections 2.2.1 to 2.2.3, followed by a proposed new criterion in Section 2.2.4. The criteria discussed in Sections 2.2.1 and 2.2.2 are targeted for space-filling design while the same in Sections 2.2.3 and 2.2.4 are focused on design orthogonality.

2.2.1 The Maximin (MM) Design Criterion

The Maximin (MM) criterion is used frequently, to assess the space-filling quality of designs. For a given k and sample size n, the MM design is attained when the smallest separation distance between all pairs of design points $min_{i\neq j}d(x_i, x_j)$ is maximal among all possible designs *X*. Formally, we define the Euclidean distance between columns i and $j, i < j$ of the design matrix as shown below:

$$d_{ij} = d(x_i, x_j) = \sqrt{\sum_{l=1}^{k}(x_{il} - x_{jl})^2} = \sqrt{(x_{i1} - x_{j1})^2 + \cdots + (x_{ik} - x_{jk})^2} \quad (2.1)$$

Where x_{il} and x_{jl} are the positions of the i-th and j-th design points of l-th factor.

The MM design maximizes min d_{ij} for all possible design matrices. The fact that in an MM based design the distance between all possible pairs of design points is bounded from below by the value $d = \max \min d_{ij}$ indicates a good space-filling property.

Let us note here that analogously to (2.1) the distance between two points can also be based on distance concepts other than Euclidean. As an example, the rectangular distance $d_R(s,u)$ of two points s and u of the design matrix X is defined using the L_1-norm:

$$d_R(s,u) = \sum_{i=1}^{n} |s_i - u_i| \tag{2.2}$$

Alternatively, the Euclidean distance of s and u is defined using the L_2-norm, leading to (2.1):

$$d_E(s,u) = \sqrt{\sum_{i=1}^{n} (s_i - u_i)^2} \tag{2.3}$$

For a given design and a selected (rectangular or Euclidean) distance, we can define a distance list $D = (D_1, \cdots, D_k)$, in which the list elements are the distinct values of pair-wise distances, sorted from the smallest to the largest. The value of the index can be as large as $n(n-1)/2$. Let J_i be the number of pairs of runs in the design that have distance D_i. Then, a design X is called a MM design if it sequentially maximizes the values D_i and minimizes the values J_i in the following order:$(D_1, J_1, D_2, J_2, \cdots, D_k, J_k)$. A related scalar-valued function proposed by Morris and Mitchell (1995) to rank competing designs is given below, for a given positive integer *p*.

$$\phi_p = \left[\sum_{i=1}^{k} J_i {d_i}^{-p}\right]^{\frac{1}{p}} \tag{2.4}$$

The criterion function ϕ_p is easier to compute as it does not require the ordering of inter-site distances. Hence, it is widely used in literature for searching optimized LHDs as indicated by Morris and Mitchell (1995), Jin et al. (2002), Grosso et al. (2009), Viana et al. (2010), Moon et al. (2011).

2.2.2 The Audze-Eglais (AE) Criterion

Audze and Eglais (1997) proposed a criterion function which is based on the analogy of minimizing the total pair-wise Coulomb potential force considering all design points as charged particles. The AE criterion based optimal design points are hence obtained by minimizing the following criterion function over a feasible set such as the unit hypercube:

$$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{1}{d(x_i,x_j)^2} \tag{2.5}$$

Where, $d(x_i,x_j)$ is the distance between i and j design points and can be calculated using Equation 2.1. Search for optimized designs using the AE criterion has been conducted by many researchers, such as Bates et al. (2003), and Fuerle and Sienz (2011).

To conclude our brief discussion of space-filling criteria, let us point out that finding MM or AE based optimized designs leads to increasingly difficult global optimization problems as the number of points grows, since the model instances based on these criterion are highly multimodal.

2.2.3 Maximum Absolute Pairwise Correlation (MC) Criterion

The maximum absolute pairwise correlation measure (ρ_{map}) has been proposed by Cioppa and Lucas (2007) to distinguish between designs by measuring their orthogonality. There are $\binom{k}{2}$ pairwise correlation coefficients between any two columns in a design matrix X. Using Pearson's equation, the correlation between X_i and X_j is given by

$$\rho_{ij}=\frac{\sum_{b=1}^{n}\left[\left(x_b^i-\bar{x}^i\right)\left(x_b^j-\bar{x}^j\right)\right]}{\sqrt{\sum_{b=1}^{n}\left(x_b^i-\bar{x}^i\right)\sum_{b=1}^{n}\left(x_b^j-\bar{x}^j\right)}} \tag{2.6}$$

Here $\bar{x}^i$ and $\bar{x}^j$ are the mean values of the i^{th} and j^{th} columns respectively and x_b^i is the b^{th} value of the i^{th} column.

The largest absolute correlation among the columns gives the most extreme pairwise correlation. This value provides the degree of orthogonality of a design. By minimizing the worst-case pairwise correlation, all pair-wise correlations are indirectly controlled. Therefore the chosen control parameter regarding the orthogonality of OLHD or NOLHD designs is to minimize

$$\rho_{map} = \max_{i \neq j}\{|\rho_{ij}|\} \tag{2.7}$$

Following Harnandez (2008), a design is somewhat arbitrarily, but plausibly defined as *nearly orthogonal* if $\rho_{map} \leq 0.05$.

2.2.4 Maximum Distance Correlation (DC) Criterion

Distance correlation (DC) is a measure of statistical dependence between two random variables or two random vectors of arbitrary dimensions. It was proposed by Székely et al. (2007) and Székely and Rizzo (2009), as a measure of independence. The key property of DC is that its value is 0, if and only if the two random variables are statistically independent. Distance covariance is a natural extension of product-moment covariance and proved to give better results. Székely and Rizzo (2009) presented six examples and compared results in terms of distance correlation and conventional Pearson correlation. In their study distance correlation always provided better insight regarding the data used. Similarly, using distance correlation a nonlinear association between the random variables can also be identified. Therefore we propose distance correlation as a new metric to identify good designs. In order to obtain a good design in terms of orthogonality, our objective is to minimize the maximum distance correlation between a particular column to the rest of the columns in a design matrix.

The distance correlation $R(x, y)$ between two random vectors x and y with finite first moments is defined by

$$R^2(x,y) = \begin{cases} \frac{\nu^2(x,y)}{\sqrt{\nu^2(x)\nu^2(y)}}, & \text{if } \nu^2(x)\nu^2(y) > 0 \\ 0, & \text{if } \nu^2(x)\nu^2(y) = 0 \end{cases} \tag{2.8}$$

Here ν is the distance covariance of the random vectors and the calculation method is outlined below.

For an observed random sample $(x,y) = \{(x_k, y_k): k = 1, \cdots, n\}$ from the joint distribution of random vectors x in $\mathbb{R}^p$ and y in $\mathbb{R}^q$, define

$$a_{kl} = |x_k - x_l|_p \tag{2.9a}$$

$$\bar{a}_{k.} = \frac{1}{n}\sum_{l=1}^{n} a_{kl} \tag{2.9b}$$

$$\bar{a}_{.l} = \frac{1}{n}\sum_{k=1}^{n} a_{kl} \tag{2.9c}$$

$$\bar{a}_{..} = \frac{1}{n^2}\sum_{k,l=1}^{n} a_{kl} \tag{2.9d}$$

$$A_{kl} = a_{kl} - \bar{a}_{k.} - \bar{a}_{.l} + \bar{a}_{..} \tag{2.10}$$

B_{kl} is defined similarly. Then the empirical distance covariance is defined by

$$V_n^2(x,y) = \frac{1}{n^2}\sum_{k,l=1}^{n} A_{kl}\, B_{kl} \tag{2.11}$$

Finally, the empirical distance correlation is defined as

$$R_n^2(x,y) = \begin{cases} \frac{{V_n}^2(x,y)}{\sqrt{{V_n}^2(x){V_n}^2(y)}}, & {V_n}^2(x){V_n}^2(y) > 0 \\ 0, & {V_n}^2(x){V_n}^2(y) = 0 \end{cases} \tag{2.12}$$

The test statistic $nR_n^2\big(rank(x), rank(y)\big)$ can be used to test whether the variables are independent or not. In Appendix B of Székely and Rizzo (2009), critical values of test statistics are provided. As an example, for a sample size of 100, the critical values are 4.24 and 5.26, for the significance levels 5% and 10%, respectively. Therefore, if the calculated statistic is greater than these values, then the hypothesis regarding their independence can be rejected with 95% and 90% confidence level, respectively.

To conclude this section, let us remark that in addition to the performance criteria reviewed here, other indicators are also used. For topical discussions consult Franco et al. (2009), Jourdan and Franco (2010), Pronzato and Muller (2012).

2.3 Fast Algorithms to generate Good Designs

In order to obtain a high-quality space-filling and/or nearly orthogonal design, two different approaches have been followed. Researchers have been using transformation algorithms or optimization methods to find designs based on the above discussed performance metrics or their suitable combinations.

Optimized LHDs have been intensively studied in recent years, partly based on applying heuristic methods (simulated annealing, evolutionary optimization, iterated local search and tabu search) or exact (combinatorial, continuous global and mixed integer-continuous) optimization approaches. For details, consult Morris and Mitchell (1995), Jin et al. (2005), Husslage et al. (2011), Liefvendahl and Stocki (2006), Grosso et al. (2009), Jourdan and Franco (2010), Chen et al. (2012) and the references therein.

Here we will use a different (not optimization based) approach based on orthogonalization. The methods of Owen adopted from Owen (1994) and Tang (1998), and Florian (1992) are described below since these will be used in the present study.

2.3.1 Owen's Algorithm

Let $rank(x)$ denote the ranks of the components of a vector $\boldsymbol{x}$. The notation $takeout(y, x)$ denotes the vector of residuals from the linear regression model $y = \beta_0 + \beta_1 x + \epsilon$. Then for a given initial Latin hypercube $L = (l_1, \ldots, l_m)$, Owen's algorithm proceeds by alternating between the forward and backward steps in the following.

- *Forward step*: for $j = 1, \ldots, m-1,$ and for $k = j+1, \ldots, m,$ set $l_k = rank(takeout(l_k, l_j)$
- *Backward step*: for $j = m, \ldots, m-1,$ and for $k = j-1, \ldots, 1,$ set $l_k = rank(takeout(l_k, l_j)$

The central idea is that for a fixed j, k we want to update $l_k = rank(r_k)$ such that the new l_k has a small correlation with l_j . It is clear that the correlation between l_k and l_j should be small, as r_k is based on the error terms of regression.

2.3.2 Florian's Algorithm

Let us consider that the design matrix $R(n \times k)$ is based on the rank of the factors in each column. The correlation matrix $T(k \times k)$ based on the correlation of a pair of columns is created with entries

$$T_{ij} = 1 - \frac{6 \sum_{l=1}^{n} \left(R_l^i - R_l^j\right)^2}{n(n^2-1)} \tag{2.13}$$

Here R_l^i is the rank of the lth element in the ith column. Since T is a symmetric positive definite matrix, it can be factorized using Cholesky's factorization method.

$$T = QQ' \tag{2.14}$$

The inverse of the lower left triangular matrix is obtained by Equation 2.14 if $S = Q^{-1}$. Florian's algorithm states that new design R_{new} will have better orthogonality if the current design is multiplied by S' as shown below.

$$R_{new} = RS'$$

2.4 NUMERICAL EXAMPLES AND DISCUSSION

As noted earlier, using an optimization algorithm to find the best design can also become a daunting task when the number of experiments (also called runs) n and factors k are increased even moderately. Therefore it is rare to find computationally tractable complete numerical examples with more than 100 runs and 10 factors.

To test the efficiency of the proposed index *maximum distance correlation*, a 25×4 design was obtained using a simple randomized sample based algorithm summarized below.

ALGORITHM 2.1: Ad hoc algorithm to compare different performance criteria

---

1. Create an initial randomly generated LHD X of given dimension and apply Owen's algorithm. Calculate the performance index of the design, then copy design X to design Y.
2. Randomize Y by swapping two random elements in each column to generate a new design. Calculate the performance index of Y as well.
3. Compare the performance of the two designs and assign the better design to X and the other to Y.
4. Repeat steps 2 and 3, until a given termination condition is reached (see below).

Initially, four performance criteria (indices) are studied, namely: maximin criterion (MM), Audze-Eglais criterion (AE), maximum absolute pairwise correlation (MC) and maximum distance correlation (DC). Recall that AE and MM are aimed at good space-filling, while MC and DC are aimed at design orthogonality. In all cases, one of these four objectives is used to find an optimized design. As a stopping criterion in the ad hoc algorithm, the total number of

perturbations was selected as $n \times (n-1) \times k \times 10$; here n and k are the number of runs and number of factors, respectively. In each experiment, the performance of the best design found is calculated in terms of all 4 metrics. This process is repeated 50 times, and for each metric the resulting statistics of these experiments are presented in Tables 2.1 to 2.4. In our numerical experiments, we used a personal computer which has an Intel i7 core, 2.8 GHz speed, 4 GB RAM and it runs under the Windows 7 Enterprise 64-bit OS.

From the results it can be seen that, in general, orthogonality might not be properly preserved when space-filling criteria are considered, and *vice versa*. Although in our study the use of AE produces better results than MM, the result fails to meet the nearly orthogonal design criterion $\rho_{map} \leq 0.05$ suggested by Hernandez (2008). Notice that the resulting performance does not differ very much using either of these criteria, while AE is easier to calculate than MM. A better design is obtained when DC is used in comparison to MC, while DC is computationally more expensive than MC.

Based on the above discussion and our results, we state that space-filling and orthogonality criteria should be simultaneously used to find better designs. This can be done in a proper multi-objective optimization framework by placing a constraint on the expected level of selected quality criterion/criteria, and optimizing the value of the chosen primary criterion. Such optimization problems can be highly nonlinear, in line with our earlier comment regarding the necessity of using advanced (global) optimization techniques.

We also tried simulated annealing (SA) in order to find optimized designs. However, SA took a long runtime compared to the ad-hoc randomization module outlined above. For example, for a 25×4 design, SA took more than 13 minutes, whereas the ad-hoc method proposed took less than 15 seconds to generate the same quality design. (It should be noted that the codes for both

SA and the ad-hoc algorithm are written by the author and that these are not optimized for performance.) Notwithstanding this note, we can state that for larger dimensions SA is expected to take such a huge amount of time that it might not be acceptable in many applications. Moreover, in our tests SA did not always produce better design for all runs. For many computer experiments, pseudo-real-time DoE is required, which is not always possible to attain using optimization alone. This observation remains valid with respect to many other easy-to-implement heuristic optimization approaches to this problem; at the same time, state-of-the-art global optimization methods and software can help to find highly optimized designs.

Table 2.1: Statistics of performance for the experiment where the AE criterion is the objective

Metric	Minimum	Mean	Maximum	Standard deviation
AE	1.032889	1.120621	1.577317	0.078413
DC	0.231187	0.337589	0.755229	0.088784
MM	2.645751	5.163882	7.141428	1.074393
MC	0.055384	0.237708	0.463077	0.100892

Table 2.2: Statistics of performance for the experiment where the MM criterion is the objective

Metric	Minimum	Mean	Maximum	Standard deviation
AE	1.014567	1.243763	7.100276	0.838989
DC	0.211223	0.381275	0.810663	0.095766
MM	2.0	5.645614	7.810250	0.983381
MC	0.006154	0.2914	1.0	0.165409

Table 2.3: Statistics of performance for the experiment where the DC criterion is the objective

Metric	Minimum	Mean	Maximum	Standard deviation
AE	1.042522	1.154269	1.510888	0.085081
DC	0.223996	0.305991	0.494722	0.057231
MM	2.0	4.572216	6.244998	0.997416
MC	0.001538	0.176508	0.379231	0.084378

Table 2.4: Statistics of performance for the experiment where the MC criterion is the objective

Metric	Minimum	Mean	Maximum	Standard deviation
AE	1.061892	1.163774	1.325872	0.065611
DC	0.247085	0.373979	0.588166	0.084802
MM	2.645751	4.869554	6.782330	0.887383
MC	0.033846	0.167246	0.452308	0.098185

Based on the experimental results presented and discussed above, our key observations are summarized below.

- There is no established pattern between the change in points of a design and the performance indices. However, one can expect that shifting a point to neighboring grid point does not change design uniformity drastically.
- Optimizing a design based on orthogonality does not guarantee good space-filling, and *vice versa*.
- Starting with a good design evidently ensures that a design of at least such quality (if not better) will be returned by the randomized sampling algorithms, using the same design criterion, of course. For example, if the initial design is based on Owen's, Tang's, and one of the orthogonality criteria is used for optimization, then a better result is rightly expected in terms of orthogonality. However, if the space-filling criterion is used in the algorithm, then the resulting design might not preserve as good orthogonality as the initial design.
- For a good quality space-filling criterion based design, one can start with a Sobol or other quasi-random sequence of designs, or apply the computationally efficient design generator method described in Pintér and Horváth (2012).

Based on the above observations, the following more efficient algorithm is proposed to generate nearly orthogonal space-filling designs.

ALGORITHM 2.2: Algorithm based on Florian's method + randomization + swapping schemes

1. *Initialization*: Create 50 random LHDs and select the best one based on performance index of choice. Say it is X.
2. *Florian's Step*: $Y = Florian(X)$
3. Compare the performance of the two designs:
    a. If Y is better than X, exchange X and Y. GOTO step 2
    b. Increment counter *notUpdated*
4. *Randomization Step*: If *notUpdated* > 5
    a. $Randomize(Y)$. Calculate performance.
        i. If Y is better than X, exchange X and Y.
        ii. *notUpdated* = 0. GOTO step 2
    b. *Perturb*(Y). Calculate performance of Y.
        i. If Y is better than X, exchange X and Y.
        ii. *notUpdated* = 0. GOTO step 2
5. Repeat step 2 until terminating condition is reached.

Algorithm termination can be based on a prefixed total number of sampling experiments. A simple statistical basis for choosing the number of experiments is given below and a termination criterion is fixed accordingly for the proposed algorithm. Note that we assume that Y is generated uniformly through randomization in steps 1 and 4(a) over the entire design space. After initialization, it is crucial to limit the number of iterations to find a better design. As noted earlier, the number of possible designs increases rapidly with the increase of n (number of sample points) and/or k (dimension of the problem). Reducing the time to find a good design without missing all potentially good designs is an essential point to consider. Let us assume that high quality designs are a (perhaps rather small) fraction $0 < \epsilon < 1$ of all possible designs. We also assume that a randomly chosen good design is required with $1 - \delta$ reliability. Here $0 < \delta <$

1 is another small parameter. Then for given ϵ and δ, the sample size n should be based on the criterion $(1-\epsilon)^n \leq \delta$. This inequality is satisfied, if $n\, ln(1-\epsilon) \geq ln\, \delta$. As an example, for $\epsilon = 0.005$ and $\delta = 0.005$, this relation results in the requirement $n \geq 1058$. Consequently, $n = 1058$ was chosen and used as termination criterion.

Different swapping schemes are used in the proposed algorithm. For random swapping, for each column, two random elements are swapped. On the other hand, the *Perturb method* in step 4(b) of the algorithm is targeted to push the random design towards the best design found so far. To do so, first an element is selected at random index in a column at the current design, then the index of the same value is searched in the same column at the best design. The values at these two indices are swapped and thereby the new design becomes more similar to the best design.

For comparison, two designs published in literature are compared with the designs generated using our algorithm in terms of different performance indices. First we selected a 14×12 design presented by Hernandez (2008) shown in Table 2.5. This author used maximum absolute pairwise correlation as performance index and Florian's method. We calculated different performance indices of the Hernandez design shown below.

In Table 2.6 the performance of this design is compared with the design obtained by our algorithm, using different design criteria.

Table 2.5: A 14×12 design from Hernandez (2008)

3	6	1	6	14	5	5	6	10	6	14	6
1	7	9	10	5	14	2	8	8	14	4	4
12	8	3	14	4	7	13	2	11	13	8	11
9	5	4	3	3	2	3	12	6	9	2	14
10	1	7	1	10	8	14	9	7	12	6	2
11	12	10	13	12	1	6	11	9	7	3	1
6	3	14	11	11	11	12	10	4	4	7	13
4	11	11	7	7	6	9	14	12	11	13	12
2	4	5	12	9	4	7	4	3	5	5	8
14	2	12	8	6	9	1	5	13	3	11	7
7	10	13	4	2	3	8	1	2	8	12	5
5	13	8	2	8	10	11	3	14	2	1	9
8	9	2	9	1	12	10	13	5	1	10	3
13	14	6	5	13	13	4	7	1	10	9	10

Table 2.6 : Comparison of 14×12 designs in terms of different performance indices

Performance Metric of DoE	Reference DoE from Hernandez (2008)	Obtained DoE using proposed algorithm with different performance criteria		
		MC	DC	AE
DC	0.3588	0.3676	0.3633	0.3681
AE	0.2205	0.2192	0.2203	0.2188
MC	0.0461	0.0813	0.0725	0.0769

From the results we can see that, in terms of orthogonality, optimization based on DC performs better compared to the results obtained based on MC in terms of orthogonality. The AE based approach gives the best space-filling design. In comparison to the reference design, the proposed algorithm gives better space-filling DoE in all cases. Although the orthogonality obtained is somewhat inferior and it does not conform with Hernandez's limit (0.05), it is still less than the critical value for orthogonality based on distance correlation. Namely, the critical value of distance correlation with 14 design points is 4.25 (or 5.16) with 90% (or 95%) confidence respectively, while the calculated value of nR_n^2 for the highest DC (using AE) is 1.897. It should also be mentioned that the algorithm based on the criterion DC took around 4 seconds, using the

MC criterion took around 2 seconds, and using the AE criterion took less than a second for this particular design.

Table 2.7: Comparison of 25×24 design in terms of different performance indices

Performance Metric of DoE	Reference DoE from Hernandez (2008)	DC index (24.96 s)	AE index (1.37 s)	MC index (6.7 s)
DC	0.3147	0.3170	0.3191	0.3209
AE	0.1155	0.1155	0.1155	0.1156
MC	0.0477	0.0631	0.0669	0.0885

Another design of dimension 25×24 from Hernandez (2008) has been considered for comparison. The results are shown in Table 2.7. Similar observations to the ones stated above apply for this case as well.

2.5 An Illustrative Case Study: Fitting a Borehole Model

In this section, we discuss a borehole model (BHM) which will be used to illustrate our design selection approach. The BHM considered here serves to describe the flow rate of water from an upper aquifer to a lower aquifer through a borehole. By assumption, the two aquifers are separated by an impermeable rock layer and the borehole is drilled from the ground surface (see Figure 2.2).

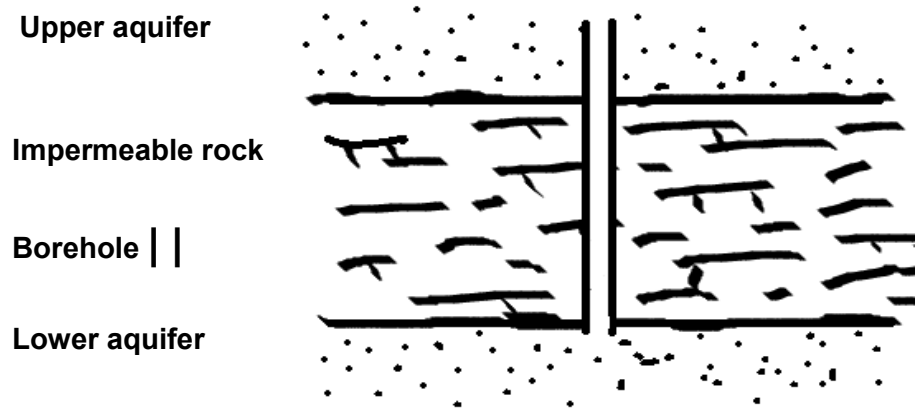

Figure 2.2: Borehole Model

The BHM has been studied by many researchers in the context of engineering applications and experimental designs, as in Worley (1987), Chin (1991), Morris and Mitchell (1995), Ahn and Owen (2001), Li (2002), Fang and Lin (2003), Fang et al. (2006).

The BHM determines the flow rate measured in cubic meters per year (m3/yr.) through the borehole drilled from the ground surface through the two aquifers. The analytical BHM formulation used here is derived from Bernoulli's law in fluid mechanics, as discussed in Kundu and Cohen (2008), under the following assumptions: steady-state flow from the upper aquifer into the borehole and from there into the lower aquifer, laminar and isothermal flow through the borehole, and the absence of groundwater gradient.

We shall use the following notation, with the corresponding units shown in brackets. We also provide ranges for the experimental domain considered for each parameter. This information will be used in the sampling based verification of the calculated surrogate model.

$r_w(m) \in [0.05, 0.15]$ radius of the borehole

$r\ (m) \in [100, 50000]$ radius of influence

$T_u(\frac{m^2}{yr}) \in [63070, 115600]$ transmissivity of the upper aquifer

$T_l(\frac{m^2}{yr}) \in [63.1, 116]$ transmissivity of the lower aquifer

$H_u(m) \in [990, 1110]$ potentiometric head of the upper aquifer

$H_l(m) \in [700, 820]$ potentiometric head of the lower aquifer

$L(m) \in [1120, 1680]$ length of the borehole

$K(\frac{m}{yr}) \in [9855, 12045]$ hydraulic conductivity of the borehole

If the entire input information vector is expressed as, $\boldsymbol{x} = (r_w, r, T_u, T_l, H_u, H_l, L, K_w)$ then the flow rate is calculated using the following equation (based on Bernoulli's law):

$$y = y(\boldsymbol{x}) = \frac{2\pi T_u (H_u - H_l)}{\ln(\frac{r}{r_w})\left(1 + \frac{2LT_u}{\ln(\frac{r}{r_w}) r_w^2 K_w} + \frac{T_u}{T_l}\right)} \tag{2.15}$$

Let us note that, in accordance with Equation 2.15, we will use $\ln(r)$ and $\ln(r_w)$, instead of r and r_w when searching for an appropriate model form.

Next, we discuss the issue of surrogate model selection to approximate the flow rate for this case. We will postulate a $1 + 8 + 8 = 17$-term simplified quadratic polynomial response surface surrogate model, without considering interaction terms. In other words, the response calculated from the surrogate model is approximated as follows:

$$y(x) \sim \hat{g}(x, \beta) = \beta_0 + \sum_{k=1}^{8} \beta_k x_k + \sum_{l=1}^{8} \beta_l x_l^2 \tag{2.16}$$

Denote by $\boldsymbol{x_i}, i = 1, \dots, n$ the input points used the design dataset; $\boldsymbol{y_i}\ i = 1, \dots, n$ the output value of the "true" model (2.15) at point $\boldsymbol{x_i}$, and $\widehat{\boldsymbol{g}}_{\boldsymbol{i}} = \boldsymbol{g}(\boldsymbol{x_i})$, $i = 1, \dots, n$ is the output value received using the quadratic model in (2.16). Our goal is to select the parameter vector $\boldsymbol{\beta}$ to obtain a close approximation of the model (2.15) by (2.16) over the experimental domain.

The quality of the surrogate model will be measured in terms of the square root of the mean square error (RMSE) shown below.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{g}_i)^2}{n}} \tag{2.17}$$

There are 17 parameters in the quadratic proxy model. These parameters will be fitted applying the widely used Levenberg-Marquardt algorithm (LMA). It is well known that LMA provides a robust numerical solution approach to locally minimizing an unconstrained (nonlinear) function. Hence, it is often used in least squares curve fitting and in other areas of nonlinear programming as described in Kelley (1999).

For comparison purposes, we will cite and use an existing design, in addition to our design in order to create a proxy. The existing design is taken from Deng-et al. (2012). Our design points are created using algorithm 2 introduced based on the maximum distance correlation criterion. In total, 30 sample points were used for two reasons, namely, i) to be able to compare our results with the reference design and ii) as the proxies are created to minimize the number of simulation runs, a limited number of evaluations of the actual function/simulation is desired. However, in general, the better the calibration of the model is expected, the more points are desirable.

In order to evaluate the quality of the approximation generated, 10000 random input points have been generated within the given experimental ranges and then we calculated the resulting RMSE

values. Using the reference design in Deng et al. (2012) and solving for the coefficient vector **β** using the LMA, the following polynomial response surface model has been obtained.

$$\hat{g}(\boldsymbol{x},\beta) = -40592.70 + 184.11\ln(r_w) + 39.60\ln(r_w)^2 - 38.28\ln(r) + 2.03\ln(r)^2 + 0.002T_u - 1.16\times10^{-8}T_u^2 + 0.802T_l - 0.005T_l^2 + 8.32H_u - 0.004H_u^2 + 4.42H_l - 0.003H_l^2 + 0.12L - 6.03\times10^{-5}L^2 + 6.30K_w - 0.0003K_w^2$$

The RMSE value calculated for this response surface model equals approximately 2.37.

Applying our proposed design selection method, we obtain the model shown below.

$$\hat{g}(\boldsymbol{x},\beta) = -15.46 - 11.32\ln(r_w) - 1.52\ln(r_w)^2 - 12.26\ln(r) + 0.89\ln(r)^2 - 0.00016T_u - 8.24\times10^{-10}T_u^2 + 0.165T_l - 0.0014T_l^2 + 0.244H_u - 0.0002H_u^2 - 0.082H_l + 0.00013H_l^2 + 0.0297L - 2.26\times10^{-5}L^2 + 0.0154K_w - 1.82\times10^{-6}K_w^2$$

The RMSE value calculated for this model equals 2.06; this represents an approximately 15% improvement over the design used here for comparison.

Let us remark finally that in a real-world setting further modelling steps would be warranted such as model interpretation, sensitivity analysis and if appropriate an importance ranking of the input variables with respect to the output.

2.6 CONCLUSIONS

In many real-world applications of optimization, the required model function evaluations are determined by expensive procedures. The objective of experimental design is to obtain meaningful information based on a limited number of function evaluations. In this study, we review several design performance metrics that address space-filling and orthogonality, and

propose a new method of measuring the performance of experimental design using an existing metric - distance correlation (DC) – to evaluate the orthogonality of experimental designs. Unlike the classic correlation concept, DC is able to capture hidden interdependence features, it is expected that using this criterion will give superior designs, in terms of orthogonality.

The numerical experiments and their results support our expectation. A simple optimization algorithm based on Florian's method along with different randomization methods is proposed for obtaining good designs rapidly. The time requirement of running the proposed algorithm has been tested for up to 100×27 design dimension, and it can produce nearly orthogonal designs in less than a minute using a personal computer as mentioned earlier. For comparison, we mention that in the study of Hernandez, Lucas and Carlyle (2012), the generation of a 16×12 design takes approximately 36 minutes (median runtime) using a laptop with 2 GB RAM, and using the General Algebraic Modeling System (GAMS Development Corporation, 2013) with CPLEX (a state-of-the-art integer programming solver engine from IBM).

In our numerical experiments, the best space-filling designs are obtained when the AE criterion is used but the designs obtained using the DC criterion produce designs with AE values that are very close to the AE-based search. This finding indicates that by using DC one can obtain nearly orthogonal as well as good space-filling designs quickly. The results can be used for proxy design and the latter can be used for subsequent optimization studies and other purposes.

To illustrate our approach, we also presented a borehole model. For this case based on our design approach we obtain improved results when compared to a recent topical study.

CHAPTER 3 A MODIFIED QUANTUM-BEHAVED PARTICLE OPTIMIZATION ALGORITHM FOR IMPROVED SINGLE OBJECTIVE OPTIMIZATION

Single objective optimization is an important field of research due to its fundamental role in complex problems, such as, multi-objective optimization. Quantum-behaved particle swarm optimization is an efficient heuristic optimization algorithm used for solving many diverse complex problems. However, like many other heuristics, it suffers from premature convergence resulting in suboptimal solution. In this paper, potential causes of local trapping are identified and relevant solutions are proposed accordingly. The proposed new algorithm incorporates Sobol initialization, re-initialization of algorithm parameter, perturbation of mean best positions and re-positioning of the particles. Numerical experiments are carried out using several simple to complex test functions each of 10, 30 and 50 dimensions, respectively. Performance of the proposed algorithm is compared with that of the standard QPSO algorithm. Where available, the best results found in the literature for different test functions are also compared. Experimental results show that the proposed algorithm shows promising performances. However, the proposed algorithm still suffers from dimensional issues that might be resolved by fine tuning the parameter(s). Two schemes are proposed for the proposed algorithm to extend for constrained optimization with mixed variables. A typical engineering problem has been solved and the results are compared with the results found in the literature. The proposed algorithm showed excellent performance in solving the mixed variable constrained engineering problem.

3.1 INTRODUCTION

Particle swarm optimization (PSO) was proposed as a heuristic optimization algorithm by Kennedy & Eberhart (1995) and Clerc & Kennedy (2002) emulating the success of searching behaviour of flocks of birds or schools of fishes. PSO algorithm is very simple to understand,

easy to implement and robust in operation while exhibiting excellent performance and, as a result, has been successfully used in many complex optimization problems as described in Fang et al. (2010) and the references therein. There is a vast and growing literature related to PSO and review articles by Banks et al. (2007), Banks et al. (2008) and Fang et al. (2010). Over time, many analyses of the algorithm have been carried out and different modifications to improve the performance of the algorithm have been proposed accordingly. These modifications meant to improve any or all of the categories that include *stability*, *premature termination*, *local entrapment*, *speed and parameters* of the algorithm.

Using discrete time-domain system theory, dynamic behaviour and convergence of simplified PSO algorithm was analyzed in Trelea (2003). It provided qualitative guidelines for parameter selection. Particles' trajectories were analyzed by van den Bergh and Engelbrecht (2006), stochastic convergence analysis was done by Jiang et al. (2007) and the conditions for selecting parameters are outlined accordingly. In Schutte et al. (2004) and Koh et al. (2006) parallelization or in Plevris and Papadrakakis (2011) hybridization with local search algorithms are incorporated to speed up the algorithm as well. However, van den Bergh (2001) showed that canonical PSO is not a global search algorithm. Moreover, like many other heuristic algorithms, PSO suffers from premature convergence resulting in suboptimal solutions. Different schemes have been proposed to overcome the entrapment at a local solution. For example, Suganthan (1999) divided the swarm into multiple 'neighborhoods', each maintaining its own local best solution that in turn makes the algorithm less prone to be trapped in local minima with slower convergence. Similarly, several other neighborhood topologies are proposed to enhance the exploration ability of PSO. Some researchers incorporated other mechanisms, like, chaotic behavior in Kaveh et al. (2014), and Levy process in Cai et al. (2007) to achieve the same goal. Inspired by quantum

mechanics and trajectory analysis of PSO, Sun et al. (2004) proposed a new version of PSO that samples new particles around the previous best particles utilizing the quantum δ potential well, known as quantum-behaved particle swarm optimization (QPSO). Later, QPSO algorithm is analyzed and improved by many researchers and a detailed discussion can be found in the review paper of Fang et al (2010). The iterative equation of QPSO is very different from that of PSO. Despite yielding better performance than PSO algorithm, QPSO also suffers from premature convergence to local optima. Therefore, like PSO, many schemes have been proposed for QPSO as well. Unfortunately, many proposed solutions increase computational demands of the algorithm significantly, like, calculations of diversity of particles' positions at each iteration. Moreover, initialization of the particles is rarely considered and to our knowledge, uniform initialization is invariably adopted in all the variants of QPSO algorithm. In this chapter, to enhance the performance, modifications are proposed to the basic QPSO algorithm addressing all the problems identified including initialization. The rest of the paper is organized as follows: In Sections 3.2 and 3.3, classical PSO and the basic QPSO algorithms are discussed, accordingly. In Section 3.3, different improvement schemes are described. Detailed analysis of QPSO algorithm is carried out in Section 3.4. Thereby, causes of local entrapment and suboptimal solution are identified. In Section 3.5, modifications are proposed considering all the identified problems in Section 3.4. In Section 3.6, numerical experiments are carried out using several well-known test functions. A mixed variable constrained pressure vessel design problem is considered to examine the efficacy of the proposed algorithm. The results obtained are compared with that of the basic QPSO algorithm and those obtained from literature using different algorithms and are analyzed accordingly.

3.2 PARTICLE SWARM ANALYSIS

In standard PSO, a swarm of N particles flies over the D-dimensional problem space to find the optimal solution where the position of each particle represents a potential solution of the optimization problem. The dimension of the problem is defined as the number of unknown variables considered for optimization. The velocity and position of each particle is updated according to the following equation at each iteration step.

$$v_{id}(t+1) = \omega v_{id}(t) + c_1\varphi_1\big(P_{id}(t) - x_{id}(t)\big) + c_2\varphi_2\left(P_{gd}(t) - x_{id}(t)\right) \quad (3.1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (3.2)$$

Where, v and x denote the velocity and position of a particle, t denotes the number of iterations. So, $x_{id}(t)$ is the position of ith particle in dth dimension at tth time. Similar notation is used for velocity. ω, c_1 and c_2 are algorithmic parameters and are known as inertia, cognitive and social constants, respectively. φ_1 and φ_2 are two independent random numbers uniformly distributed in $(0,1)$ for each dimension. P_{id} is the position with the best fitness found so far in dth dimension for the ith particle, also known as personal best or *pbest*. Similarly, P_{gd} is the position with the best fitness found so far in dth dimension for all the particles or swarm, also known as global best or *gbest*. In Clerc and Kennedy (2002), an alternative velocity update equation was proposed using constriction coefficient. According to Jiang et al. (2007), convergence property and performance of PSO depends on algorithmic parameters, velocity clamping, position clamping, topology of neighbourhood, etc. For high dimensional optimization problems, PSO algorithm can easily be trapped in local minima. Through different analyses, it is shown that if personal best (P_{id}) is not improved, change in global best (P_{gd}) does not take place and there is no change in searching direction. As a result, PSO cannot jump out from local minima. Different mechanisms are proposed to overcome the entrapment by i) changing control parameters, ii)

combining with other algorithms iii) perturbing current state of the particle, and iv) discarding old and introducing new particles.

However, linear equations in PSO cannot capture the complex interaction and quantum behaviour is incorporated to overcome the limitations of classical PSO accordingly, named quantum-PSO (QPSO).

3.3 QUANTUM-BEHAVED PARTICLE SWARM OPTIMIZATION (QPSO)

Like quantum models, in QPSO, the state of a particle is described by wavefunction $\psi(\bar{x}, t)$, as such, position and velocity of a particle cannot be determined simultaneously in quantum mechanical sense. Instead, particles are considered in quantum state and position of each particle is determined probabilistically using probability density function $|\psi(\bar{x}, t)|^2$.

Using trajectory analysis of basic PSO algorithm Clerk and Kennedy (2002) showed that each particle converges to a local attractor point p_{id}. The local attractor point depends on based on *pbest* and *gbest* positions, and is defined as:

$$p_{id} = \frac{c_1\varphi_1 P_{id} + c_2\varphi_2 P_{gd}}{c_1\varphi_1 + c_2\varphi_2} \tag{3.3}$$

Or

$$p_{id} = \varphi P_{id} + (1-\varphi)P_{gd}, \quad \varphi \sim U(0,1) \tag{3.4}$$

Assuming the particle follows quantum behavior, the probability distribution function of the location of a particle at any dimension can be obtained by solving Schrödinger equation for that dimension considering one dimensional Delta potential well centering at local attractor defined by equation 3.3 or 3.4. The probability distribution function *F* is:

$$F(y) = \int_{-\infty}^{y} Q(y)dy = e^{-2|q-x|/L} \quad (3.5)$$

Where, q is the center of the potential well and L is the search scope of the particle. Using Monte Carlo method the position of each particle is obtained:

$$x = q \pm \frac{L}{2}\ln(1/u), \quad u \sim U(0,1) \quad (3.6)$$

In Sun et al. (2004), the parameter L is evaluated as

$$L = 2\beta|q - x| \quad (3.7)$$

Where, β is called contraction-expansion (CE) coefficient, which controls the exploration and exploitation behavior of the swarm and plays an important role for the algorithm. Using the usual notations, the position update equation of QPSO algorithm becomes:

$$x_{ij}^{t+1} = p_{ij}^{t} \pm \beta\left|p_{ij}^{t} - x_{ij}^{t}\right| \ln(\frac{1}{u}) \quad (3.8)$$

Equation 3.8 is the basic QPSO algorithm with one algorithmic parameter β.

Later, individual local attractors were replaced with the mean of the personal best positions of the swarm, known as 'mainstream thought' or 'mean best position' as described in Sun et al. (2004) and Equation 3.8 for position update is changed accordingly,

$$x_{ij}^{t+1} = p_{ij}^{t} \pm \beta\left|mbest_{j}^{t} - x_{ij}^{t}\right| \ln(\frac{1}{u}) \quad (3.9)$$

Where, mean best position (*mbest*) is defined as

$$mbest = (\frac{1}{M}\sum_{i=1}^{M} P_{i1}, \frac{1}{M}\sum_{i=1}^{M} P_{i2}, \cdots, \frac{1}{M}\sum_{i=1}^{M} P_{iD}) \quad (3.10)$$

Where, M is the population size and P_i is the personal best position of particle i.

Exponential distribution of positions gives QPSO a wider search capability compared to normal distribution, resulting from quantum harmonic oscillator potential well and incorporating *mbest* position makes it more cooperative behavior as it does not leave lagged particles, eventually makes the algorithm better in global search ability as claimed in Fang et al. (2010).

The basic QPSO algorithm is described below.

```
Input: Swarm size (number of particles), Problem dimension,
Initial and final β, (and maximum number of iterations (maxIter)).
Initialize particle positions randomly, uniformly in each
dimension. Calculate maxIter.
For t= 1 to maxIter
     Compute the mean best position (mbest) using equation 3.10.
     β = finalβ + (finalβ − initβ) (maxIter−t)/maxIter
    For i = 1 to population size M
         If f(x_i) < f(P_i) then P_i = x_i Endif
         P_g = min(P_i)
         For j = 1 to dimension D
             φ = rand(0,1);  u = rand(0,1);
             p_ij = φ.P_ij + (1 − φ).P_gj
             If (rand(0,1) > 0.5)
                 x_ij = p_ij + β.abs(mbest_j − x_ij).ln(1/u)
             Else
                 x_ij = p_ij − β.abs(mbest_j − x_ij).ln(1/u)
             Endif
         Endfor //end for loop j
     Endfor //end for loop i
Endfor //end for loop t
```

Convergence analysis of QPSO particles in Sun et al. (2012) shows that $\beta \leq e^{\gamma} \approx 1.781$ can prevent explosion of the particles' position. The most commonly used values for β is to initially setting it to 1.0 and is reduced linearly to 0.5, Fang et al. (2010).

Different schemes have been proposed for improving the global search ability, convergence speed, solution precision, and robustness of the basic QPSO algorithm. These schemes range from changing the algorithmic parameter β proposed by Sun et al. (2005), Sun et al. (2012), by controlling the swarm diversity in Sun et al. (2006a, 2006b), Wang et al. (2013), and Sun et al. (2011), using a different distribution function in Sun et al. (2006), Coelho (2010), utilizing cooperative methods in Li et al. (2012), perturbing local attractor Sun et al. (2011), introducing mutation operator (Singh & Mahapatra, 2016),incorporating elititist breeding of the personal and global best particles (Yang, Wu, & Min, 2015), hybridizing with other algorithms, like quasi-Newton and QPSO by (Liu, Shen, & Lu, 2014), Bacterial Foraging Algorithm and QPSO by (Pradhan & Patra, 2015) and incorporating novel search methods, like chaotic search in Coelho (2008), Li et al. (2012). In the following section, evolution of different particles' positions is studied to find the causes for sub-optimal solution.

3.4 ANALYSIS OF QPSO

QPSO algorithm is analyzed through inspecting evolution of particles' positions at different time steps in Section 3.4.1. Impact of β is analyzed further to find out how it affects wasted function evaluation in Section 3.4.2.

3.4.1. Evolution of Particles' Positions

In this sub-section, evolution of particles' positions is studied visually for a 2-dimensional Shekel function with 10 local maxima using QPSO algorithm and different aspects of this algorithm is analyzed. The function and the corresponding contour map is shown in Figure 3.1.

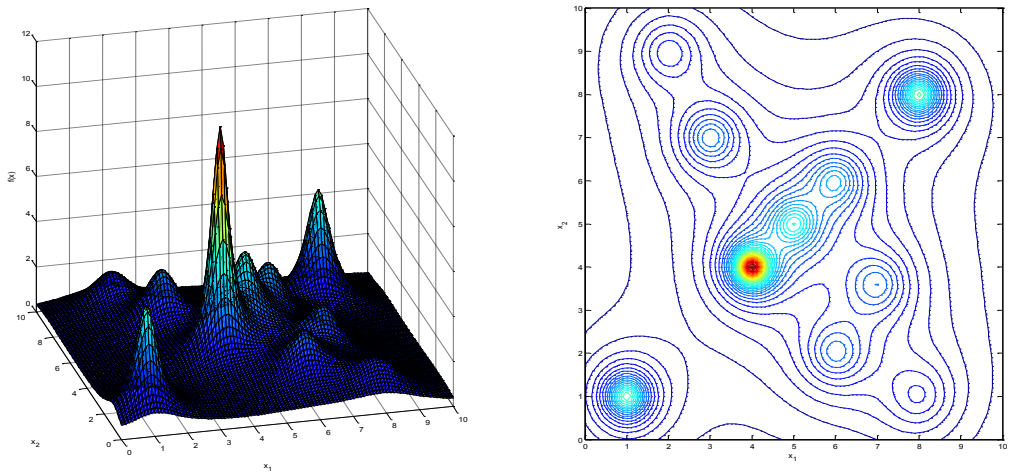

Figure 3.1: Surface and contour plot of 2D Shekel function with 10 local optima

In Figure 3.2, positions of different particles are shown just after initialization. Positions of the particles are initialized using uniform distribution (Mersenne-Twister algorithm) for each dimension. Each particle provides information about the problem space and uniform sampling provides best information in the absence of *a priori* knowledge of the problem. However, as seen in the figure, particles are not distributed uniformly over the whole region of the problem space. This problem might be more severe for higher dimensional cases. Global optimum is located near a local optima. As expected, the mean best position does not coincide with the *gbest*.

In Figure 3.3, evolution of the positions of different particles is shown over different iteration level. It is evident that initially (Figures 3.3a & 3.3b), particles are attracted towards a local optimum as no particle was within the basin of the global optimum point. However, once a particle reaches within the basin of attraction of the global optimal point, all the particles along with their personal bests start moving towards the global optimum point (Figures 3.3c & 3.3d). Even if one or more particles reaches global/local optimal point, the exploration capability of the swarm persists at an earlier stage of the optimization process (Figure 3.3e) due to random part of update Equation 3.9 and higher β value. However, eventually, diversity of particles' positions

diminishes (Figure 3.3f) and as the contraction-expansion coefficient β decreases at each iteration level, particles are unlikely to escape from the trapped position at the later stage of the optimization process (Figure 3.3g). Due to low β value, the swarm loses its exploration capability and can exploit only the local domain. Ultimately, all the particles reached an optimum location (Figure 3.3h). Even if the found optimal point is merely local, the algorithm can't escape the local entrapment due to small multiplication factor or flight range, as described below.

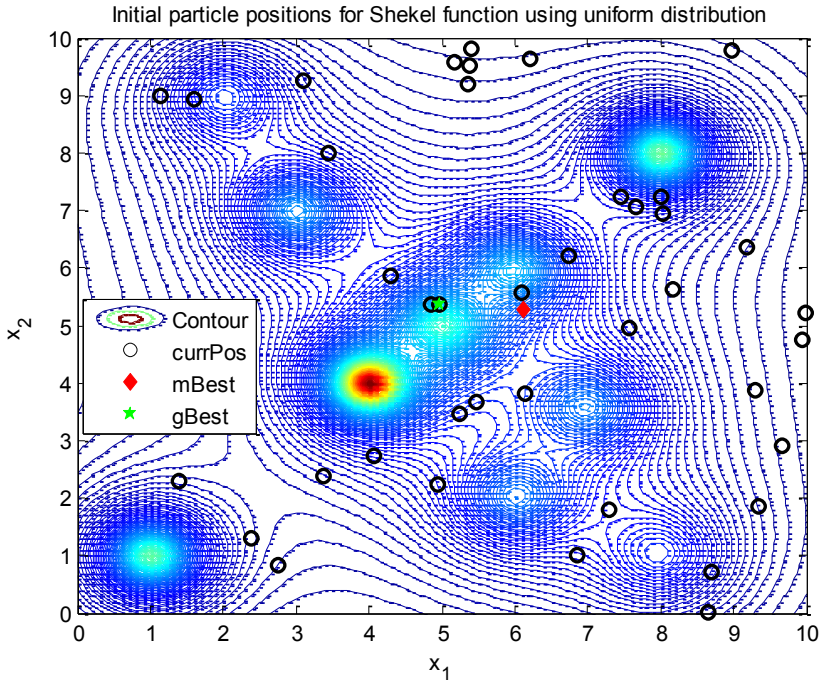

Figure 3.2: Locations of particles along with mean best, global best at initialization using uniform distribution

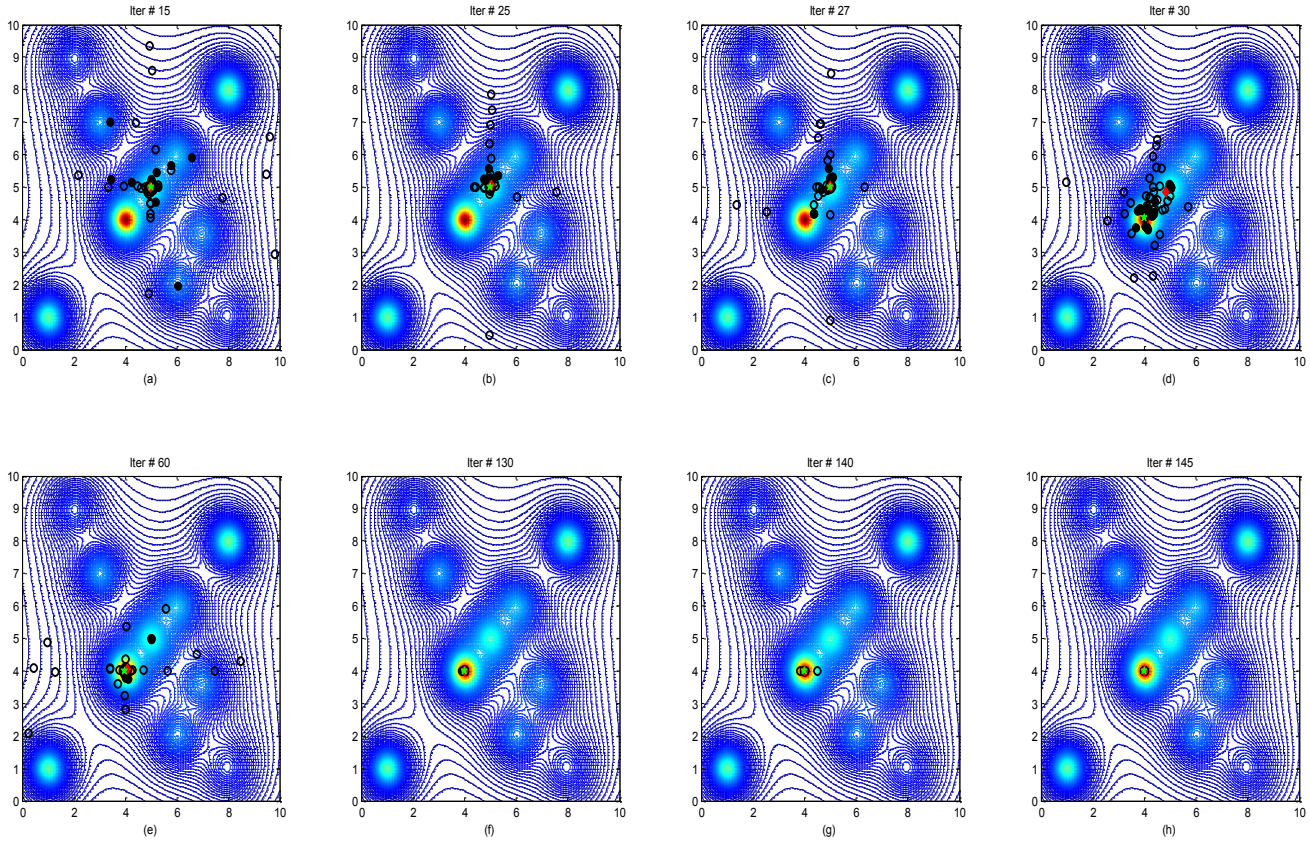

Figure 3.3: Evolution of particle positions along with personal best, mean best and global best positions at different iteration level

For further investigation, histogram of multiplication factor or flight range is shown in Figure 3.4, where, *multiplication factor* is defined as the multiplication of the random term and the contraction-expansion factor in QPSO algorithm in Equation 3.9 ($ln\left(\frac{1}{u}\right) \times \beta$). The histogram is shown in blue, green and red for $\beta = 1.7, 1.0$ and 0.4, respectively. As expected, the larger the β value is, the farther a particle can be placed from its current position at each iteration. Therefore, the searching capability of QPSO algorithm is localized as β decreases.

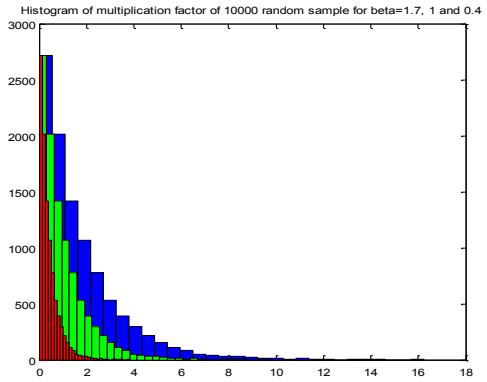

Figure 3.4: Histogram of multiplication factor for different contraction-expansion factor β for 10000 random samples

3.4.2. Impact of $\boldsymbol{\beta}$ on Function Evaluation

Apart from searching capacity, β might also affect the number of wasted function evaluations, explained below. It is common practice to place a particle at the boundary once the updated position using Equation 3.9 goes below or above the lower limit or upper limit of the search space, respectively. Depending on the locations of the particles and value of β, clipping to the limiting value can be quite frequent, as such, function evaluation is done using the same argument/position resulting in a wasted function evaluation. A simple experiment is carried out to understand the impact of β and the

location of global optimum (*gbest*) on the number of particles that exceeds the limits of a given search range.

Two possible global optimum locations are considered in this experiment, if the *gbest* is located closest to the minimum/maximum value, or to the centre (mean) of the given range. The value of β is changed from $1.7 - 0.2$. For a particular β, 40 uniformly random numbers are generated and those closest to the minimum and the mean value particles are considered as *gbest* particles for the two cases considered above. Following Equation 3.9, the new position of each particle is calculated and the number of particles that exceeds any of the limits is counted accordingly. The simulation is carried out for 10^6 times and the average number of occurrences recorded. In Figure 3.5, a fraction of particles that cross the limit after position update is plotted if the global best position is located nearest i) to the minimum (blue), and ii) to the centre (green) for 1D case. All 40 particles are assigned uniformly [-100, 100]. As seen in Figure 3.5, if the global best position is located near the minimum, the number of particles that exceed the limit is more than if the *gbest* is located near the centre. The experiment is carried out if *gbest* is located near the 1st and 3rd quartile and the maximum too. Results for the maximum case are similar to that of the minimum case and shows symmetry in quartile cases as well. As the value of β increases, the fraction of particles that overflow the boundaries increases.

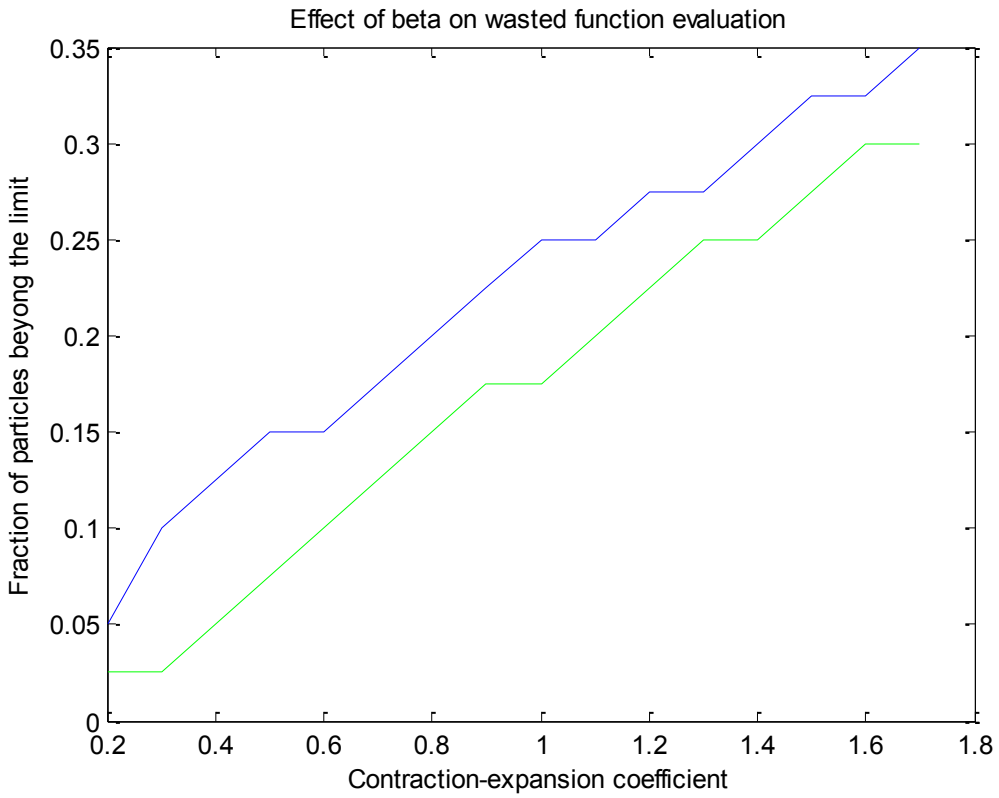

Figure 3.5: Fraction of particles exceeds limit based on β value for 1D case, 40 particles and range [-100, 100]

The same experiment is carried out for multiple dimensions with varying ranges. We did not find any noticeable difference in results due to changes in the boundary values. We tried the following ranges in our experiments: [-0.5, 0.5], $[-\pi, \pi]$, [-5, 5] and [-100, 100]. However, as the number of dimensions is increased, the fraction of particles exceeding limits increases too. In our simulation, the particle that exceeds the boundary for at least one dimension is considered violating boundary restriction. In Figure 3.6, the effect of dimension on the fraction of particles exceeds the limits when plotted for 2, 10 and 30 dimensions. As the number of dimension increases, the number of particles increases, violating boundary restrictions. As in a one dimensional case, if the global optimum is located near the boundary (solid lines), the fraction is always greater than that of the

cases where the optimal location is closer to the centre of the search space (dotted lines). However, it is evident that once a particle exceeds the limit of the search space, it is not a good idea to put the value just at the limit, especially, if the β is large, as a good number of particles might have the same upper/lower limit value resulting in wasteful function evaluation.

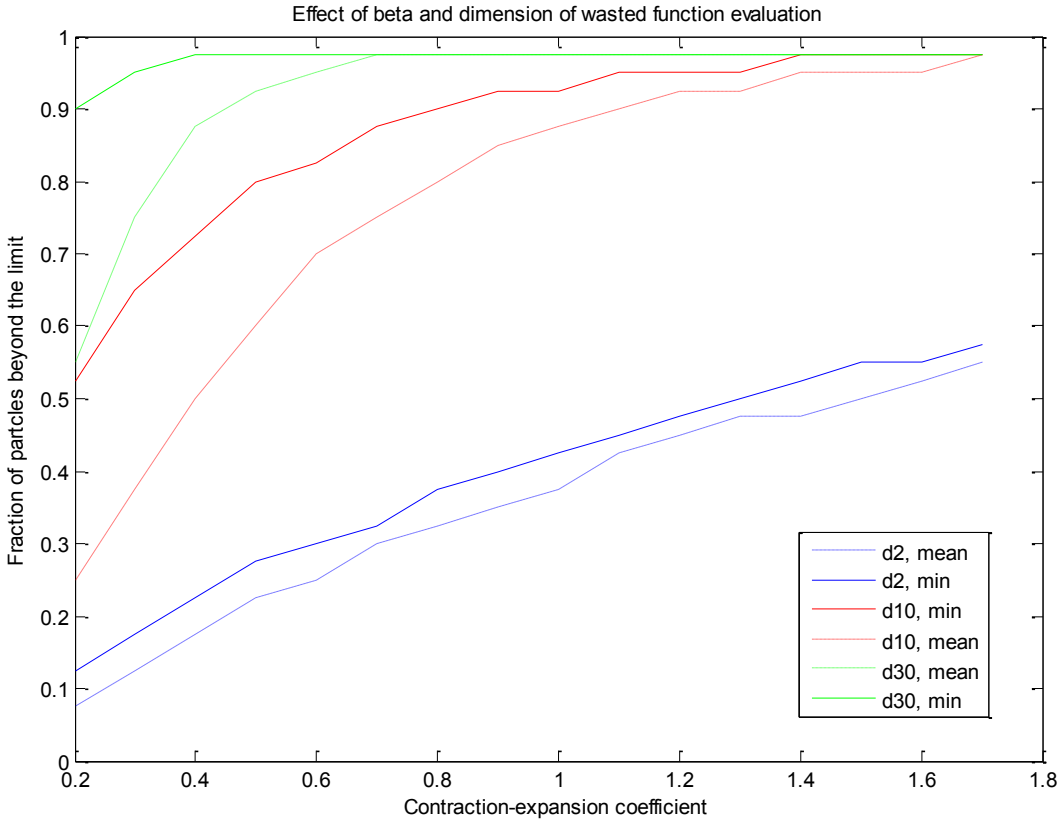

Figure 3.6: Fractions of particles exceeds limit for multidimensional cases with 40 particles and range [-100, 100]

3.4.3. Observations

From the above analyses and literature survey, the following observations are considered to improve the basic algorithm:

- If β is kept within a stable region (< 1.781) and a sufficient number of iterations is allowed, QPSO algorithm guarantees converge to an optimal location.
- At earlier stages of the experiment, when β has a larger value, the algorithm can escape from the local trap, but is unable to do so at later stages, as it loses the exploration capability depicted in Figure 3.4.
- While searching, if none of the particles reaches within a basin of attraction, the algorithm is unable to search that basin, even if the global optimum is located there. So, reaching all the basins should be the primary objective of any population-based heuristics. However, to our knowledge, no algorithm guarantees this 100% and QPSO is not an exception.
- It is not a good idea to put the particle position at the limiting value once the calculated new position exceeds the limit, especially, in multidimensional problems.
- The reasons particles converged to a local optimum can't escape even at high β could be because of the following reasons or their combinations:
    - The basin of the local optimum extends beyond the maximum jump provided by the algorithm. Though, 2 or more jumps in one direction might be sufficient to overcome this distance, due to the random nature of the algorithm, the consecutive jumps might change the direction, contributing to a very small change at each iteration or both.
    - The problem might have a very narrow region to search for global optimum (needle in a haystack) and very easy to be missed by any search algorithm.

- Due to *curse of dimensionality*, high dimensional problems are more prone to be trapped to local optimum as the change in one dimension might ruin all the gains achieved through changes in other dimensions.

Based on the above observations, the new algorithm should have the following characteristics:

1. Initial distribution of the particles should be as uniform as possible to cover the whole search domain.
2. For each iteration level, there should be some mechanism to search the unsearched region systematically.
3. Sufficient number of function evaluations should be carried out at each β level before decreasing it (exploitation), which depends on the dimension of the problem, to allow a complete search around a particular point. In this regard, β acts like the temperature in annealing method.
4. Once the particles converges to an optimum (local or global) and β is small enough to escape the trap, it should be reassigned to a larger value for exploration.
5. A method should be used to put the particle inside the feasible range other than putting it on the boundary once it exceeds the limit, like reflection.

3.5 PROPOSED MODIFICATIONS

A number of modifications are proposed to incorporate the aforementioned characteristics in basic QPSO algorithm and implemented in C# language. Sobol sequence is used for initialization for better uniformity (Sobol, 1967). C# code is developed based on Joe and Kuo (2003, 2008) that supports up to 21201 dimensions. Moreover, after initialization, a new point can be generated preserving finer uniformity.

Therefore, if new particles are generated using Sobol sequence generator the unsearched locations can be probed systematically. Unlike traditional QPSO where β is decreased after each iteration, if a better *pbest* is found at a certain iteration level, it is left unchanged. In fact, β is preserved for a certain number of iterations even if there is no improvement of any *pbest*. The number of iterations to preserve β is dimension-dependent and for this study, it is set at half the number of dimensions of the problem to be solved. Though, diversity of particles' position is a good indicator of convergence of particles to a global/local optimal point, it is computationally expensive. On the other hand, from the above analysis, we know that once the β value is small, QPSO can only do the local search efficiently. Therefore, once the β value reaches its minimum and for a certain number of iterations (for this study, number of dimensions), if there is no improvement for any of the *pbest*s, β is re-initialized to its initial higher value to boost up the exploring capability of the swarm. Once the second cycle is completed and the number of function evaluations still left, positions of the particles are either re-assigned using Sobol sequence generator or perturbed current positions using Stable distribution with $\alpha < 2.0$. It is to be noted that $\alpha = 1$, and $\alpha = {}^{1}/_{2}$ correspond to Cauchy and Lévy distributions, respectively. Moreover, calculated mean best at each iteration is based on some finite samples (number of particles). According to the Central Limit Theorem, this mean is not the true mean of the population, therefore, *mbest* is perturbed using Gaussian distribution with calculated mean as the mean, and cal_std/n as the standard deviation, where *cal_std* is the calculated standard deviation of all the *pbest*s and *n* is the number of particles. The flowchart of the proposed algorithm is shown in Figure 3.7.

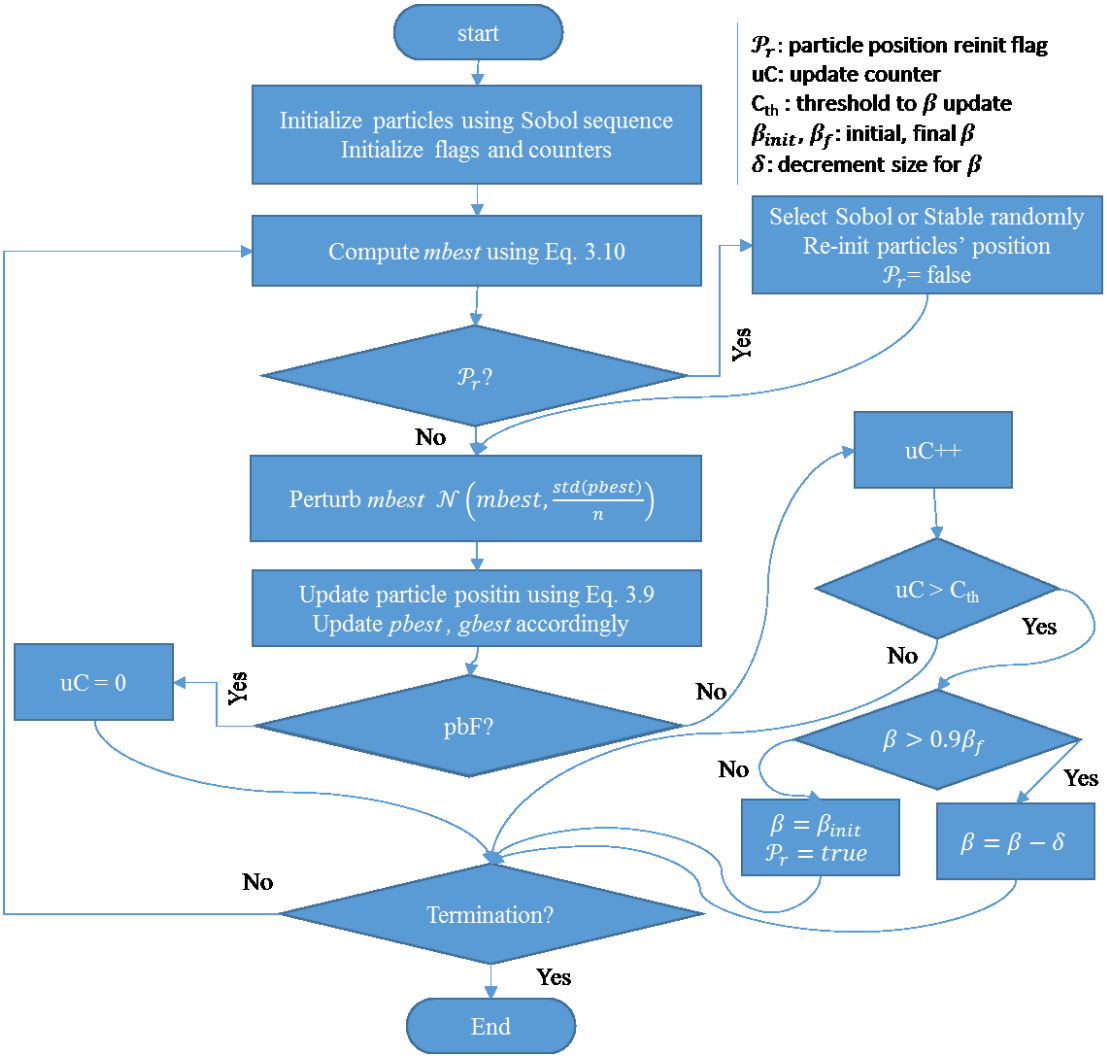

Figure 3.7: Flowchart of the proposed algorithm

In many cases, factors in an optimization problem contain both continuous and discrete values. Moreover, different constraints are also present in most engineering design and/or optimization problems. QPSO is targeted for the continuous value problems alone. To cope with the mixed-integer constrained optimization problems, different approaches are taken and interested readers might find them for PSO algorithm in the review paper by Jordehi and Jasni (2012). In our study, all the variables are considered continuous for position update equation. Once the position is calculated, the nearest discrete value is adopted for that discrete factor. For constraint handling either penalty is assigned to the

objective function for each constraint violation or priority is given to the particles without any constraint violation even if the calculated objective function is worse than that of the particles with violated constraints. More details follow in the constrained mixed-variable problem provided in Section 3.6.2.

3.6 EXPERIMENTAL STUDY

Performance of the proposed algorithm is tested using several benchmark functions discussed in Section 3.6.1. In Section 3.6.2, the performance of the proposed algorithm is compared with that of the basic QPSO and other algorithms found in literature in terms of the benchmark functions.

3.6.1 Benchmark Functions

Several scalable benchmark functions are considered in this paper. These functions are all widely used minimization problems in literature like, IEEE Congress on Evolutionary Computation (CEC10). These problems contain some basic functions and their shifted / rotated / shifted and rotated / hybrid combinations of other functions. In Table 3.1, all these functions are defined, along with their search domains.

Table 3.1: Benchmark function definition and search range

Function Name	**Function expression**	**Search Range**
Sphere	$f_1(\boldsymbol{x}) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$
Rosenbrock	$f_2(\boldsymbol{x}) = \sum_{i=1}^{D} 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2$	$[-100, 100]^D$
Rastrigin	$f_3(\boldsymbol{x}) = \sum_{i=1}^{D} x_i^2 - 10\cos(2\pi x_i) + 10$	$[-5, 5]^D$
Non-continuous Rastrigin	$f_4(\boldsymbol{x}) = \sum_{i=1}^{D} \lfloor x_i \rfloor^2 - 10\sin(2\pi \lfloor x_i \rfloor) + 10$	$[-5, 5]^D$

Function Name	**Function expression**	**Search Range**
Schwefel 1.2	$f_5(\boldsymbol{x}) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2$	$[-100,100]^D$
Griewangk	$f_6(\boldsymbol{x}) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600,600]^D$
Ackley	$f_7(\boldsymbol{x}) = -20e^{\left(-0.2\sqrt{\frac{1}{D}\sum_{i-1}^{D} x_i^2}\right)} - e^{\left(\frac{1}{D}\sum_{i-1}^{D}\cos(2\pi x_i)\right)} + 20 + e$	$[-32,32]^D$
Schaffer F6	$f_8(x,y) = 0.5 + \frac{\left(\sin\sqrt{x^2+y^2}\right)^2 - 0.5}{[1.0 + 0.001(x^2+y^2)]^2}$	$[-100,100]^D$
Weierstrass	$f_9(\boldsymbol{x}) = \sum_{i=1}^{D}\left(\sum_{k=0}^{20} 0.5^k \cos(2\pi.3^k.(x_i+0.5))\right) - D\sum_{k=0}^{20} 0.5^k\cos(2\pi.3^k.0.5)$	$[-0.5,0.5]^D$
Shifted Sphere	$f_{10}(x) = \sum_{i=1}^{D} \boldsymbol{z}_i^2 + bias,\ \ \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o}, bias = -450$	$[-100,100]^D$
Shifted Schwefel 1.2	$f_{11}(\boldsymbol{x}) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} z_j\right)^2 + bias_2,\ \ \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o},\ \ bias_2 = -450$	$[-100,100]^D$
Shifted Rotated High Conditional Elliptic Function	$f_{12}(\boldsymbol{x}) = \sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}} z_i^2 + bias_3,\ \ \boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{o}) * \boldsymbol{M},\ \ bias_3 = -450$	$[-100,100]^D$
Shifted Schwefel 1.2 with Noise	$f_{13}(\boldsymbol{x}) = \left(\sum_{i=1}^{D}\left(\sum_{j=1}^{i} z_j\right)^2\right) * (1 + 0.4\lvert N(0,1)\rvert) + bias_4, \quad \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o},\ \ bias_4 = -450$	$[-100,100]^D$
Shifted Schwefel 2.6	$f_{14}(\boldsymbol{x}) = \max\{\lvert \boldsymbol{A}_i\boldsymbol{x} - \boldsymbol{B}_i\rvert\} + bias_5, \quad bias_5 = -310$	$[-100,100]^D$
Shifted Rosenbrock	$f_{15}(\boldsymbol{x}) = \sum_{i=1}^{D} 100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2 + bias_6,\ \boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o}, \quad bias_6 = 390$	$[-100,100]^D$
Shifted Rotated Griewank	$f_{16}(\boldsymbol{x}) = \frac{1}{4000}\sum_{i=1}^{D} z_i^2 - \prod_{i=1}^{D}\cos\left(\frac{z_i}{\sqrt{i}}\right) + 1 + bias_7,\ \boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{o}) * \boldsymbol{M},\ \ bias_7 = -180$	$[-600,600]^D$
Shifted Rotated Ackley	$f_{17}(\boldsymbol{x}) = -20e^{\left(-0.2\sqrt{\frac{1}{D}\sum_{i-1}^{D} z_i^2}\right)} - e^{\left(\frac{1}{D}\sum_{i-1}^{D}\cos(2\pi z_i)\right)} + 20 + e + bias_8,\ \boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{o}) * \boldsymbol{M},\ \ bias_8 = -140$	$[-32,32]^D$
Shifted Rastrigin	$f_{18}(\boldsymbol{x}) = \sum_{i=1}^{D} z_i^2 - 10\cos(2\pi z_i) + 10 + bias_9,\ \boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{o}),\ \ bias_9 = -330$	$[-5,5]^D$

Function Name	**Function expression**	**Search Range**
Shifted Rotated Rastrigin	$f_{19}(\boldsymbol{x}) = \sum_{i=1}^{D} z_i^2 - 10\cos(2\pi z_i) + 10 + bias_{10},$ $\boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{o}) * \boldsymbol{M},\ bias_{10} = -330$	$[-5,5]^D$
Shifted Rotated Weierstrass	$f_{20}(\boldsymbol{x}) = \sum_{i=1}^{D}\left(\sum_{k=0}^{20} 0.5^k \cos(2\pi.3^k.(x_i + 0.5))\right) - D\sum_{k=0}^{20} 0.5^k \cos(2\pi.3^k.0.5) + bias_{10},$ $\boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{o}) * \boldsymbol{M},\ bias_{11} = 90$	$[-0.5,0.5]^D$
Schwefel 2.13	$f_{21}(\boldsymbol{x}) = \sum_{i=1}^{D} (\boldsymbol{A}_i - \boldsymbol{B_i}(\boldsymbol{x}))^2 + bias_{12}, bias_{12} = -460,$	$[-\pi,\pi]^D$
Shifted Expanded Griewank plus Rosenbrock	$f_{22}(\boldsymbol{x}) = f_6(f_2(z_1,z_2)) + f_6(f_2(z_2,z_3)) + \cdots + f_6(f_2(z_{D-1},z_D)) + f_6(f_2(z_D,z_1)) + bias_{13},$ $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{o} + \boldsymbol{1},\quad bias_{13} = -130$	$[-3,1]^D$
Shifted Rotated Expanded Schaffer F6	$f_{23}(\boldsymbol{x}) = f_8(z_1,z_2) + f_8(z_2,z_3) + \cdots + f_8(z_{D-1},z_D) + f_8(z_D,z_1) + bias_{14},$ $\boldsymbol{z} = (\boldsymbol{x} - \boldsymbol{o}) * \boldsymbol{M},\quad bias_{14} = -300$	$[-100,100]^D$
Hybrid Composition	$f_{24}(\boldsymbol{x})$= Hybrid composition of functions f_1, f_3, f_6, f_7, and f_9.	$[-5,5]^D$

Different features of the test functions are shown in Table 3.2 along with their true global optimum. The feature list includes whether the function is unimodal/multimodal (U/M), shifted (Sh), rotated (Ro), separable/Non-separable (Se/Ns), scalable (Sc), and if noise (No) is added. Comments are included for special features of a particular test function, if there are any.

Table 3.2: Different features of the standard test functions studied in this paper

Function	**Global Optimum**	**Features**	**Comment**
f_1	0	U, Convex	
f_2	0	U, valley/banana function	Global minimum lies in narrow, parabolic valley and convergence is difficult.

Function	Global Optimum	Features	Comment
f_3	0	M	Highly multimodal
f_4	0	M	
f_5	0	M	Complex with many local minima
f_6	0	M	Many widespread local minima
f_7	0	M	Nearly flat outer region and a large hole at the centre.
f_8	0	U	
f_9	0	M	
f_{10}	-450	U,Sh, Se, Sc	
f_{11}	-450	U, Sh, Ns, Sc	
f_{12}	-450	U, Sh, Ro, Ns, Sc	
f_{13}	-450	U, Sh, Ns, Sc, No	
f_{14}	-310	U, Ns, Sc	Global optimum is located at boundary
f_{15}	390	M, Sh, Ns, Sc	Very narrow valley from local to global minimum
f_{16}	-180	M, Ro, Sh, Ns, Sc	
f_{17}	-140	M, Ro, Sh, Ns, Sc	Global optimum at boundary
f_{18}	-330	M, Sh, Se, Sc	Huge number of local optima
f_{19}	-330	M, Sh, Ro, Ns, Sc	Huge number of local optima
f_{20}	90	M, Sh, Ro, Ns, Sc	
f_{21}	-460	M, Sh, Ns, Sc	
f_{22}	-130	M, Sh, Ns, Sc	
f_{23}	-300	M, Sh, Ns, Sc	
f_{24}	120	M, Sc	Separable near global optimum, huge number of local optima, two flat areas due to sphere function

3.6.2 Comparison with Standard QPSO and Other Algorithms

Each function is tested with three different dimensions, 10, 30 and 50. The maximum number of function evaluation (FE) is set as $10000 \times D$, where D is the dimension of the problem. So, maximum 100000, 300000 and 500000 number of function evaluations are allowed for dimensions 10, 30 and 50, respectively. For each set of experiments, 50 trial runs are carried out and 40, 60 or 80 particles are used as a single swarm depending on the 10, 30 or 50 dimensions of the problems, respectively.

In Table 3.3, results from the functions with 10 dimensions are shown. Mean and standard deviations of multiple runs of the results are tabulated. From the tabulated results, it is evident that, in most cases, the proposed algorithm outperformed the basic QPSO algorithm due to better exploration capability of the proposed algorithm and better initial distributions of the particles.

Table 3.3: Numerical results for different functions of dimension 10

Function	**QPSO**		**Proposed QPSO**	
	Mean	**Std**	**Mean**	**Std**
f_1	1.71×10^{-61}	5.47×10^{-61}	**0**	**0**
f_2	6.4544	6.8249	**0.8832**	**0.5340**
f_3	0.7431	0.8434	**0**	**0**
f_4	2.5427	0.9202	**0**	**0**
f_5	6.89×10^{-10}	1.72×10^{-09}	**0**	**0**
f_6	0.0811	0.0286	**0**	**0**
f_7	4.14×10^{-15}	6.96×10^{-16}	$\mathbf{4.44 \times 10^{-16}}$	**0**
f_8	0.5161	0.4231	**0**	**0**
f_9	0	0	**0**	**0**
f_{10}	**-450**	**0**	-450	8.74×10^{-14}

Function	QPSO		Proposed QPSO	
	Mean	**Std**	**Mean**	**Std**
f_{11}	**-449.9999**	$\mathbf{3.41 \times 10^{-13}}$	-449.9999	1.55×10^{-07}
f_{12}	**-450**	**0**	**-450**	**0**
f_{13}	0.0124	0.0162	$\mathbf{3.2 \times 10^{-06}}$	$\mathbf{4.82 \times 10^{-06}}$
f_{14}	-30.9487	352.6168	**-309.9999**	$\mathbf{3.69 \times 10^{-08}}$
f_{15}	399.9332	18.5156	**396.5553**	**6.3414**
f_{16}	-179.5205	0.3167	**-179.8151**	**0.1005**
f_{17}	**-119.6312**	**0.0721**	-119.6366	0.0661
f_{18}	-326.3515	1.6716	**-329.6870**	**0.7646**
f_{19}	-318.2585	5.0180	**-321.0837**	**4.9333**
f_{20}	**92.2605**	**0.9549**	92.4116	1.0949
f_{21}	321.4152	1328.9292	**58.0019**	**694.4023**
f_{22}	-129.3747	0.2638	**-129.311**	**0.3122**
f_{23}	-297.2250	0.5305	**-297.5159**	**0.5478**
f_{24}	365.1386	157.8967	**147.1859**	**79.9920**

For functions f_{10}, f_{11}, f_{17} and f_{20}, the results of the proposed algorithm are inferior to that of the standard QPSO algorithm. However, the results are very close. For example, the mean results are the same but the standard deviations are larger by 8.7447×10^{-14} and 1.554×10^{-07} for the functions f_{11} and f_{12}, respectively. Similarly, for function f_{17}, the mean value of QPSO is smaller by 0.0054 but the standard deviation is larger by 0.006. However, for function f_{20}, both the mean and the standard deviation of the proposed algorithm are inferior to that of the standard QPSO (off by 0.1511 and 0.14 for the mean and standard deviation, respectively). The proposed algorithm reinitializes β to a higher value. As stated earlier, larger β enhances exploring capacity but reduces exploitation ability. Therefore, if a new optimal location is found at the later stage with

larger β (due to re-initialization), there might not be enough iteration available for local search with smaller β. Moreover, for a few functions, namely f_{13}, f_{21} and f_{24}, global optimal positions were not found by both the algorithms. However, for function f_{14}, though the basic QPSO failed, the proposed algorithm could overcome the local trap.

In Table 3.4, the same experiments are carried out for the functions with 30 dimensions. However, apart from our experiments alone, the best results obtained using different PSO variants found in Sun et al. (2011) for different functions are also presented for better comparison. In this case, unlike Table 3.3 for dimension 10, the mean value is the differences between the best fitness value found by the algorithm and the actual global optimum after Suganthan, et al. (2005) and for better comparison. Although, the results show similar trends as dimension 10, the results are inferior in general due to the increase in number of dimension. For the basic functions, the proposed algorithm performed very well. For other functions, the proposed algorithm performed better for f_{12}, f_{13}, f_{19}, f_{21}, and f_{24}. The functions, namely, f_{20}, f_{22} and f_{23}, where the basic QPSO algorithm performs better, the results of the proposed algorithm are comparable. It is to be noted that, the best results from the reference are not from the same algorithm. Therefore, although the proposed algorithm looks inferior to the published results for many functions, the results are comparable, if not better, compared to that of the basic QPSO algorithm. However, as evident from the results, both algorithms suffer from the *curse of dimensionality*.

Table 3.4: Numerical results for different functions of dimension 30

Function	**QPSO**		**Best results found in Sun et al (2011)**		**Proposed QPSO**	
	Mean	**Std**	**Mean**	**std**	**Mean**	**Std**
f_1	1.75e-62	6.43e-62	-	-	**0**	**0**
f_2	41.1027	35.1226	-	-	**19.458**	**1.3775**
f_3	18.4399	5.3564	-	-	**0**	**0**
f_4	24.1694	5.0847	-	-	**0**	**0**
f_5	0.3498	0.2915	-	-	**0**	**0**
f_6	0.0063	0.0062	-	-	**0**	**0**
f_7	1.18e-14	4.04e-15	-	-	**4.44e-16**	**0**
f_8	6.8719	0.9151	-	-	**0**	**0**
f_9	**0**	**0**	-	-	**0**	**0**
f_{10}	5.49e-14	1.02e-14	**1.27e-27**	**3.71e-28**	1.10e-06	4.34-07
f_{11}	0.2068	0.2075	**0.0988**	**0.3362**	1.0445	0.3507
f_{12}	**0**	**0**	4.16e06	1.87e06	**0**	**0**
f_{13}	2109.5229	1175.1658	2.35e03	1.85e03	**1161.7485**	**636.8994**
f_{14}	4446.4296	944.3803	**2.53e03**	**947.8740**	2964.9074	570.9447
f_{15}	**47.7386**	**51.0449**	56.6257	90.6767	80.3121	67.5664
f_{16}	0.0214	0.0173	**0.0152**	**0.0125**	0.0456	0.0145
f_{17}	60.9444	0.0480	**1.56e-14**	**3.11e-15**	20.9492	0.0536
f_{18}	22.5524	5.2166	**0.6990**	**0.7983**	12.6554	3.7234
f_{19}	51.7090	15.0923	112.8426	71.2957	**46.3868**	**14.1507**
f_{20}	**13.7164**	**2.4156**	25.8903	3.1488	19.7979	5.5980
f_{21}	9299.2134	6996.2582	6.87e03	6.32e03	**5490.0109**	**4482.3580**
f_{22}	**3.0005**	**0.5495**	-	-	3.9647	2.3682
f_{23}	**11.8154**	**0.5993**	-	-	12.3577	0.4170
f_{24}	341.5774	171.2591	-	-	**334.3574**	**111.6796**

In Table 3.5, performance of the test functions are evaluated for 50 dimensions. Like 30 dimension cases, results are compared with that of Lim and Isa (2014). As seen, for basic functions except f_7, the proposed algorithm performs best among the 3 algorithms. The proposed algorithm shows mixed performance (better for the functions $f_{12} - f_{14}$, f_{17}, f_{19} and f_{24} but worse for other functions) for the more complex problems. However, the overall performance is degraded due to increased dimension.

Table 3.5: Numerical results for different functions for dimension 50

Function	**QPSO**		**Best found results**		**Proposed QPSO**	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**
f_1	2.34e-54	1.05e-53	0	0	**0**	**0**
f_2	71.9562	42.0583	4.22e01	2.39-01	**39.4252**	**0.6480**
f_3	35.8589	5.4041	0	0	**0**	**0**
f_4	54.4401	11.9407	0	0	**0**	**0**
f_5	73.4037	50.5503	0	0	**0**	**0**
f_6	0.0025	0.0044	0	0	**0**	**0**
f_7	2.82e-14	6.70e-15	**0**	**0**	4.44e-16	0
f_8	15.2902	1.5363	**-**	**-**	**0**	**0**
f_9	2.55e-06	7.94e-06	**0**	**0**	**0**	**0**
f_{10}	1.04e-13	2.57e-14	**5.68e-14**	**0**	0.0079	0.0017
f_{11}	**160.3748**	**80.3082**	-	-	351.7737	88.3264
f_{12}	**0**	**0**	1.62e06	7.45e05	**0**	**0**
f_{13}	12917.1865	3916.1453	-	-	**10187.7415**	**2355.2127**
f_{14}	7700.4212	1308.0993	-	-	**5307.2841**	**716.7731**
f_{15}	**81.7782**	**44.9233**	-	-	152.8812	137.0062
f_{16}	**4.37e-03**	**0.0102**	4.96e-03	7.37e-03	0.9675	0.1036
f_{17}	21.1472	0.0273	-	-	**21.1280**	**0.0261**
f_{18}	46.1662	8.1667	**1.75e-07**	**5.88e-08**	35.7007	8.1550

Function	QPSO		Best found results		Proposed QPSO	
	Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**
f_{19}	99.4096	32.3581	-	-	**89.2391**	**40.1354**
f_{20}	**26.6565**	**3.6851**	-	-	44.8142	9.9422
f_{21}	**38860.1489**	23254.1061	-	-	46930.9206	24413.3730
f_{22}	5.3212	0.8404	**1.15e00**	**4.49e-01**	7.6654	2.7459
f_{23}	**21.0124**	0.6774	-	-	22.1908	0.3805
f_{24}	302.7078	103.2192	-	-	**276.6212**	**75.9773**

3.7 PRESSURE VESSEL DESIGN – A MIXED INTEGER OPTIMIZATION PROBLEM

Pressure vessels usually hold gases or liquids at higher than the ambient pressure. A cylindrical pressure vessel with hemispherical caps at both ends is used as an engineering design and optimization problem. The objective of this problem is to minimize the total cost which includes welding cost, material cost and forming cost of the pressure vessel. There are four design variables: the shell thickness (T_s), the thickness of the head (T_h), the inner radius (R), and the length of the cylindrical section (L). Both thicknesses (T_s and T_h) can have only integer multiples of 0.0625 in., in accordance with the available thickness of rolled steel plates while R and L are continuous.

Being a well-known benchmark, the pressure vessel design problem is used extensively for validating optimization algorithms and details can be found in Yang et al. (2013) and the references therein. The optimization problem can be expressed as follows:

Minimize: $f(T_s, T_h, R, L) = 0.6224 T_s R L + 1.7781 T_h R^2 + 3.1661 T_s^2 L + 19.84 T_s^2 R$

Subject to (for minimum volume of 750 ft^3):

$$g_1 = -T_s + 0.0193R \leq 0$$

$$g_2 = -T_h + 0.0095R \leq 0$$

$$g_3 = -\pi R^2 L - \frac{4}{3}\pi R^3 + 750 \times 11728 \leq 0$$

$$g_4 = L - 240 \leq 0$$

Where, $1 \times 0.0625 \leq T_s, T_h \leq 99 \times 0.0625$ and $10 \leq R, L \leq 200$

To incorporate discrete variables in QPSO algorithm, a discrete variable is updated like a continuous variable. However, once a new value is obtained, the closest discrete value is used for the calculations of objective function and constraints. For handling of constraints, two different schemes were adopted. In the first method, an augmented objective function is used.

$$F(x) = f(x) + r \sum_{k=1}^{ncon} \max[0, g_k(x)] \tag{3.11}$$

Where, $F(x)$ is the augmented objective function, $f(x)$ is the original objective function, r is the penalty parameter for the constraints and g_k is the k-th constraint. In this study, $r = 10^8$ is used.

In the second scheme, no penalty function is used but an objective function without constraint violation is preferred over the same with violated constraint(s).

In Table 3.6, results for pressure vessel design are given for the proposed QPSO algorithm with 2 schemes of modification for discrete variables and constraint handling. Results from other algorithms, namely, PSO, QPSO and QPSO with chaotic mutation (C-QPSO), are used from Coelho (2008) for comparison purpose. For this study, 20 particles and 40000 function evaluations are used for each run and 50 trial runs are carried out for

each proposed scheme. It should be mentioned that, Coelho (2008) used 1000 function evaluation for the study. For fair comparison, results from both proposed schemes using 1000 function evaluations are provided in brackets. It is seen that, using both schemes, the obtained results are very close to the best result obtained by Hu et al. (2003) using 200000 function evaluation as shown in Table 3.7. Though obtained best results are inferior to that of those obtained using C-QPSO algorithm when only 1000 function evaluations are used, others indicators, like worst, mean and standard deviations are far better for both schemes. In Table 3.7, the best results obtained from our algorithm are compared with those of obtained by other researchers, like Coelho (2008); Hu et al. (2003). It can be concluded from the results that both schemes with proposed new QPSO algorithms are highly efficient to mixed discrete-continuous optimization problem, at least for pressure vessel design problem. Most importantly, the results are more consistent (smaller standard deviation) with the proposed algorithm. However, scheme 2 performs better if the number of function evaluations is smaller.

Table 3.6: Result of pressure vessel design problems in 50 runs

Algorithm	Worst	Best	Mean	Std
PSO	37206.3491	8329.4908	20327.7221	5601.1564
QPSO	34457.1969	6183.4325	11301.6617	6540.5048
C-QPSO	48505.9499	6112.5619	12352.7764	7610.4237
Proposed QPSO with scheme 1	7332.8415 (7854.8346)	6059.7143 (6209.5090)	6570.6340 (7036.4172)	375.4103 (361.5791)
Proposed QPSO with scheme 2	7332.8415 (8471.6174)	6059.7163 (6171.5080)	6479.3052 (6759.0852)	364.1335 (372.8248)

Table 3.7: Comparison of results for the design of a pressure vessel

Parameters	Hu et al	PSO	QPSO	CQPSO	Scheme 1	Scheme 2
Ts	0.8125	1.00	0.8125	0.8750	0.8125	0.8125
Th	0.4375	0.6875	0.4375	0.4375	0.4375	0.4375
R	42.0985	51.1288	51.1137	45.2802	42.0984	42.0984
L	176.6366	117.7362	189.2375	141.4063	176.6366	176.6368
g1	0.00	-1.32e-2	-1.9e-2	-1.09e-3	0.00	0.00
g2	-0.0359	-1.99e-1	-4.53e-2	-5.55e-3	-0.0859	-0.0859
g3	-5.82e-11	-2.31e5	-2.04e1	-3.70e3	-0.2640	-0.2640
g4	-63.3634	-1.22e2	-5.08e1	-9.86e1	0.00	0.00
f(x)	6059.1313	8329.4908	6183.4325	6112.5619	6059.7143	6059.7163

3.8 CONCLUSIONS

Modifications to the basic QPSO algorithm are proposed in terms of initialization, particle positioning, β-update and exploration enhancement. Initialization using low discrepancy number sequence using Sobol generator is proposed for more uniform distribution of the particles. Experimental design, like LHD or other low discrepancy numbers can be used instead. If the calculated new position crosses the given boundary, the usual practice of putting it on the boundary might result in wasted function evaluation. Therefore, reflected positioning is proposed for these particles. Contraction-expansion coefficient (β) plays a very crucial role for QPSO algorithm and controls the extent of randomness. In this study, it is considered as temperature and is allowed to decrease only when better solutions are not found for a few iterations, following simulated annealing method. However, once β is reduced to a very low value, it loses exploration capability. Therefore, it is re-initialized to a higher value. To explore a new location, new points are generated using Stable distribution or Sobol sequence.

Numerical experiments are carried out to find the efficiency of the proposed algorithm using several standard test functions along with their shifted, rotated and hybrid composition versions of dimensions 10, 30 and 50, respectively. Complexity of finding the global minimum in these test functions ranged from very simple to quite complex. Proposed modifications improve the performance of the basic algorithm in general but still suffer from *curse of dimensionality*. In many experiments, if the number of function evaluations is increased 10 times, the global optimum is found. Impact of threshold values to decrease β and the choice of re-initialization using Sobol or Stable distribution are not carried out in this study and will be considered in the future. Modifications are proposed for discrete constrained optimization and a practical problem is solved. Results are compared with that of the published results. Proposed algorithm is observed to perform better in most occasions.

CHAPTER 4 PARAMETER ESTIMATION OF VIRUS TRANSPORT USING KRIGGING METHOD

Simulation of virus transport process in groundwater is a valuable tool for prediction of movement and remediation of viruses in an aquifer. However, parameters of the governing equation of the virus transport process are to be estimated from the collected data of the observation wells. Being an inverse method, parameter estimation is a difficult problem and needs an optimization algorithm to find a proper solution. Moreover, in virus transport process in saturated porous media, the four parameters, namely, the linear distribution coefficient, the dispersion coefficient, the inactivation coefficients of aqueous and adsorbed viruses are known to be nonlinear and nonconvex and thus imposing more difficulties for estimation. Most importantly, almost all the optimization algorithms need a high number of simulation-runs that might be very expensive if a detailed model is used for forward simulation. As a remedy, a proxy model can be generated using the results from a limited number of expensive but detailed simulation-runs and the inexpensive proxy model is used for the optimization. In this paper, a kriging or Gaussian process (GP)-based proxy model is proposed to estimate the parameters of a virus transport process in saturated aquifer. The temporal vector data is converted to scalar objective function for the kriging model. The model is updated sequentially to improve the quality. The estimated parameters are comparable to that of other published results.

4.1 INTRODUCTION

Groundwater contamination with colloids including virus and bacteria might pose a risk to public health where it is used as the source of water for consumption. The sources of the contamination include but are not limited to leakage from septic tanks, sanitary landfills, sewerage pipes, and wastewater disposal from industrial and agricultural practices. However, septic tanks are considered to be the main source of virus contamination, Robertson et al. (1991). Once the water contaminated with pathogens reaches the water table through the vadose zone, the migration of pathogens follows the groundwater flow. Usually, the groundwater is consumed without any treatment. Therefore, it is crucial to know the range and level of spreading of viruses for prevention and remediation to avoid health hazards. The spatial and temporal spread of a virus can be numerically simulated by solving proper governing equations.

Virus transport in groundwater is a complex process and depends on various factors. For example, ionic strength and pH of soil water, physiochemical makeup of virion capsid (outer protective shell of virus), mineralogy of soil, composition and amount of organic matter in soil affects the sorption of virus while temperature, dissolved O_2, and saturation levels play an important role on survival of viruses Yates et al. (1985), Yates & Jury (1988), Powelson et al. (1990), Powelson et al. (1991), Powelson & Gerba (1994). Electrostatic attraction and repulsion, van der Waals forces, covalent-ionic interactions, and hydrophobic effects play roles for sorption of viruses Yates & Jury (1988). For modeling, it is considered that the fate and transport of viruses in groundwater are governed by four main processes: *advection*, *hydrodynamic dispersion*, *inactivation processes*, and *adsorption* by the solid matrix (Bhattacharjya et al. (2014). The first two

processes are related to the fluid flow through porous media while the last two processes are considered the most important processes controlling virus mobility directly related to the virus itself. Accordingly, several mathematical models have been proposed considering virus attachment onto the solid matrix and inactivation constants in solid and aqueous media respectively Vilker et al. (1978), Tim & Mostaghimi (1991), Powelson et al. (1993), Redman et al. (2001), Yates & Yates (1988), and Sim & Chrysikopoulos (2000).

It is known that the pathogenic viruses might travel a large distance at field scale and detailed numerical simulation of virus transport would take a long time. On the other hand, the parameters of the processes must be estimated from the available data for predicting the fate and transport of viruses in an aquifer. Parameter estimation can be carried out using an inverse optimization model. In this approach, the difference between the virus concentration of observed and simulated cases are minimized using an optimization model. However, as reported in the literature, gradient-based optimization methods are not suitable for estimating all the parameters simultaneously of the virus transport process Bhattacharjya et al. (2014), Ratha et al. (2009), Prasad et al. (2012). Moreover, local search methods, like simplex method, might fail as they cannot avoid local optima. As a result, as an efficient heuristic, genetic algorithm (GA) has been used to find the global near-optimal solution and further refined by simplex method to find the true parameters by Bhattacharjya et al. (2014). Unfortunately, heuristic optimization methods, including GA, typically require thousands of function evaluations. Therefore, objective function evaluation for optimization using straightforward field scale simulation might be computationally demanding and excessively time consuming; as

such, parameter estimation might not be feasible for such cases. In this chapter, a kriging or Gaussian process (GP)-based surrogate or proxy model is proposed to circumvent this problem. Our proposed method is tested for virus transport in one-dimensional, homogeneous and saturated porous media. The results from numerical models are used to create the proxy model and the light-weight proxy model is used for optimization. The developed method shows promising results to be used for field-scale parameter estimation for virus transport problems.

4.2 GOVERNING EQUATION AND SIMULATION

Virus transport in groundwater is based on advection-dispersion equation, with terms for retention in solid-water interface, and inactivation. The one-dimensional virus transport in homogeneous, saturated porous media with first order adsorption and inactivation can be written following Sim & Chrysikopoulos (1995):

$$\frac{\partial C}{\partial t} + \frac{\rho}{\theta}\frac{\partial C^*}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - V\frac{\partial C}{\partial x} - \lambda C - \lambda^*\frac{\rho}{\theta}C^* \tag{4.1}$$

Where C is the concentration of virus in aqueous phase $[M/L^3]$, C^* is the concentration of virus adsorbed in solid matrix $[M/M]$, D is the hydrodynamic dispersion coefficient $[L^2/T]$, V is the average interstitial velocity $[L/T]$, ρ is the bulk density of the solid matrix $[M/L^3]$, λ is the inactivation constant of suspended viruses $[T^{-1}]$, λ^* is the inactivation constant of adsorbed viruses $[T^{-1}]$, θ is the porosity of the soil media $[L^3/L^3]$, and t is time $[T]$.

Assuming equilibrium adsorption process, the solid phase concentration C^* is linearly proportional to the equilibrium liquid phase concentration C, i.e., $C^* = k_d C$, where, k_d is the linear distribution coefficient. Then the virus transport equation reduces to,

$$R\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - V\frac{\partial C}{\partial x} - \lambda C - \lambda^* \frac{\rho k_d}{\theta} C \tag{4.2}$$

Where $R = 1 + \frac{\rho k_d}{\theta}$ is the retardation coefficient.

Initial virus concentration is assumed to be zero, i.e.,

$$C(x, 0) = 0$$

At the upstream boundary, constant concentration is assumed which implies,

$$C(0, t) = C_0$$

Where, C_0= concentration at source.

The downstream boundary condition is

$$\frac{\partial C(\infty, t)}{\partial x} = 0$$

That preserves concentration continuity for a semi-infinite system.

The MATLAB partial differential equation (PDE) solver is used to solve 1-dimensional initial-boundary value problems in this study. The governing equation to be solved by the MATLAB PDE solver has to be written in the standard form specified in The MathWorks Inc. (2014). Comparing the standard form required by Matlab and the virus transport

equation, the following equations are obtained to solve the initial-boundary value problem.

$$c\left(x,t,u,\frac{\partial u}{\partial x}\right) = R$$

$$f\left(x,t,u,\frac{\partial u}{\partial x}\right) = D\frac{\partial C}{\partial x} \tag{4.3}$$

$$s\left(x,t,u,\frac{\partial u}{\partial x}\right) = -v\frac{\partial C}{\partial x} - \lambda C - \lambda^* \frac{\rho}{\theta} C^*$$

$$m = 0$$

4.2.1 Kriging / Gaussian Process (GP) Methodology

In kriging/GP metamodeling, the simulation output is considered as a random process $Z(\boldsymbol{x})$, where $\boldsymbol{x} = \left(x_1, \cdots, x_p\right)^T$ is a p-dimensional input vector for the simulation. Suppose the simulation model is run for m combinations of the input and gives the outputs of vector $\boldsymbol{y} = (y_1, \cdots, y_m)^T$. Like most authors in simulation, *ordinary kriging* is used in this study as shown in Equation 4.4.

$$Y(x) = \mu + Z(x) \tag{4.4}$$

Where, μ represents unknown constant trend or mean output, also known as large scale variation and $Z(\cdot)$ is a zero-mean stationary Gaussian process. Being stationary, $Z(\cdot)$ has an unknown but constant variance σ_z^2 and its covariance σ_{ij} $(i, j = 1, \cdots, m)$ between the outputs of the input combinations $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are solely determined by the distance between these p-dimensional points. The elements of the resulting covariance matrix are considered as the products of the k individual correlation functions. For example, a

Gaussian correlation function implies the elements of the following correlation matrix where $\sigma_{ij} = \sigma_z^2 R_{ij}$ and,

$$R_{ij} = \sigma_z^2 exp\left(-\sum_{k=1}^{p} \theta_k \left(x_{ik} - x_{jk}\right)^2\right) = \sigma_z^2 \prod_{k=1}^{p} exp\left|-\theta_k \left(x_{ik} - x_{jk}\right)^2\right| \qquad (4.5)$$

Where, $\theta_k \geq 0$ measure the importance or sensitivity of the kth input dimension; x_{ki} is the ith entry of the kth input and $|x_{ki} - x_{kj}|$ is the distance between the ith and jth entry at kth dimension. The symmetric and positive-definite $m \times m$ covariance matrix $\boldsymbol{\Sigma}$ has entries from covariances σ_{ij} and is related to the correlation matrix $\boldsymbol{R}$ as $\boldsymbol{\Sigma} = \sigma_z^2 \boldsymbol{R}$**.** It should be noted that for any $\theta_k \to \infty$ the correlation reduces to zero implying the outputs at locations i and j are independent. On the other hand, if $\theta_k = 0$ then changes in input k has no effect on the correlation. These two extreme values result in a singular matrix due to identical columns as mentioned in Kleijnen and Mehdad (2014).

Although kriging predictors assumes known parameters of the model, in reality, mean and covariance structure are *estimated* from the known/given data. Considering Gaussian process (GP), the density function f of output $\boldsymbol{y}$ is expressed as multivariate normal:

$$f(\boldsymbol{y}) = \frac{1}{(2\pi)^{\frac{m}{2}}(|\Sigma|)^{\frac{1}{2}}} \exp\left[\frac{1}{2}(\boldsymbol{y} - \mu\mathbf{1})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \mu\mathbf{1})\right] \qquad (4.6)$$

$$= \frac{1}{\left(2\pi\sigma_z^2\right)^{\frac{m}{2}}(|\mathrm{R}|)^{\frac{1}{2}}} \exp\left[\frac{1}{2\sigma_z^2}(\boldsymbol{y} - \mu\mathbf{1})^T \mathbf{R}^{-1}(\boldsymbol{y} - \mu\mathbf{1})\right]$$

Where, $|\Sigma|(|\mathrm{R}|)$ denotes the determinant of $\boldsymbol{\Sigma}(\boldsymbol{R})$ and $\mathbf{1}$ denotes the m-dimensional vector of ones.

Once a particular correlation model is selected, the parameters (θ in a Gaussian correlation model, for example) are estimated using maximum likelihood estimation (MLE) method following Santner (2003). According to Cressie (1993) the likelihood function for the deterministic kriging model is:

$$l(\theta) = \frac{m}{2}\ln \sigma_z^2 + \frac{1}{2}\ln|\boldsymbol{R}| + \frac{1}{2}\frac{(\boldsymbol{y}-\mu\mathbf{1})^T\mathbf{R}^{-1}(\boldsymbol{y}-\mu\mathbf{1})}{\sigma_z^2} \tag{4.7}$$

Removing the constant term and following Forrester and Keane (2009), the concentrated log-likelihood function that is to be minimized can be written as:

$$\min_{\hat{\theta}}\left[m\ln\widehat{\sigma_z^2} + \ln\left|\widehat{\boldsymbol{R}}\right|\right] \tag{4.8}$$

The likelihood function is minimized using the following steps:

1. Initialize: select preliminary values of $\hat{\theta}\,(\widehat{\theta_1},\cdots,\widehat{\theta_p})$ and calculate $\widehat{\mathbf{R}}$ accordingly
2. Compute GLS estimation of mean using

$$\hat{\mu} = \frac{\mathbf{1}^T\widehat{\boldsymbol{R}}^{-1}\boldsymbol{y}}{\mathbf{1}^T\widehat{\boldsymbol{R}}^{-1}\mathbf{1}} \tag{4.9}$$

3. Substitute estimated mean and correlation for variance estimate:

$$\sigma_z^2(\mathbf{x}) = \frac{(\boldsymbol{y}-\hat{\mu}\mathbf{1})^T\widehat{\mathbf{R}}^{-1}(\boldsymbol{y}-\hat{\mu}\mathbf{1})}{m} \tag{4.10}$$

   Note that, for calculation of σ_z^2, unlike classic unbiased estimator m is used instead of $m-1$ at the denominator.
4. Solve the minimization problem using Equation 4.7
5. Use the optimized $\hat{\theta}$ values to calculate mean and variance using Equations 4.9 & 4.10 in steps 2 and 3, respectively.

Minimization problem of likelihood function in step 4 is a difficult task as it is non-convex and multimodal in nature. The estimated kriging parameters are used for prediction and variance calculation of the predicted value.

In Ordinary Kriging (OK) methodology, the predicted response $\hat{y}_0$ at point $\boldsymbol{x}_0$ can be calculated using best linear unbiased predictor (BLUP) in terms of mean squared prediction error (MSPE),

$$\begin{aligned}\hat{y}_0 &= \hat{\mu} + \boldsymbol{r}^T \boldsymbol{R}^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu}) \\ &= \hat{\mu} + \boldsymbol{c_0}^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{y} - \mathbf{1}\hat{\mu})\end{aligned} \tag{4.11}$$

Where, $\hat{\mu}$ is the generalized least square estimator of the mean μ shown in Equation 4.9, $\boldsymbol{r}(\boldsymbol{c_0})$ is the vector of correlations (covariance) between the outputs at new and the old m locations, and $(\boldsymbol{y} - \mathbf{1}\hat{\mu})$ is the vector of residuals. If one of the known points is used for prediction, then the predictor $\widehat{y_i}$ equals the observed output y_i, which implies that Kriging is an exact interpolator.

One of the key benefits of kriging is its ability to provide an estimated error along with the prediction. The estimated MSPE of the kriging predictor is calculated based on Forrester and Keane (2009); Kleijnen and Mehdad (2014).

$$\begin{aligned}\widehat{MSPE} &= \widehat{\sigma_z^2} + \left(\mathbf{1}^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{c}}_0\right)^T \left(\mathbf{1}^T \hat{\boldsymbol{\Sigma}}^{-1} \mathbf{1}\right)^{-1} \left(\mathbf{1}^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{c}}_0\right) - \left(\hat{\boldsymbol{c}}_0{}^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{c}}_0\right) \\ &= \widehat{\sigma_z^2} \left[1 - \hat{\boldsymbol{r}}^T \hat{\boldsymbol{R}}^{-1} \hat{\boldsymbol{r}} + \frac{1 - \mathbf{1}^T \hat{\boldsymbol{R}}^{-1} \hat{\boldsymbol{r}}}{\mathbf{1}^T \hat{\boldsymbol{R}}^{-1} \mathbf{1}}\right]\end{aligned} \tag{4.12}$$

4.3 SEQUENTIAL DESIGN

Initial experiments are carried out using DoE as mentioned in Chapter 2. However, the main objective of DoE is to capture the underlying process as uniformly as possible. Therefore, many interesting features of the problems are not properly addressed as the response is not known *a priori* for DoE. However, once the experiments/simulations are carried out according to the DoE, the underlying behavior is known better and the interesting feature space in explored and/or exploited according to the objective. In Section 4.3.1, the mechanisms of adding new experimental points in the design are explained.

4.3.1 Infill Criteria for Sequential Design:

Surrogates are used in place of detailed simulation as the latter is expensive and should be used as little as possible. Therefore, an initial surrogate is made based on the data available and more experimental points are added sequentially after judicious selection using the surrogate model. These new experimental points are known as *infill points*. A simulation run is called using the infill data and the surrogate model is updated accordingly. Associated data, like the correlation function parameters, is updated as well. This process is continued until the stopping criterion is met, for example, MPSE is less than a threshold value. Therefore, optimization and search for a new design point goes hand in hand. Actually, success or failure of surrogate-based optimization rests on the correct choice of model and infill criteria as mentioned by Forrester and Keane (2009). Optimization using surrogate model and updating the model sequentially in fact falls under Bayesian optimization approach and can be described by the following algorithm.

1. For $t = 1, 2, \cdots$ Do
2. Find $\boldsymbol{x}_t$ from the posterior (surrogate model) that maximizes the utility function $u: \max_x u(\boldsymbol{x}|D_{1:t-1})$.
3. Sample the objective function: $y_t = f(\boldsymbol{x}_t) + \boldsymbol{\varepsilon}_t$
4. Augment the data, $D_{1:t} = \{D_{1:t-1}, (x_t, y_t)\}$ and update the model
5. End For

The utility function in step 2 is also known as *acquisition function*, *infill criteria* or *figure of merit* in the literature. Each infill criterion combines exploration (high-variance regions) and exploitation (high-mean regions). Several infill criteria for sequential surrogate improvement are proposed in literature and expected improvement (EI) is one of the most efficient methods. In this study, EI is used to verify the efficacy of parameter estimation of virus transport problems using kriging surrogate models.

EI is a popular infill criteria and has been shown to be efficient if the initial design is not too sparse or deceptive, Forrester and Jones (2008). However, EI function is often multimodal and difficult to maximize, Franey et al. (2011).

Let the best function evaluated so far be y_{min}. The response of the function is considered as a realization of Gaussian random variable $Y(\boldsymbol{x})$ with variance $s^2(\boldsymbol{x})$ implying $y(x) \sim \mathcal{N}(\hat{y}(\boldsymbol{x}), s^2(\boldsymbol{x}))$. The improvement over the current best point is defined as,

$$I(\boldsymbol{x}) = \max\{0, y_{min} - y(\boldsymbol{x})\} \tag{4.13}$$

Using kriging to predict $\hat{y}$ (estimated response) and $\hat{s}$ (estimated mean prediction error) the expected improvement at point $\boldsymbol{x}$ is defined as

$$E[I(\boldsymbol{x})] = \begin{cases} (y_{min} - \hat{y}(x))\Phi(u) + \hat{s}(\boldsymbol{x})\phi(u) & if\ \hat{s} > 0 \\ 0 & otherwise \end{cases} \quad (4.14)$$

Where, $\Phi(\cdot)$ and $\phi(\cdot)$ are cumulative distribution function (cdf) and probability density function (pdf) of standard normal distribution, respectively and

$$u = \frac{y_{min} - \hat{y}(x)}{\hat{s}(x)} \quad (4.15)$$

EI criterion exhibits balance between 'local' and 'global' search. The first term in EI equation supports local search around the current best (exploitation) while the second term emphasizes the regions with greater uncertainty or the unexplored regions and thus the global search (exploration).

4.4 ESTIMATION OF PARAMETERS

The parameters of the virus transport process can be estimated using an inverse technique, where a forward simulation model that solves the governing equation of virus transport process in groundwater is incorporated with an optimization model. In simulation-optimization technique, the optimization model calls the simulation model when it requires any information from the simulation model and has been successfully applied to large-scale groundwater management models as described in Finney et al. (1992), Emch and Yeh (1998), and Bhattacharjya and Datta (2005). The simulation model calculates virus concentration using different transport parameters while the optimization model minimizes the difference between the simulated and observed virus concentration at given observation locations. The optimization formulation may be written as

$$\min f(k_d, D, \lambda, \lambda^*) = \sum_{i=1}^{N} \sum_{j=1}^{T} \left(C_{oij} - C_{sij}\right)^2 \quad (4.16)$$

Subject to the following constraints,

$$k_d^{min} \leq k_d \leq k_d^{max}$$

$$D^{min} \leq D \leq D^{max}$$

$$\lambda^{min} \leq \lambda \leq \lambda^{max}$$

$$\lambda^{*min} \leq \lambda^* \leq \lambda^{*max}$$

Where, C_{oij} and C_{sij} are the observed and simulated virus concentration at ith location and jth time, respectively and $\{k_d, D, \lambda, \lambda^*\}$ is the parameter vector with respective lower and upper bounds represented as constraints.

However, as mentioned before, calling the expensive simulation module too many times for optimization might not be a feasible task. Therefore, a kriging based proxy model is created using the results from limited number of simulation runs. Design of experiment (DoE) methodology is applied to the parameter vectors considering their limits and number of simulation budget. A proxy model is created based on the available simulation data. However, kriging-based proxy models deal with scalar output while the virus transport simulation produces vector(s) of temporal output. The number of such vectors depends on the number of observation locations and the length of each vector depends on the number of temporal steps taken for each observation. Therefore, the vector(s) of output is(are) converted to a scalar by aggregating the error using different distance metrics, namely, Manhattan distance or ℓ_1 norm, Euclidean distance or ℓ_2 norm, Chebyshev distance or infinity norm and Hausdorff distance. Eventually, the error

surface is created against the parameters of interest and the above optimization problem in Equation 4.16 is converted to find the minimum error in the fitted kriging model.

4.5 NUMERICAL EXPERIMENTS & RESULTS

The parameter estimation procedure discussed in Section 4.4 is carried out in this section. First, the simulator for virus transport in groundwater is validated in Section 4.5.1 followed by the results of the estimated parameters in Section 4.5.2.

4.5.1 Validation of Simulation Model

The Matlab PDE solver is used to simulate the virus transport process in a saturated, homogeneous aquifer. If the inactivation constants are ignored in Equation 4.2, the governing equation for virus transport in groundwater, the resulting equation becomes the governing equation for chemical transport with advection, dispersion and retardation with available analytic solution for the later (van Genuchten & Alves, 1982). For the following governing equation and related initial-boundary conditions

$$R\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2} - V\frac{\partial C}{\partial x}$$

$$C(x,0) = 0, \qquad C(0,t) = C_0 \;\; 0 < t \leq T, \qquad \frac{\partial C(L,t)}{\partial x} = 0$$

The analytic solution is given by van Genuchten and Alves (1982, p. 12),

$$c(x,t) = \frac{C_0}{2}\left[\operatorname{erfc}\left(\frac{Rx - Vt}{2\sqrt{DRt}}\right) + \exp\left(\frac{Vx}{D}\right)\operatorname{erfc}\left(\frac{Rx + Vt}{2\sqrt{DRt}}\right)\right.$$
$$+\left[2 + \frac{V(2L-x)}{D} + \frac{V^2 t}{DR}\right]\exp\left(\frac{VL}{D}\right)\operatorname{erfc}\left(\frac{R(2L-x)+Vt}{2\sqrt{DRt}}\right)$$
$$\left. - 2\left(\frac{V^2 t}{\pi DR}\right)^{\frac{1}{2}}\exp\left[\frac{VL}{D} - \frac{R}{4Dt}\left(2L - x + \frac{Vt}{R}\right)^2\right]\right]$$

The parameters used for the numerical simulation are flow velocity of 34 cm/day, density of soil 1.1 g/cm^3, length of aquifer 120 cm, $k_d = 0.02$ ml/gm, hydrodynamic dispersion coefficient 34 m^2/s and porosity of 0.4. The inactivation coefficients are set to zero in order to compare the results with that of the analytic solution found in van Genuchten and Alves (1982). In Figure 4.1, both the simulation results and analytic solutions for virus concentration profiles at times 2 days and 2.5 days are plotted along the aquifer length. From the figure, it is evident that the results obtained from the numerical simulation are in good agreement with that of the analytic solutions. Moreover, the concentration profile of virus after 2 and 2.5 days are obtained numerically considering inactivation coefficients and plotted in Figure 4.2. In this case the value of both λ and λ^* are set to 0.58/day. The results are intuitive. Therefore, the numerical model developed with Matlab PDE solver is capable of simulating the virus transport process in an aquifer.

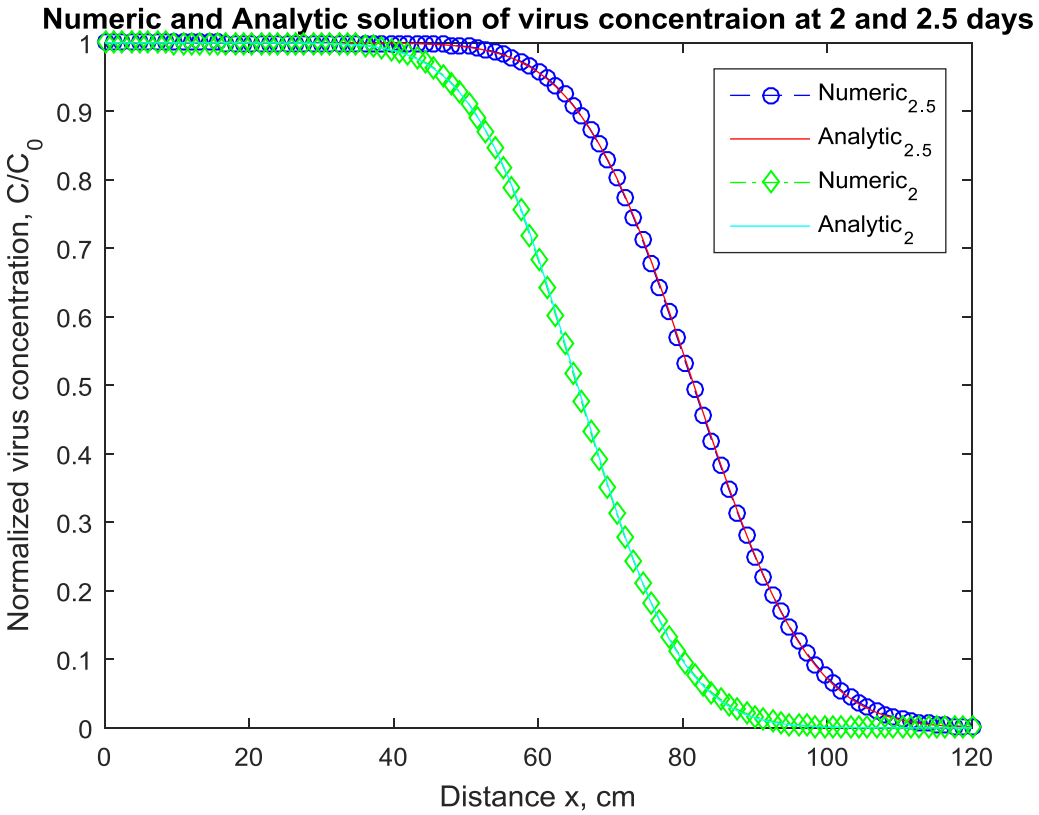

Figure 4.1: Comparison of analytical and numerical solution of 1d virus transport equation after 2 and 2.5 days

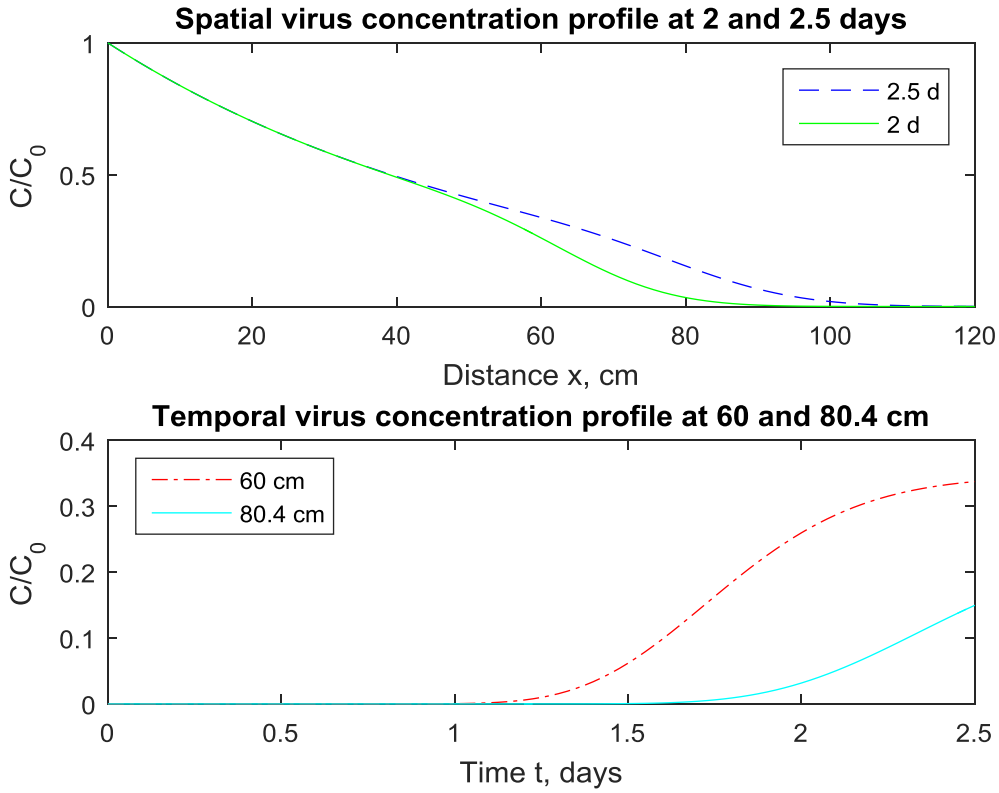

Figure 4.2: Virus concentration profiles considering inactivation constants $\lambda = \lambda^* = 0.58/day$

4.5.2 Parameter Estimation

For parameter estimation, we consider the same model used for validation purpose with non-zero inactivation constants. However, we consider that observations are taken at two discrete locations, at 11 cm and 22 cm, respectively and the samples are taken at the discrete instances as well, namely 0.5, 1.0, 1.5, 2.0 and 2.5 days. The actual parameters used for the simulation to generate the observed data at the known locations and times are, $k_d = 0.02\ ml/g$, $D = 34\ m^2/s$, $\lambda = 0.58/day$ and $\lambda^* = 0.50/day$. It is to be noted that as reported by Ratha et al. (2009), four parameters cannot be estimated simultaneously using classical gradient-based method.

For this study, ooDACE package by Couckuyt et al. (2012, 2014) is used for ordinary kriging. Built-in default optimization modules are used for estimation of kriging meta-model parameters. Initially, 20 design points are created following our DoE algorithm discussed in Chapter 2 and in Shamsuzzaman et al. (2015) considering the upper and lower limits of the parameters as $\{0.04, 40, 0.6, 0.55\}$ and $\{0.01, 30, 0.5, 0.4\}$ for $\{k_d, D, \lambda, \lambda^*\}$, respectively.

As mentioned before, the observed values are 2 vectors corresponding to 2 observation locations and the length of each vector depends on the sampling frequency. The simulation results are obtained at the same locations and times. The error is calculated by finding the difference between the observed and simulated values using four distance metrics, namely, ℓ_1 (Manhattan distance), ℓ_2 (Euclidean distance), ℓ_∞ (Chebyshev distance) and ℓ_h (Hausdorff distance). The model is developed using resulting errors and the objective is to find the minimum error for the estimated parameters.

The initial model is updated by adding a new point found using EI criterion and the process is repeated until 40 simulations (10x4) in total are carried out. Once the final model is achieved after the allotted simulation budget (40 runs), the optimal parameters are found finding the minimum from the resulting final GP model. However, being stochastic in nature, different runs might result in varying outcomes, if a fixed seed is not used for the random number generator.

Therefore, 50 experiments were carried out for each distance metric and the error is calculated by subtracting the corresponding known parameter values. The root mean square errors for each parameter are shown in Table 4.1(corresponding standard deviations are shown in the brackets). It is evident that on average, the estimated parameter values are

satisfactorily identified using all 4 distance measures as the calculated error is not very large. For better understanding and fair comparison, the errors are normalized with their known values. The average normalized errors are shown in the last column in Table 4.1. Performance of ℓ_2 and ℓ_∞ norms are comparable and ℓ_∞ norm performed best in terms of average normalized error. Although, numerical value of error due to k_d terms looks small, it has the highest impact on calculated average normalized error followed by λ^*.

Table 4.1: Error statistics for estimated parameters using different distance measures

Distance metric	k_d (0.02)	D (34.0)	λ (0.58)	λ^* (0.50)	Average of normalized errors
ℓ_1	0.004 (0.002)	2.771 (1.602)	0.01 (0.004)	0.043 (0.018)	0.096
ℓ_2	0.003 (0.126)	0.8 (0.024)	0.007 (0.002)	0.062 (0.024)	0.072
ℓ_∞	0.002 (8.19e-4)	0.839 (0.339)	0.003 (0.002)	0.064 (0.06)	0.066
ℓ_h	0.005 (0.005)	3.304 (3.237)	0.016 (0.008)	0.055 (0.053)	0.122

In Figure 4.3, boxplots for the estimated parameters are shown for the same experiment. Estimated average values for different parameters using ℓ_∞ norm are seen to be closest to the real values. The performance of estimation using ℓ_2 norm is similar to that of using ℓ_∞ norm. However, ℓ_∞ norm shows greater spread (standard deviation), especially for λ and λ^*, compared to that of ℓ_2 norm.

In this virus transport experiment, only 5 temporal data for each location are used to calculate the errors using different distance metric. Sparse data might lose some important information relevant to parameter estimation. Therefore, we doubled the observation frequency and carried out the same experiment to test the impact of adding more observations on parameter estimation using GP-based metamodel.

In Table 4.2, the calculated errors are shown and the boxplot for the estimated parameters are shown in Figure 4.4 after doubling observation frequency. It is evident from both the table and figure that increasing the observation frequency improves the quality of parameter estimation.

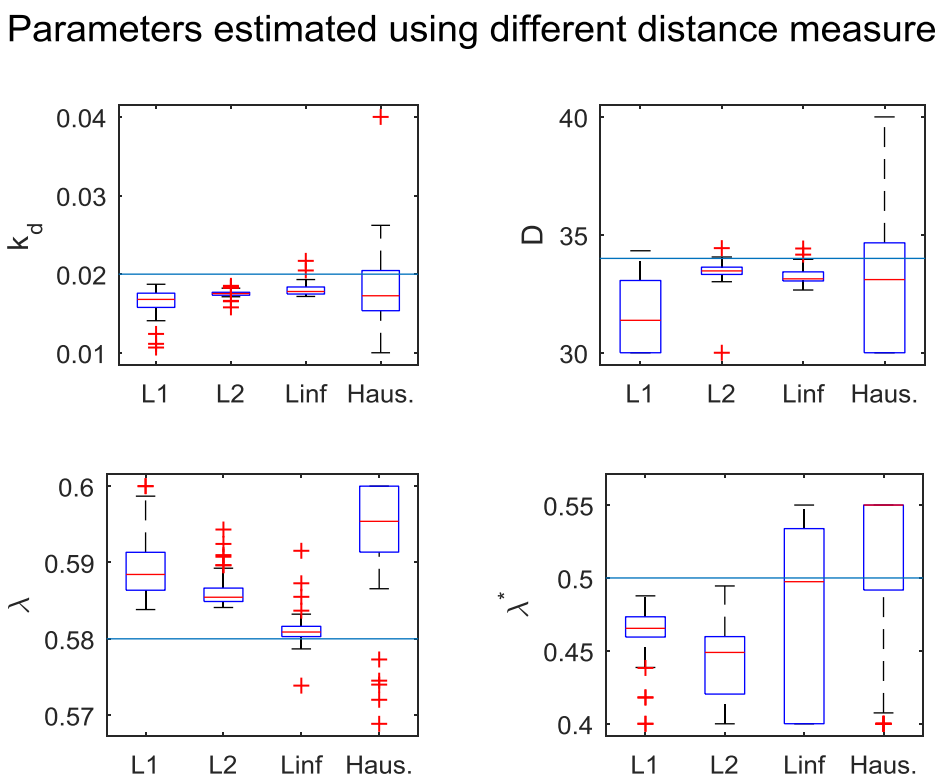

Figure 4.3: Estimated parameters using low density simulation

Table 4.2: Estimated parameters when observation frequency doubled

Distance metric	k_d (0.02)	D (34.0)	λ (0.58)	λ^* (0.50)	Average of normalized errors
ℓ_1	0.001 (0.0012)	1.032 (0.968)	0.006 (0.003)	0.079 (0.038)	0.066
ℓ_2	0.002 (8.68e-4)	0.448 (0.333)	0.009 (0.003)	0.06 (0.028)	0.067
ℓ_∞	0.5.6e-4 (1.97e-4)	0.569 (0.176)	0.002 (0.002)	0.057 (0.052)	0.041
ℓ_h	0.005 (0.005)	3.26 (3.242)	0.019 (0.015)	0.051 (0.049)	0.121

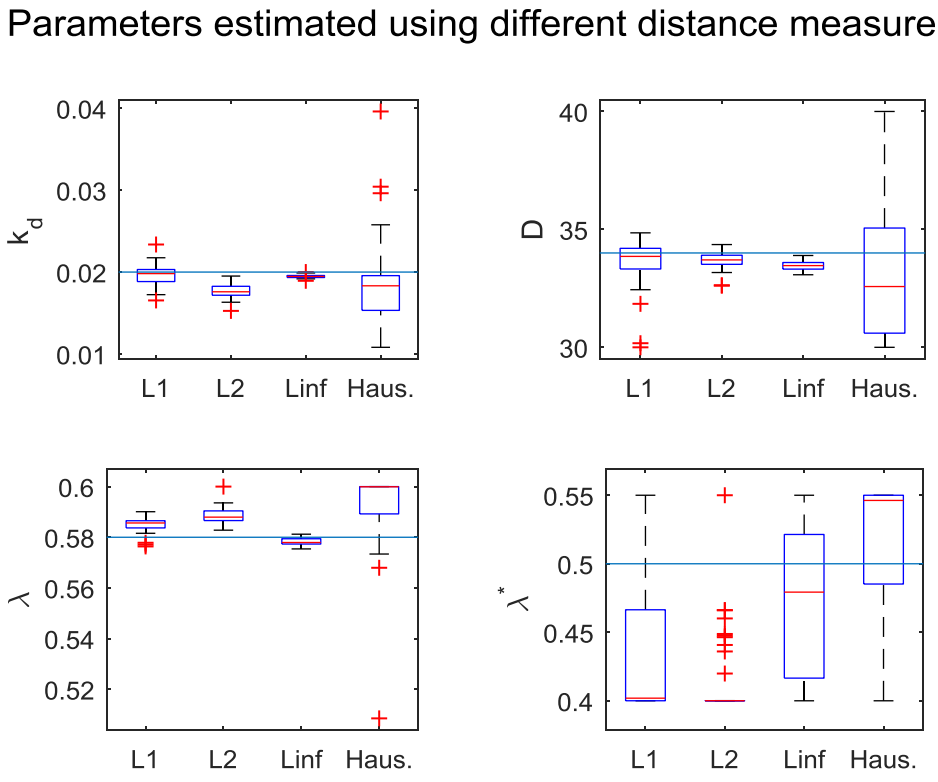

Figure 4.4: Boxplot of estimated parameters using different distance norms

To accommodate kriging meta-model, temporal information is lost due to aggregation of results (transforming from vector to scalar metric), therefore, sensitivity of the parameters over time horizon cannot be depicted using this model. To overcome this, either spatio-temporal model should be developed or different kriging models for each time-step with fitted hyper-parameters for that instance should be considered.

4.6 CONCLUSIONS

Possibility and efficacy of parameters estimation of virus transport phenomenon in groundwater flow using GP models with basic kernels is carried out. Multiple temporal output from the developed model are converted to a scalar quantity as basic GP models are designed in this manner. Different distance metrics were tested. The Chebyshev distance measure (ℓ_∞-norm) and the Euclidean distance measure (ℓ_2-norm) provide balanced results considering all 4 parameters. The obtained results are encouraging and conform to the results published by other researchers. However, temporal information is lost and sensitivity of parameters over the time horizon cannot be done using this method. Modification to the basic GP structure is recommended for experiments those require vector output from GP models.

CHAPTER 5 COMPOSITE KERNEL GAUSSIAN PROCESS

In the previous chapter, we introduced kriging or GP for parameters estimation. In this chapter, GP models are elaborated further and model selection criteria is incorporated for efficient proxy design. It is well known that incorporating extra terms (e.g., more power in polynomial regression) in a model is always prone to overfitting, i.e., provides low error with the training data while giving high error in actual or test data. In this chapter, an appropriate kernel selection method is proposed to overcome this limitation thereby avoiding overfitting and being capable of capturing underlying signals. In general, it is very difficult, if not impossible to guess the underlying signal that truly describes the real data. Therefore, intuition about the underlying process, like smoothness, periodicity or other trends might help to start with a set of base processes and combining them, resulting in a composite process that is closer to the underlying signal. As a result, better models can be obtained for further analysis.

5.1 INTRODUCTION

A model is defined by its structure and associated parameters. Hence, model selection is related to finding the structure of the model and associated parameters given the training data. A GP is fully defined by the mean and covariance functions (Rasmussen & Williams, 2006). Therefore, selection of a GP model translates to finding the proper structure of mean and covariance functions and their associated hyper-parameters. Usually, a 0 (zero)-mean GPs are used without loss of generality and selecting a GP model becomes the problem of finding proper covariance function and its hyper-parameters.

A simple model can explain relationships well only for small data sets while complex models can explain large data sets but somewhat poorly, so a proper model is selected that is neither too simple nor too complex (Rasmussen & Williams, 2006, p. 110). In the following section, GP models are elaborated further followed by the model selection procedure.

5.2 GAUSSIAN PROCESS

A stochastic process is generalization of probability distribution. A GP can be defined as a distribution over functions and inference taking place directly in the function-space (Rasmussen & Williams, 2006, p. 7). It is a collection of random variables,$x_1, x_2, \cdots$, any number of which is Gaussian distributed. In GP, instead of the distribution of the random variables, the distribution over the function $f(x)$ is considered. The covariance matrix required for the multi-variate Gaussian is obtained through the kernel function, k that must be symmetric and positive definite. The kernel function computes the covariance between two function values $f(x_i)$ and $f(x_j)$ by just looking at the corresponding inputs x_i and x_j.

$$\Sigma_{ff}^{(i,j)} = Cov\big[f(x_i), f\big(x_j\big)\big] = k(x_i, x_j) \tag{5.1}$$

Using the definition of GP it can be written in the following form:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x')) \tag{5.2}$$

Where, mean $m(x) = \mathbb{E}[f(x)]$ and covariance is defined in Equation 5.1.

It implies that any collection of function values, evaluated at any points, is jointly Gaussian:

$$[f(x_1), \cdots, f(x_N)] \sim \mathcal{N}(\boldsymbol{\mu}, K) \tag{5.3}$$

With mean vector $\boldsymbol{\mu}_i = \mathbb{E}[f(x_i)]$ and covariance $K_{ij} = k(x_i, x_j)$.

It is evident that GP can be fully specified using the mean function m and the kernel/covariance function k. The kernel incorporates structural information of the latent function f, like, smoothness and differentiability. For example, the popular Gaussian or *squared exponential* (SE) covariance function,

$$k_{Gauss}(x_i, x_j) = \theta_1^2 e^{-\frac{\frac{1}{2}(x_i - x_j)^T (x_i - x_j)}{\theta_2^2}} \tag{5.4}$$

θ_1 defines the magnitude of the latent function and controls the variability from the mean while $\theta_2$denotes length scale – that governs significant changes of the function due to change in length, i.e., θ_2 determines how wiggly the function is. The covariance function also express that the values at nearby locations are more correlated than the values at locations far away from each other. Interested readers can consult any topical book for GP, like Rasmussen & Williams (2006).

The kernel trick in Equation 5.1 yields predictive distribution resulting GP regression as Bayesian inference problem. If the training data $\boldsymbol{\mathcal{D}}(\boldsymbol{X}, \boldsymbol{y})$ is available, the prior is a GP $\mathcal{GP}(m, k)$ with mean m and kernel k. The posterior is also a GP and can be obtained using Bayes' theorem.

$$\mathcal{GP}(m_{post}, k_{post}) = p(f|\boldsymbol{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|f, \boldsymbol{X})p(f)}{p(y|\boldsymbol{X})} \tag{5.5}$$

Where the *likelihood* or noise model is $p(\boldsymbol{y}|f,\boldsymbol{X}) = \mathcal{N}(f(\boldsymbol{X}),\sigma_\epsilon^2\boldsymbol{I})$, $\epsilon \sim \mathcal{N}(0,\sigma_\epsilon^2)$, the marginal likelihood or evidence is $p(y|\boldsymbol{X}) = \int p(\boldsymbol{y}|f,\boldsymbol{X})p(f)df$. The predictive distribution of GP $p(f_*|\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_*)$ at test inputs $\boldsymbol{x}_*$ yields:

$$p(f_*|\boldsymbol{X},\boldsymbol{y},x_*) = \mathcal{N}(\mathbb{E}[f_*],\mathbb{V}[f_*]) \tag{5.6}$$

$$\mathbb{E}[f_*|\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_*] = m_{post}(x_*) = m(x_*) + k(\boldsymbol{X},x_*)^T(\boldsymbol{K}+\sigma_\epsilon^2\boldsymbol{I})^{-1}(\boldsymbol{y}-m(x_*)) \tag{5.7}$$

$$\mathbb{V}[f_*|\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_*] = k_{post}(x_*,x_*) = k(x_*,x_*) - k(\boldsymbol{X},x_*)^T(\boldsymbol{K}+\sigma_\epsilon^2\boldsymbol{I})^{-\mathbf{1}}(k(\boldsymbol{X},x_*) \tag{5.8}$$

In Figure 5.1(a) different functions are sampled from 0 mean GP prior. The probability mass around the mean is shown shaded gray. The darker shade implies more probability of finding a sample and vice versa. The effect of adding known data points on posterior distributions reflected in Figures 5.1(b-c) for incorporating 1-3 data points respectively. As the data points are added, the uncertainties shrink around those points. The remaining uncertainty around the known data points is due to the noisy measurements.

If the likelihood $p(\boldsymbol{y}|\boldsymbol{f})$ is not Gaussian, the predictive distribution cannot be obtained using an analytic form and a sampling method is used instead, like Monte Carlo method. However, Gaussian likelihood is considered in this study where analytic form is available and no approximation was considered Rasmussen & Williams(2006).

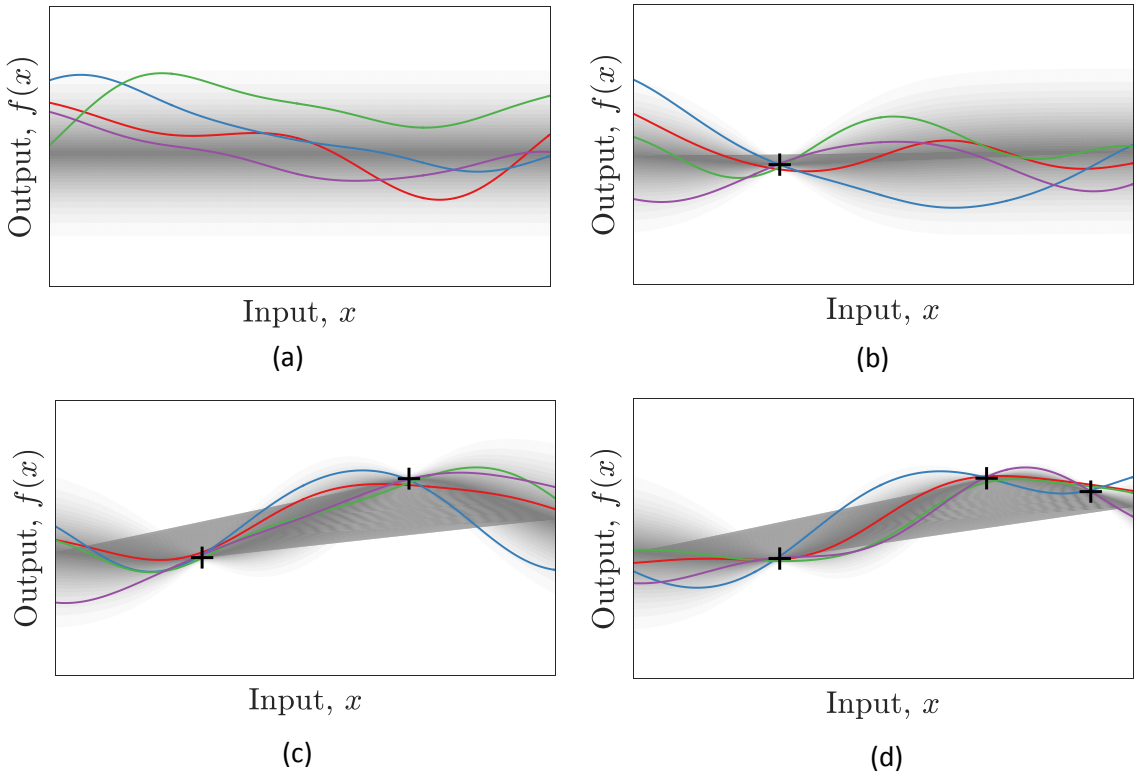

Figure 5-*1*: Sample functions from a) GP prior, b), c) and d) from posterior, given 1, 2 and 3 data points with an SE kernel. Probability mass around the mean is shown with gray shade and the data are presented by '+' symbol

5.2.1 Model Selection

In Chapter 4, the hyper-parameters of the GP model were selected by maximizing the marginal likelihood or evidence using Equation 4.7. Logarithm of marginal likelihood incorporates both data fit and model complexity terms. While, the model fit term ($\frac{1}{2}\boldsymbol{y}^T(K_\theta + \sigma^2 I)^{-\mathbf{1}}\boldsymbol{y}$) incorporates data, complexity penalty ($\frac{1}{2}\log|K_\theta + \sigma^2 I|$) relies on the amount of data the model can accommodate. For example, complexity penalty depends on the length scale hyper-parameter of GP model with SE kernel. The trade-off between the data-fit and model complexity in GP is automatic if marginal likelihood is used Rasmussen & Williams (2006, p. 110). Therefore, optimizing the marginal likelihood for hyper-

parameter estimation is used in most cases, including this study. It is to be noted that marginal likelihoods for hyper-parameters suffer from multiple local optima but do not pose devastating problems in practical applications Rasmussen & Williams (2006, p. 115).

However, a GP model can use any valid kernel out of many available alternatives but not all the kernels capture the underlying process to be modeled. Therefore, there is a need to find a suitable kernel for modeling using GPs. As a result, researchers investigated different approaches to find proper kernels for different reasons, like, pattern discovery and extrapolation Wilson & Adams (2013), linear system identification Pillonetto & Nicolao (2011) .

If several alternative models M_i are available with corresponding mean function m_i and kernel k_i, finding a model can be carried out using a similar approach such as hyper-parameter selection. So, if a prior $p(M_i)$ is assigned to the ith model, the posterior model probability can be written as

$$p(M_i|\boldsymbol{X}, \boldsymbol{y}) = \frac{p(M_i)p(y|X, M_i)}{p(y|X)} \tag{5.9}$$

A model that maximizes the marginal likelihood can be selected as the preferred model. Various proposals for model selection from Bayesian-theoretic perspective are discussed in Kadane & Lazar (2004). In Section 5.4 cross-validation will be discussed as an alternative for model selection.

Using multiple kernels instead of a single one is useful for support vector machines Gönen & Alpaydın (2011). Being a kernel method, GP is expected to behave in a similar manner. Linear combination of complex Gaussian kernels is more reasonable too Gönen &

Alpaydın (2011). In the following section, properties of kernels and different kinds of GP kernels are described.

5.3 KERNELS FOR GP

Covariance function or kernel k plays a very important role in GP modeling. In fact, covariance functions control the power of the GP models. *Kernel* is a general term used for a function that maps any pair of inputs (could be vectors) into $\mathbb{R}$ (a scalar quantity). The covariance function is a kernel that is used in GP models, therefore, covariance function and kernel are used interchangeably in this thesis. A necessary and sufficient condition for a valid covariance function is that the covariance matrix must be positive semi-definite Bishop (2006, p. 295). A new valid covariance function can be constructed using the existing kernels following the properties of covariance function as described in Bishop (2006, p. 296). Therefore, more complex kernels can be constructed using some available basis kernels. In Table 5.1, some commonly used basis kernels are mentioned mainly taken from Chapter 4 of the GP book and its companion gpml-package by Rasmussen & Williams (2006).

Table 5.*1*: Example of some basis covariance functions

Covariance function	**Expression**
Constant	a_0
Linear	$x.x'$
Polynomial	$(x.x' + a_0)^p$
γ-Exponential	$\exp\left(-\left(\frac{\lvert x - x' \rvert}{l}\right)^{\gamma}\right)$
Matérn	$\frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}\lvert x - x' \rvert}{l}\right)^{\nu} K_{\nu}\left(\frac{\sqrt{2\nu}\lvert x - x' \rvert}{l}\right)$

Covariance function	**Expression**
Ornstein-Uhlenbeck	$\exp\left(-\frac{\lvert x-x'\rvert}{l}\right)$
Rational Quadratic	$\left(1+\frac{\lvert x-x'\rvert^2}{2\alpha l^2}\right)^{-\alpha}$
Periodic	$\exp\left(-\frac{2\sin^2\left(\frac{x-x'}{2}\right)}{l^2}\right)$
Neural network	$\sin^{-1}\left(\frac{2\bar{x}^T\sum\bar{x}'}{\sqrt{(1+2\bar{x}^T\sum\bar{x})(1+2\bar{x}'^T\sum\bar{x}')}}\right)$

5.4 CROSS-VALIDATION

Cross-validation (CV) is one of the most commonly used methods for evaluating predictive performance of a model due to its simplicity and its (apparent) universality, Zhang & Yang (2015); Arlot & Celisse (2010). The main idea of cross-validation is to split the whole available dataset into *training* and *validation* sets where the former set is used to identify the hyper-parameters of the model and the later set is used for the calculation of error predicted by models developed using the training sample. As the training and testing samples are independent, CV can avoid overfitting. A detailed discussion on CV can be found in Arlot & Celisse (2010).

The *generalization error* of a GP model is defined as

$$\epsilon = \mathbb{E}[\left(f(\boldsymbol{X}) - \hat{f}(\boldsymbol{X})\right)^2 \tag{5.10}$$

Where, f and $\hat{f}$ are the original and predicted values using the GP model, respectively.

In case of small sample (which is the case of this study), it is difficult to estimate the generalization error. Therefore, out-of-sample cross validation is more appropriate for the assessment of a GP model. However, for a small number of experiments, it is difficult (unreasonable) to divide the data-set into training and test set. Therefore, only one sample is left for testing to measure the error resulting in leave-one-out cross-validation technique (LOOCV). To calculate LOOCV error, each data point is successively kept aside from the sample for validation. A model is developed using the rest of the training sample and prediction error is calculated using the left-out validation sample. The process is repeated for all the sample points and the generalization error is estimated by averaging leave-one-out error. Therefore, if there are N samples, Prediction error at $\boldsymbol{x}_i$ is:

$$\Delta_i = f(\boldsymbol{x}_i) - \hat{f}_{-i}(\boldsymbol{x}_i) \tag{5.11}$$

Predicted residual sum of squares (PRESS) is a credible criterion based on the leave-one-out prediction error and can be defined as Allen (1974):

$$PRESS = \sum_{i=1}^{N} \Delta_i^2 \tag{5.12}$$

Performance of CV for model selection is comparable to other information-criteria based methods. CV was found asymptotically equivalent to Akaike information criterion (AIC) Peng et al. (2013). CV on the original models behaves somewhere between AIC and Bayesian information criterion (BIC) and LOOCV typically performs optimally for the infinite-dimensional case Zhang & Yang (2015).

However, PRESS in Equation 5.12 depends on the data size and therefore, comparing the performance of different models for varying data size is not possible. One can use root mean square PRESS and the corresponding statistics to overcome this problem. The leave-

one-out prediction error statistics can be considered as an adjusted sum of square error and can be defined as:

$$\epsilon_{LOOCV} = \sqrt{(\frac{1}{n} PRESS)} = \sqrt{(\frac{1}{n}\sum_{i=1}^{N} \Delta_i^2)} \tag{5.13}$$

$$R^2_{LOOCV} = 1 - \frac{PRESS}{\sum(f(\boldsymbol{x}_i) - \bar{f}_{-i})^2} \tag{5.14}$$

The $\bar{f}_{-i}$ indicates the average of the leave-one-out (LOO) response in Equation 5.14 and the denominator can be considered as PRESS of y's mean value. The R^2_{LOOCV}-statistics in Equation 5.14 adjusts the estimate of how much variation the model can explain. Therefore, the closer R^2_{LOOCV} is to 1, the better the generalization capacity of the GP model. If the model is very poor, the R^2-statistics can be negative.

In this study, Equation 5.14 is used for selecting a suitable covariance function, basis or composite in nature. Please note that a number of covariance functions might have very similar performance. We selected the one with the least number of hyper-parameters. One can go for model averaging but this is beyond the scope of this thesis. In the following sections, an algorithm is proposed to find the optimal GP kernel based on Equation 5.14followed by the applications of resulting GP models in Sections 5.6 and 5.7.

5.5 A New Method for Finding Proper Kernel in GP

Combining kernels in a different manner to find a new optimal kernel for GP is a combinatorial optimization problem. Combining more than one valid kernel in different ways can produce new valid kernels as mentioned in (Bishop, 2006, p. 296). For our proposed algorithm based on QPSO described in Chapter 3, three basic operations are

carried out, namely, *add*, *multiply*, and *swap*. Deleting a part from a composite kernel is like swapping operation with null. So, if k_1, k_2, k_3 are valid kernels then a new valid kernel k_n can be obtained using $k_n \leftarrow k_1 + k_2$ using *add* operator and $k_n \leftarrow k_1 \times k_2 = k_1 k_2$ using *multiply* operator. However, from the properties of the kernels we know that $(k_1 + k_2)k_3 = k_1 k_3 + k_2 k_3$. In the proposed algorithm, the composite kernels are always expressed in the later form or sum of product form of the basis kernels. Moreover, two kernels are considered the same if $k_1 k_2 = k_2 k_1$. The proposed algorithm is shown in Figure 5.2.

5.6 CASE STUDY 1: REVISITING VIRUS TRANSPORT PROBLEM

In the previous chapter parameters were estimated using sequential GP. However, the kernel was arbitrarily chosen, despite its promising result. Following the outline discussed in previous sections, the LOOCV errors and the corresponding R^2_{LOOCV} values using Equations 5.13 and 5.14, respectively are calculated for the initial DoE data of virus transport problem. In Table 5.2, ϵ_{LOOCV} and R^2_{LOOCV} values for different basis kernels are shown. Among the simple kernels considered, neural network (NN) kernel performs best. Surprisingly, performance of linear kernel is next to NN. Isotropic kernels in general could not capture the underlying process and should not be used as a kernel to model the GP proxy for virus transport problem at hand.

Our next goal is to find the best GP kernels using our proposed algorithm. Our algorithm provided the composite kernel composed of multiplication of NN and Linear kernel as the best. To verify the accuracy of our algorithm, performance of some of the composite

kernels were calculated and the results are shown in Table 5.3. It is clear from the results that composite kernels perform better, and our algorithm could find a better kernel for

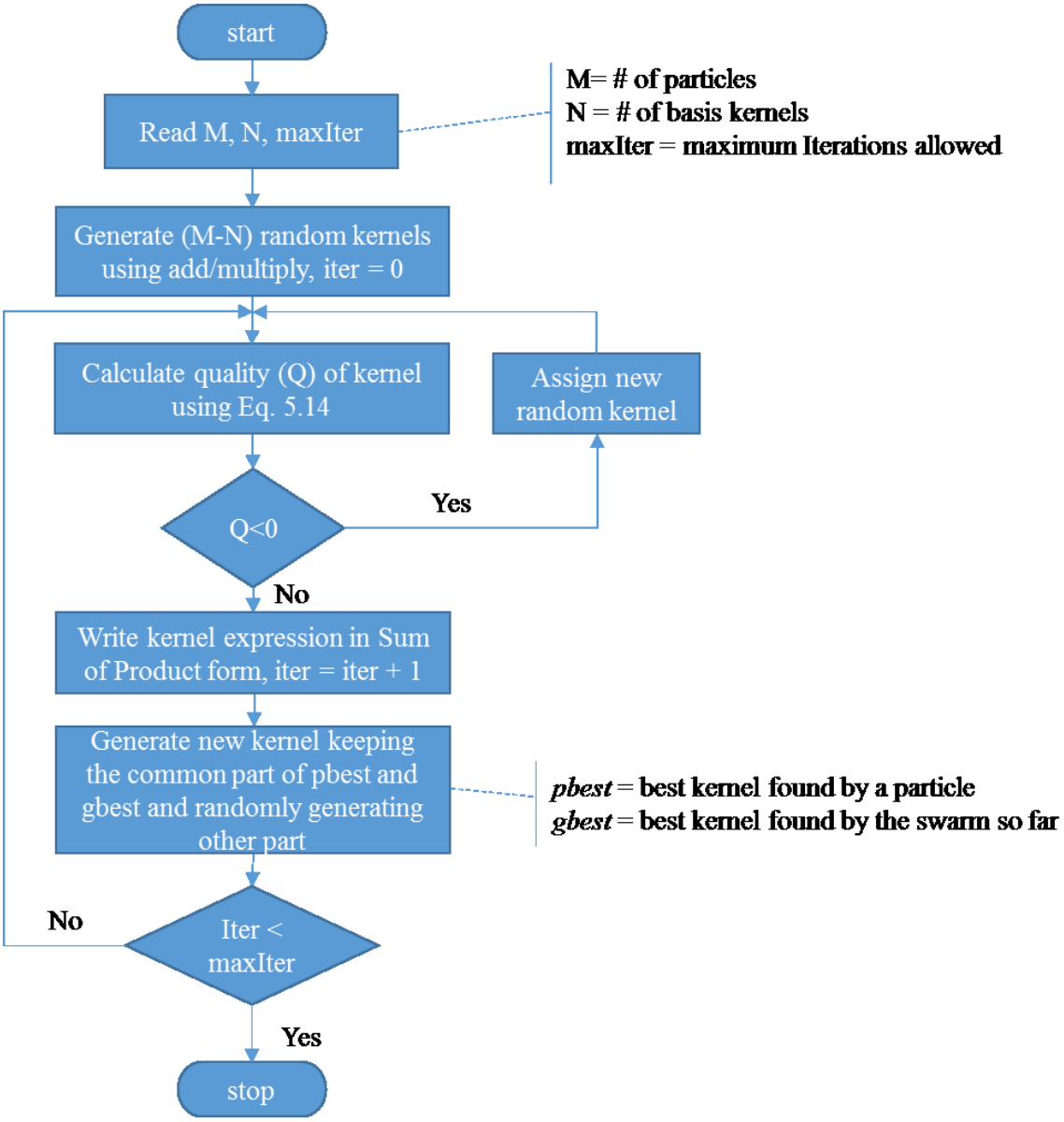

Figure 5.2: Flowchart to generate efficient composite kernel

this particular problem. Some interesting observations can be drawn from the data shown in Table 5.3. For example, simple isotropic SE kernels were very poor performing but adding these kernels for each variable separately shows better performance. On the other

hand simple linear kernels performed better as shown in Table 5.2, but multiplication of 2 linear kernels results in poor performance. This happens as the underlying processes have linear strong trends but higher order terms in the kernel affect the relation negatively. Performance of different composite kernels varies a lot and therefore a kernel of GP modeling should be selected judiciously.

Table 5.2: LOOCV error and related R^2_{LOOCV} values of different simple kernels for virus transport problem

Kernel Name	ϵ_{LOOCV}	R^2_{LOOCV}
Isotropic SE	0.0011	-0.0283
SE ARD	0.0006	0.3945
Linear	0.0004	0.5949
Isotropic Matern, 3	0.0009	0.1167
Matren ARD, 5	0.0007	0.3247
Isotropic RQ	0.0011	-0.0371
Neural Net	0.0004	0.6630

Table 5.3: LOOCV error and related R^2_{LOOCV} values of different composite kernels for virus transport problem

Composite Kernel Name	ϵ_{LOOCV} $(\times 10^{-3})$	R^2_{LOOCV}
SE ADD	0.6371	0.3967
Linear × Linear	0.7785	0.2628
Linear + SE_{ard}	0.4233	0.5991
Linear × SE_{ard}	0.6278	0.4055
Linear + Mattern 5	0.4664	0.5583
Linear × Mattern5	0.7353	0.3037
Linear + RQ_{ard}	0.4554	0.5688
Linear× RQ_{ard}	0.7283	0.3103
Linear + NN	0.4682	0.5567
Linear × NN	0.3354	0.6824

Following the kernel selection metric, a composite kernel based on multiplication of a linear kernel and a neural network kernel is used for parameter estmation of virus transport in groundwater flow. Other than replacing the kernel, the same procedure is used as in Chapter 4. However, this time the gpml toolbox is used (Rasmussen & Williams, 2006). In Table 5.4, the estimated parameters are shown. As expected, the quality of estimation improved both in terms of average value and spread (standard deviation) of the results. In Table 5.4, estimated parameters are shown using the composite kernel (Linear × NN).

Table 5.4: Estimated parameters using composite kernel

	$\mathbf{k_d}$ (0.02)	D (34.0)	$\boldsymbol{\lambda}$ (0.58)	$\boldsymbol{\lambda}^*$ (0.50)
L1 norm	0.0199 (0.0001)	33.92 (0.1182)	0.5812 (0.008)	0.4701 (0.0172)
L2 norm	0.0198 (0.0009)	34.1692 (0.1321)	0.5798 (0.0030)	0.4933 (0.0279)
L inf	0.0197 (0.0002)	33.7591 (0.157)	0.5872 (0.0021)	0.4884 (0.0601)
Hausdorff	0.0191 (0.0055)	33.8152 (1.4212)	0.5897 (0.0091)	0.5103 (0.0512)

5.7 CASE STUDY 2: HISTORY MATCHING OF PUNQ-S3

In this section, the porosity of layer 5 of PUNQ-S3 reservoir model is predicated using GP proxy model. Production forecast uncertainty quantification (PUNQ)-S3 model was developed from a real field data, extensively studied for reservoir engineering cases and is qualified as a small-size industrial reservoir engineering model (Department of Earth Science and Engineering, 2015). The model is obtained as Eclipse® datafile containing $19 \times 28 \times 5$ gridblocks, of which 1761 blocks are active from (Department of Earth

Science and Engineering, 2015). The obtained model is converted to Petrel® model and is shown in Figure 5.3.

In this study, only the porosity of layer 5 is considered. Therefore, in our GP model about 400 (exactly, 396 cells are defined and 136 cells are inactive) independent variables are considered. The porosity at the location of six wells are known. Rest of the 390 values are to be estimated through inference. Original porosity distribution at layer 5 is shown in Figure 5.4.

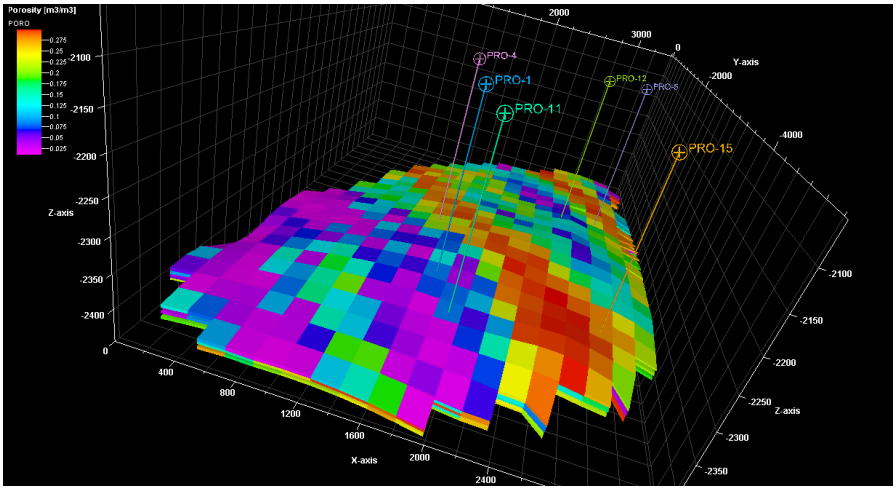

Figure 5.3: PUNQ-S3 Model

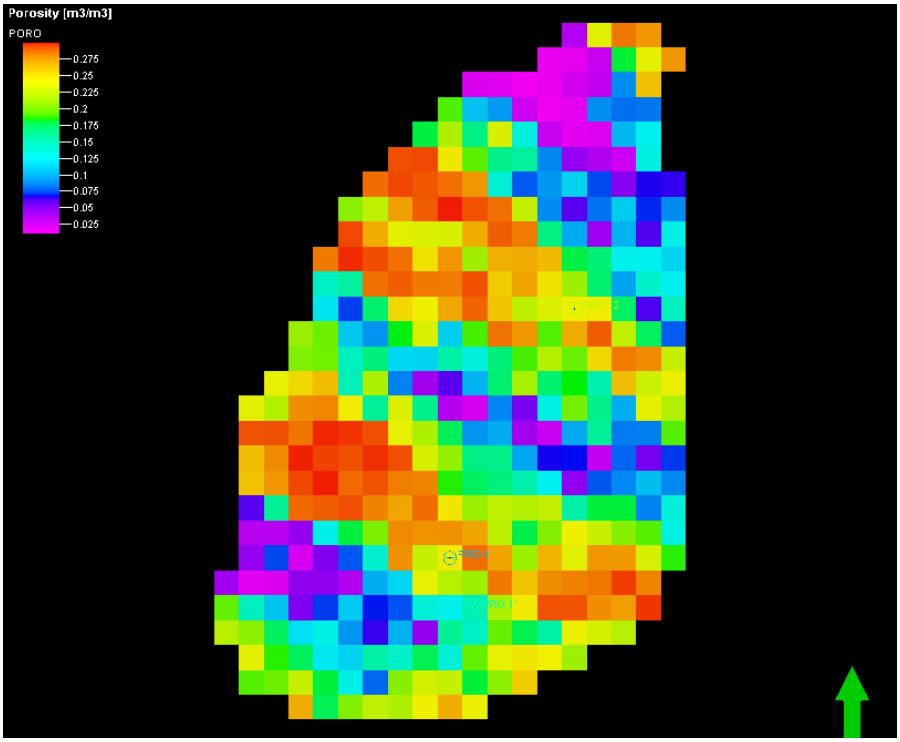

Figure 5.4: Porosity distribution at layer 5 in PUNQ-S3 model

The original porosity in layer 5 of the defined cells has the following statistics are:

Minimum: 0.0108, Maximum: 0.2982, Mean: 0.1791, and Std.: 0.0857.

The locations of 6 wells and the porosity at these locations are shown in Table 5.5.

Table 5.5: Locations of wells and the porosity values at well-locations

Location (grid)			**Location (m)**			**Porosity**
I	**J**	**K**	**x**	**y**	**z**	
17	11	5	2970	-3149	-2366.908	0.078950
15	12	5	2610	-2969	-2361.656	0.096693
9	17	5	1530	-2069	-2391.26	0.240232
10	22	5	1710	-1169	-2407.046	0.286667
17	22	5	2970	-1169	-2397.611	0.050042
11	24	5	1890	-809	-2410.651	0.165366

The corresponding statistics of the known porosity values are:

Minimum: 0.0500, Maximum: 0.2867, Mean: 0.1530, and Std: 0.0947

Estimation of porosity is a two-stage process for this study. In the first phase of the estimation, known porosity values are used to build a GP model to populate the porosity

in rest of the gridcells. Therefore, estimated porosity values with associated uncertainty will be known before using any production data. The kernel based on $Linear_{ard}$ + $Mattern_{ard}$ + Neural Network gives the best kind of result in terms of LOOCV using the known porosity values. The unknown porosities are predicted using the resulting GP model. We calculated the error due to estimation in the first stage and found that maximum prediction error occurs at (2, 6) gridcell. This make sense as we can see from Figure 5.2 that grid (2, 6) is one of the farthest grids from the wells. However, the error statistics (minimum: 0.0014, mean: 0.1068, standard deviation: 0.0846, maximum: 0.4018) suggests that prediction using known porosities at well-locations alone is not sufficient for parameter estimation. Therefore, parameters (in this case, porosity) are to be conditioned on the available production data to predict it more reliably. However, in the latter case, the values predicted in stage 1 are used along with their associated variance. For example, at grid (2,6), the predicted mean porosity is 0.4191 with variance of 0.0011. So, different realization for simulation run will produce a porosity value for this particular grid is normally distributed with mean 0.4191 and standard deviation of 0.0662.

In the second phase, bottom-hole pressure, oil production rate and water cut of each well are considered as dynamic data. Dynamic data of one of the wells are shown in Figure 5.4.

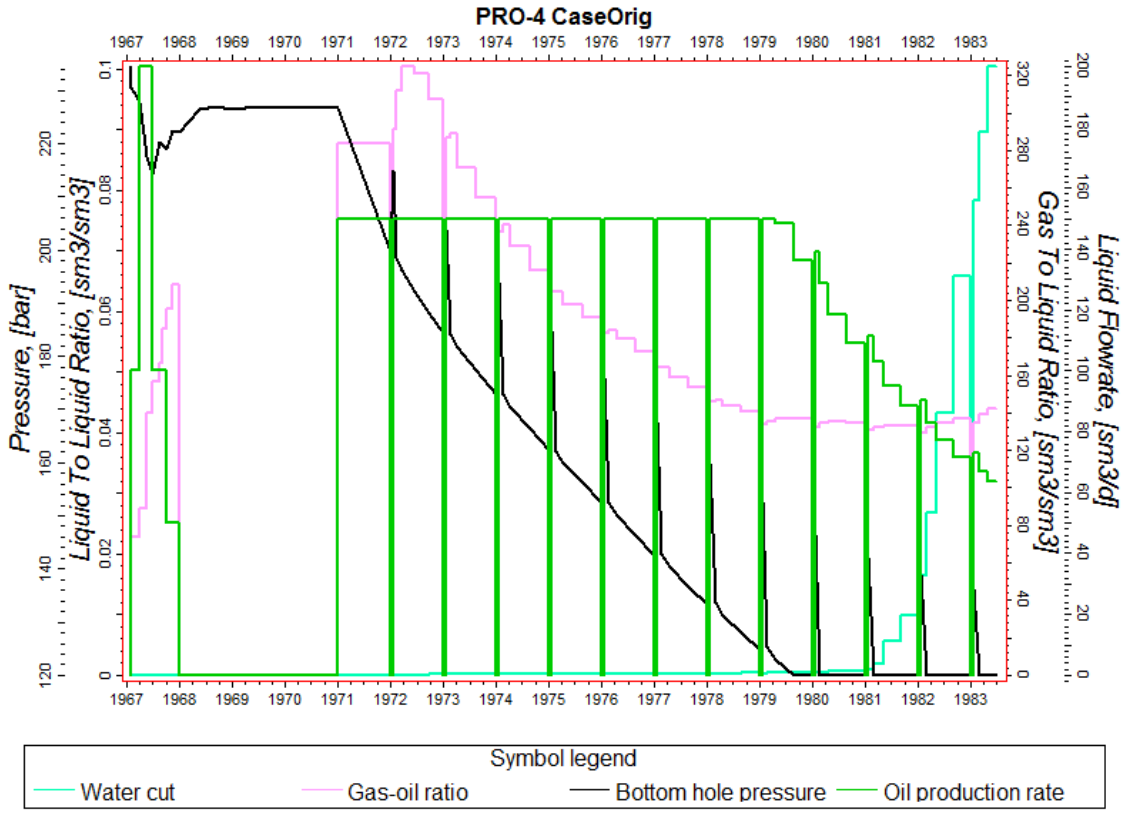

Figure 5.5: Dynamic production data for a typical well (PRO-4) showing water cut (WCT), gas-oil ration (GOR), bottom-hole-pressure (BHP) and oil production rate (OPR). First 3 production data are used for history matching in this study

For different porosity distribution, the error is calculated between the original production data and the simulated data. Forty simulation cases were considered for GP model development and same procedure is followed as described for other cases.

After the second phase of experiment, the dynamic data are compared and is shown in Figure 5.5.

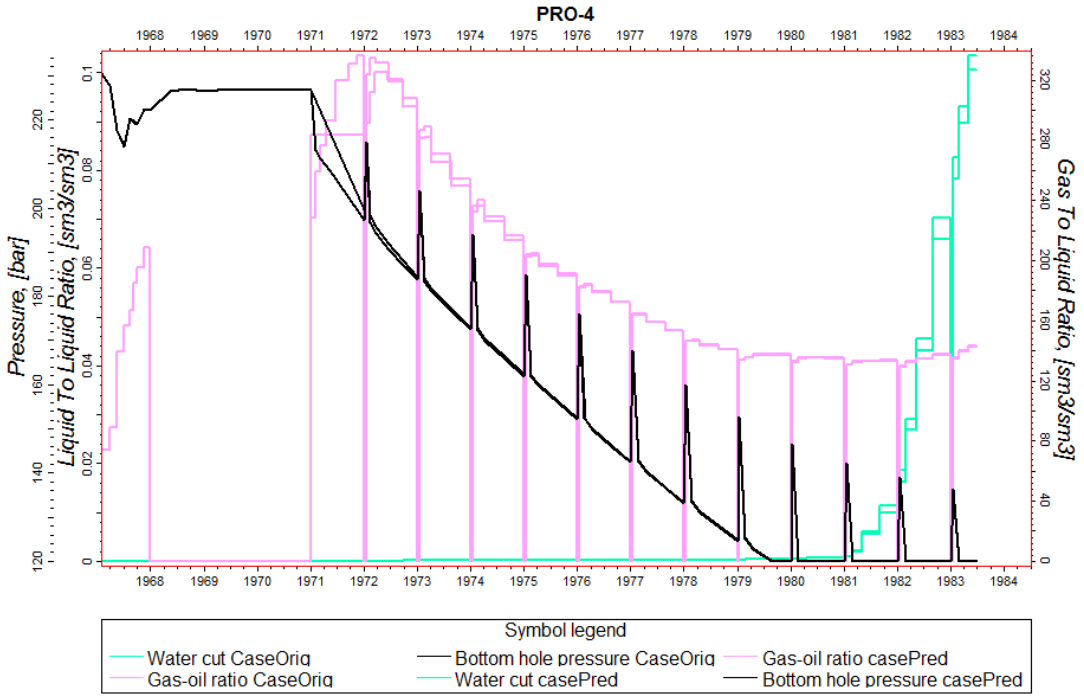

Figure 5.6: Comparison of dynamic data after history match

It is evident from Figure 5.5 that the predicted dynamic data conform to that of the observed data. Therefore, resulting porosity distribution at layer 5 is accepted as the history matched or valid porosity with associated uncertainty.

5.8 Case Study 3: Optimal Well Placement at North Triumph

Determination of optimal locations for injection and production wells in petroleum reservoir has immense industrial interest as it has potentially high economic impact, see for instance Bangerth et al. (2006), Yetenet al (2003), Donget al (2011). For vertical wells, location of a well is determined by its (x, y) coordinates alone. Fluid production from a reservoir can be determined by complex reservoir simulation that needs predefined well locations apart from settings of other parameters. In our case, the reservoir model is developed using Petrel® and the simulations are carried out using

Eclipse® reservoir simulator, both from Schlumberger®. The objective for optimization could be the economic parameter net present value (NPV), cumulative oil/gas production, sweep efficiency or any other metric that represents the performance of a reservoir. In this case, we considered cumulative gas production from the North Triumph reservoir, discussed in the following sections.

5.8.1 North Triumph Reservoir

The North Triumph gas field is located in the Sable Island Delta, Nova Scotia, Canada. It was discovered in 1986 and developed by ExxonMobil. According to ExxonMobil (1999), the North Triumph reservoir is a rollover anticline that is bounded by major listric faults and divided by minor en echelon faults. Closure to the north and west is provided by rollover into the listric normal fault. Closure to the south and east is provided by cross-fault and/or fault smear. Two minor en echelon growth faults partially dissect the reservoir. It is sealed on the top by Logan Canyon Formation shale.

In Figure 5.6, the contour map of the top of the Mississauga Formation, with reservoir extent and faults are shown. The highest elevation is at a depth of 3600 m below sea level in the east. The formation dips to the north, south and west. The free water level is estimated to be 3771 meters below sea level. The field is extends over an area of 19 sq. kilometers.

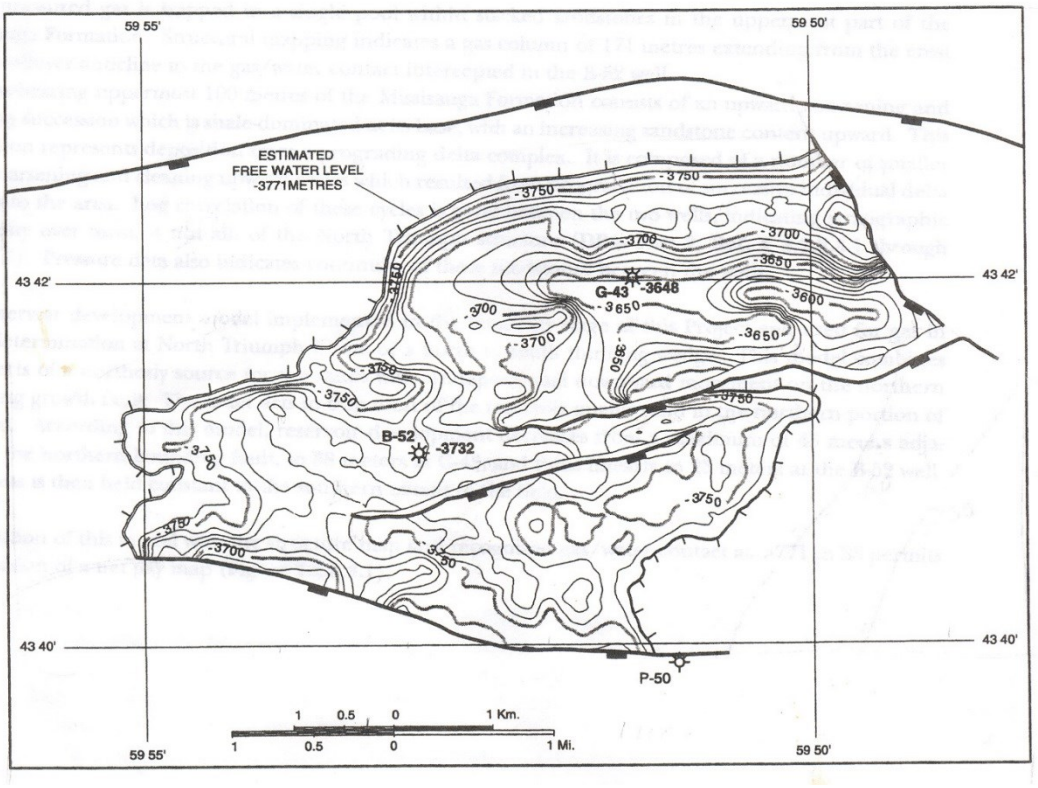

Figure 5.7: Top of the Mississauga Formation of North Triumph (ExxonMobil, 1999)

Very limited geophysical information is publicly available for the North Triumph Field. Two exploratory wells, B-52 and G-43 and two development wells NT-1, and NT-2 were drilled. The logs from the development wells are used to develop the reservoir simulation model. Based on the log data, 8 different layers were selected with the properties shown in Table 5.6. Initial PVT properties of gas and water are shown in Table 5.7. These data were available from Canada-Nova-Scotia Offshore Petroleum Board (CNSOPB) and personal contact with ExxonMobil.

Table 5.6: Summary of layer information of NT at development wells

Layer/ Well ID	**NT-1**				**NT-2**			
	h (m)	**k (mD)**	**ϕ**	**Sg**	**h (m)**	**k (mD)**	**ϕ**	**Sg**
1	3.5	5	0.14	0.50	4.7	5	0.10	0.50
2	3.0	0.1	0.01	0	3.3	0.1	0.01	0
3	7.5	1	0.15	0.70	8.5	10	0.15	0.50
4	2.0	0.1	0.01	0	1.9	0.1	0.01	0
5	13	300	0.20	0.75	16.5	100	0.20	0.75
6	14.5	100	0.15	0.70	16.0	50	0.15	0.60
7	23	100	0.17	0.60	18.9	100	0.18	0.80
8	3.5	1	0.18	0.55	3.8	10	0.15	0.20
Mean	NA	115	0.160	0.62	NA	61	0.158	0.60

Abbreviations: h = layer thickness; NA = not applicable; k = permeability (horizontal); Mean = thickness weighted arithmetic average; ϕ = porosity; Sg = gas saturation

Based on the available data, the top contour map was digitized manually and a reservoir model was developed accordingly using Petrel® in Newman-Bennett et al. (2008). The resulting grid is composed of $101 \times 54 \times 8$ cells and is shown in Figure 5.7 including the production wells.

Table 5.7: Initial fluid properties of NT reservoir

Well ID	**Formation Compressibility (kPa^{-1})**	**Initial Gas Viscosity (μPa.s)**	**Initial Gas FVF (m3/m3)**	**Initial Water Viscosity (mPa.s)**	**Water Compressibility (kPa^{-1})**
NT1	5.8176E-07	25.2416	3.8033E-03	0.22876	4.2535E-07
NT2	5.9273E-07	24.7124	3.8306E-03	0.23186	4.1917E-07

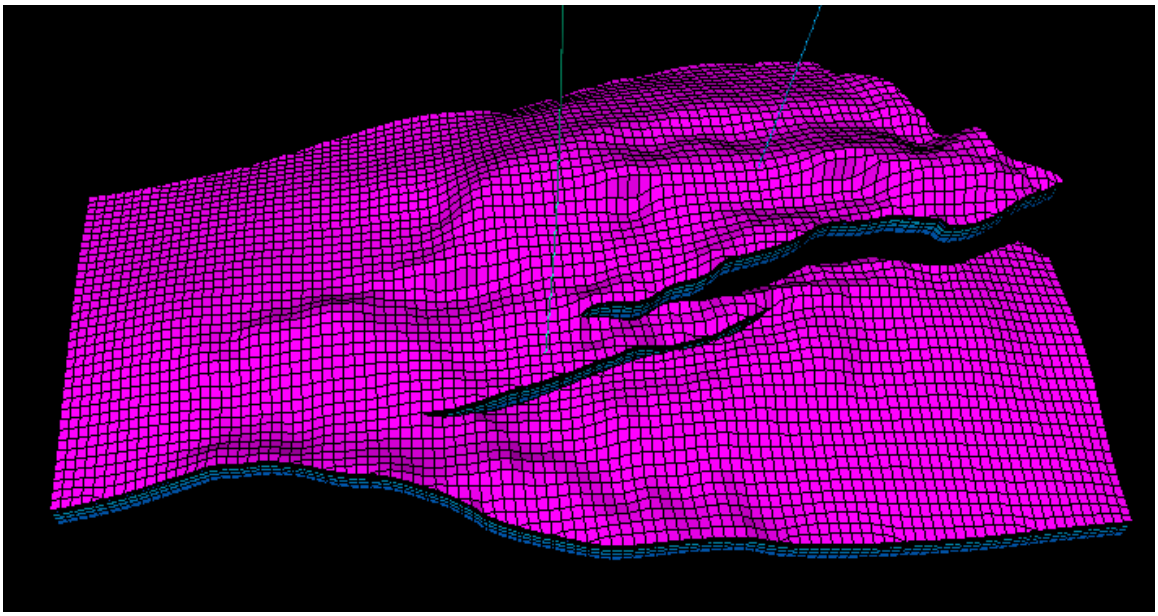

Figure 5.8: The simulation grid for North Triumph gas reservoir

5.8.2 Well Placement Using GP Model

The X-coordinate of the modelled reservoir extends from -736609.05 to -726326.35 and the Y-coordinate from 4838629.59 to 4844163.64 in global coordinate system. Therefore, the location for each well requires 2 variables (x, y) and must satisfy $-736609.05 < x < -726326.35$ and $4838629.59 < \mathrm{y} < 4844163.64$, for X- and Y-coordinates, respectively to be within the reservoir. Moreover, the well should not be located at the faults. As 2 production wells are to be placed for North Triumph reservoir, 4 variables are to be identified in this problem. Roughly, as rule of thumb, $10 \times 4 = 40$ simulation runs are required for a proxy design. We started with 30 initial simulation runs for 10 years production and recorded cumulative gas production of the field. Considering the faults, the reservoir is divided into two parts both in X and Y direction, resulting in 4 quadrants. The first well is randomly chosen and the second well is placed randomly at any location in the remaining 3 quadrants, avoiding the same quadrant. Once the continuous location

values are selected, the closest (I, J) locations in Eclipse® data file is assigned for the wells, accordingly. If the simulation grid does not allow such well locations, a nearby grid is assigned. If that process does not allow the case to run successfully, a new set of random locations are chosen until the case can be run successfully. Again, a GP model is developed using the combination of a linear and a neural network kernels. The GP model is updated using EI in a similar manner, described in Section 5.6.

This above process is repeated 10 times. Each time the porosity, permeability, initial water saturation values are reassigned using Petrel®'s built-in geostatistical module to capture the uncertainty of these variables.

The error is calculated based on the distance between the calculated optimal wells' locations using the GP proxy and actual wells' locations. The statistics of the error for 10 sets of experiment is given below for NT2:

Table 5.8: Statistics of error for optimum well location using GP proxy in North Triumph reservoir

Error	NT2	NT1
RMS	327	250
Maximum	830	690
Minimum	50	45

There are several reasons, why our method could not identify the exact or very close well locations. First, the available data was not sufficient to build a very accurate model. This is true for geological structure and rock properties (like, porosity, permeability, etc.), fluid properties and SCAL data. Moreover, while the well locations calculated using the

GP model is continuous, corresponding wells are assigned at the centre of a grid in Eclipse® data file and the grid size is $\sim 100.45m \times 99.75\ m$ for this study. Therefore, 800m represents roughly 8 grid-blocks. Considering the above conditions, we consider that the optimal location of wells found by the GP model is quite satisfactory.

5.9 CONCLUSIONS

Kernels or covariance functions in Gaussian process play the most important role. Efficiency of a GP model is highly dependent on the structure of a kernel. As new valid kernels can be generated from existing valid kernels through some algebraic operations, a new combinatorial algorithm is proposed using a swarm of basis kernels. The efficiency of the resulting kernel is measured using leave-one-out cross-validation. The operation is expensive but for the small number of simulation runs, the overall cost is justified.

The algorithm was tested against the parameter estimation of groundwater virus transport problem discussed in Chapter 4, history matching of PUNQ-S3 oil reservoir and optimal well placement problem in North Triumph gas reservoir. In the first case, better results were obtained compared to the basic covariance function used in Chapter 4. In history matching, the predicted porosity of a layer were very close to the original and the simulated data matched with the historical production data closely. In the last case, different reservoir parameters were used to honor the uncertainty of the parameters. The optimized well locations were reasonably close to the real production wells in all cases, considering the model was developed using limited amount of available information about the reservoir. It proves the efficacy and robustness of GP meta-models.

CHAPTER 6 CONCLUSIONS

In this research, steps were taken to develop efficient proxy models for porous media flow simulators which tend to be very expensive. Effective nearly orthogonal space filling design of experiment is proposed for the initial set of simulation. Appropriate kernel is identified for the GP modeling using the proposed algorithm that utilizes leave-one-out cross-validation. As the number of simulation required is small, use of LOOCV is justified. In the case studies considered, neural network kernels showed the most promising results among the basis kernels. Following are the conclusions from this study.

6.1 CONCLUSIONS

- The distance correlation-based method to measure the performance of Design of Experiment is proposed along with different algorithms to generate efficient experimental designs. Several designs generated using the proposed algorithms are compared in terms of existing and proposed metrics. The performance of these designs are compared with that of the designs published in literature. The proposed methods are capable of generating nearly-orthogonal space-filling designs quickly. Improved results are obtained when the proposed design approach is applied to the 'borhole' model in comparison to a recent topical study.
- Several modifications are proposed to the existing quantum-behaved particle swarm optimization (QPSO) algorithm. Performance of the proposed algorithm is compared with that of the other algorithms for different standard benchmark functions with 10, 30 and 50 dimensions. Modifications are proposed for discrete

value constrained optimization as well. The modified optimization algorithm is applied to pressure vessel design problem, a standard test problem for optimization, as well. Results from the benchmark functions and the test problem are promising and shows the efficacy of the proposed algorithm.

- Simultaneous estimation of four parameters of the virus transport problem using Gaussian Process-based proxy models with squared exponential kernel is carried out succesfully using different distance metrics to convert several temporal vectors to a scalar quantity. Increasing the observation frequency improved the quality of the solutions of this inverse problem.
- An algorithm is proposed to find the proper composite kernel for a GP model that can capture the underlying process more efficiently. Leave-one-out cross-validation is used to compare the performance of different kernels (including the composite ones). The proposed method is successfully applied to three different problems. For parameter estimation of the virus transport problem, the estimated parameters are closer to the known values when compared with the GP method with a base kernel. Composite kernel GPs are also applied for the history matching problem of a standard oil reservoir and the placement of wells at optimal locations in a gas reservoir. Results from both studies are promising. As such, GP models can be used as effcient proxy when proper kernels are used.

6.2 FUTURE RESEARCH RECOMMENDATIONS

The use of GP-based proxy models for several porous media flow problems show promising results in this thesis. In the future, the following research should be carried out to establish it as an efficient tool.

- In this research, the proxy models have been created with the simulator output data. By nature, output generated from computer programs are noiseless. However, in general, obtained data from real problems are corrupted with noise. The governing equations for GP-model incorporate the noise term and the current research can be tested for noisy data easily.
- So far, both inference and prediction are considered to be Gaussian. The GP model can be extended to incorporate other distributions as well. However, in such cases, analytic solutions might not be available and the approximation of the integrals in the governing equation should be carried out.
- In this research, multiple temporal outputs are converted to a scalar quantity to honor a single output GP model. This can be extended to multiple output GP models by incorporating spatio-temporal kernels or convolution of kernels.

BIBLIOGRAPHY

Allen, D. M. (1974). The Relationship between Variable Selection and Data Agumentation and a Method for Prediction. *Technometrics, 16*(1), 125-127.

Arlot, S., & Celisse, A. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys, 4*, 40-79.

Bangerth, W., Klie, H., Wheeler, M., Stoffa, P., & Sen, M. (2006). On optimization algorithms for the reservoir oil well placement problem. *Computational Geosciences, 10*(3), 303-319.

Barthelemy, J. F., & Haftka, R. T. (1993). Approximation concepts for optimum structural design - A review. *Structural Optimization, 5*(3), 129-144. Retrieved from http://dx.doi.org/10.1007/bf01743349

Bhattacharjya, R. K., & Datta, B. (2005). Optimal management of coastal aquifer using linked simulation optimization approach. *Water Resour. Manage., 19*(3), 295–320.

Bhattacharjya, R. K., Srivatava, A., & Satish, M. G. (2014). Hybrid-Optimization Approach for Estimating Parameters of a Virus Transport Process in Aquifer. *Journal of Hazardous, Toxic, and Radioactive Waste*. doi:10.1061/(ASCE)HZ.2153-5515.0000195

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning.* Springer.

Cai, X., Zeng, J., Cui, Z., & Tan, Y. (2007). Particle Swarm Optimization Using Lévy Probability Distribution. In *LNCS 4683* (pp. 353-361).

Clark, A., Ng, J. Q., Morlet, N., & Semmens, J. B. (2016). Big Data And Ophthalmic Research. *Survey of Ophthalmology*, Accepted manuscript.

Clerc, M., & Kennedy, J. (2002). The Particle Swarm: Explosion, Stability, and Convergence in a Multi-dimensional Complex Space. *IEEE Transaction on Evolutionary Computation, 6*, 58-73.

Coelho, L. d. (2008). A quantum particle swarm optimizer with chaotic mutation operator. *Chaos, Solitons and Fractals, 37*, 1409–1418.

Coelho, L. d. (2010). Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. *Expert Systems with Applications, 37*, 1676 - 1683.

Couckuyt, I., Dhaene, T., & Demeester, P. (2014). ooDACE Toolbox: A Flexible Object-Oriented Kriging Implementation. *Journal of Machine Learning Research, 15*, 3183-3186.

Couckuyt, I., Forrester, A., Gorissen, D., De Turck, F., & Dhaene, T. (2012). Blind Kriging: Implementation and performance analysis. *Advances in Engineering Software, 49*(3), 1-13.

Cranganu, C., Luchian, H., & Breaban, M. E. (Eds.). (2015). Artificial Intelligent Approaches in Petroleum Geosciences. Springer International Publishing. doi:10.1007/978-3-319-16531-8

Demirel, M., & Kayan, B. (2012). Application of response surface methodology and central composite design for the optimization of textile dye degradation by wet air oxidation. *International Journal of Industrial Chemistry, 3*(24).

den Hertog, D., & Stehouwer, H. (2002). Optimizing color picture tubes by high-cost nonlinear programming. *European Journal on Operations Research, 140*(2), 197–211.

Department of Earth Science and Engineering, I. C. (2015). *Standard Models - PUNQ-S3 Model*. Retrieved January 15, 2014, from https://www.imperial.ac.uk/engineering/departments/earth-science/research/research-groups/perm/standard-models/

Dong, X., Wu, Z., Dong, C., Chen, Z., & Wang, H. (2011). Optimization of vertical well placement by using a hybrid particle swarm optimization. *Wuhan University Journal of Natural Sciences, 16*(3), 237-240.

Emch, P. G., & Yeh, W. W. (1998). Management model for conjunctive use of coastal surface water and ground water. *J. Water Resour. Plann.Manage., 124*(3), 129–139. doi:10.1061/(ASCE)0733-9496(1998)124:3(129)

Fang, W., Sun, J., Ding, Y., X., W., & Xu, W. (2010). A Review of Quantum-behaved Particle Swarm Optimization. *IETE Tech Rev, 27*(4), 336-348.

Finney, B. A., Samsuhadi, & Willis, R. (1992). Quasi-three-dimensional optimization model of Jakarta Basin. *J. Water Resour. Plann. Manage.*, 18–31. doi:10.1061/(ASCE)0733-9496(1992)118:1(18)

Fluid Mechanics. By P. K. KUNDU I. M. COHEN. Academic, 2002. 730 pp. ISBN 0-12-178251-4. 69.95. (2003, feb). *J. Fluid Mech., 477*, 410-411. Retrieved from http://dx.doi.org/10.1017/s0022112002233704

Forrester, A. I., & Keane, A. J. (2009). Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences, 45*(1–3), 50-79.

Forrester, A., & Jones, D. (2008). Global optimization of deceptive functions with sparse sampling. *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization*

Conference,American Institute of Aeronautics and Astronautics. Retrieved June 25, 2014, from http://www.southampton.ac.uk/~aijf197/ForresterandJones.pdf

Forrester, A., Sobester, A., & Keane, A. (2008). *Engineering design via surrogate modelling: a practical guide.* John Wiley & Sons.

Franey, M., Ranjan, P., & Chipman, H. (2011). Branch and bound algorithms for maximizing expected improvement functions. *Journal of Statistical Planning and Inference, 141*, 42 - 55.

Gönen, M., & Alpaydın, E. (2011). Multiple Kernel Learning Algorithms. *Journal of Machine Learning Research, 12*, 2211-2268.

Grolinger, K., L'Heureux, A., Capretz, M. A., & Seewald, L. (2016). Energy Forecasting for Event Venues: Big Data and Prediction Accuracy. *Energy and Buildings, 112*, 222–233.

Hari Prasad, K. S., Ratha, D. N., & Ojha, C. S. (2012). Identification of virus transport parameters in groundwater: Data errors and bias. *Journal of Hydro-environment Research, 6* , 41-50.

Hu, X., Eberhart, R., & Shi, Y. (2003). Engineering optimization with particle swarm. *IEEE Congress on Swarm Intelligence*, (pp. 53-57). Indianapolis (USA).

Hung, Y. (2014). Sequential probability-based Latin hypercube designs without replacement. *Statistica Sinica*. Retrieved from http://dx.doi.org/10.5705/ss.2012.335

Jiang, M., Luo, Y., & Yang, S. (2007). Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Information Processing Letters, 102*, 8 - 16.

Joe, S., & Kuo, F. Y. (2003, March). Remark on Algorithm 659: Implementing Sobol's Quasirandom Sequence Generator. *ACM Trans. Math. Softw., 29*(1), 49--57.

Joe, S., & Kuo, F. Y. (2008). Constructing Sobol Sequences with Better Two-Dimensional Projections. *SIAM Journal on Scientific Computing, 30*(5), 2635-2654.

Jourdan, A., & Franco, J. (2010, dec). Optimal Latin hypercube designs for the Kullback-Leibler criterion. *Advances in Statistical Analysis, 94*(4), 341-351.

Kadane, J. B., & Lazar, N. A. (2004). Methods and Criteria for Model Selection. *Journal of the American Statistical Association, 99*(465), 279-290.

Kadane, J. B., & Lazar, N. A. (2004). Methods and Criteria for Model Selection. *Journal of the American Statistical Association, 99*(465), 279-290.

Kaveh, A., Sheikholeslami, R., Talatahari, S., & Keshvari-Ilkhichi, M. (2014). Chaotic swarming of particles: A new method for size optimization of truss structures. *Advances in Engineering Software, 67*, 136–147.

Kelley, C. T. (1999, jan). *Iterative Methods for Optimization.* Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611970920

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. *Proceedings of IEEE International Conference on Neural Networks*, *4*, pp. 1942-1948.

Kim, G.-B. (2015). Optimal distribution of groundwater monitoring wells near the river barrages of the 4MRRP using a numerical model and topographic analysis. *Environmental Earth Sciences, 73*(9), 5497-5511.

Kleijnen, J. (1995). Sensitivity analysis and optimization in simulation: design of experiments and case studies. *Winter Simulation Conference Proceedings, 1995.* Institute of Electrical & Electronics Engineers. doi:10.1109/wsc.1995.478715

Kleijnen, J. P. (2015). Response Surface Methodology. In M. C. Fu (Ed.), *Handbook of Simulation Optimization* (pp. 81-104). Springer New York.

Kleijnen, J. P., & Mehdad, E. (2014). Multivariate versus univariate Kriging metamodels for multi-response simulation models. *European Journal of Operational Research, 236*(2), 573-582.

Kluwer Academic Publishers. (1996, aug). *Annals of Oncology, 7*(6), 649-649. Retrieved from http://dx.doi.org/10.1093/oxfordjournals.annonc.a010697

Koh, B.-I., George, A. D., Haftka, R. T., & Fregly, B. J. (2006). Parallel asynchronous particle swarm optimization. *International Journal for Numerical Methods in Engineering, 67*(4), 578-595.

Kroese, D. P., & Chan, J. C. (2013). Monte Carlo Sampling. In *Statistical Modeling and Computation* (pp. 195-226). Springer New York.

Li, R. (2002, dec). Model Selection for Analysis of Uniform Design and Computer Experiment. *International Journal of Reliability, Quality and Safety Engineering, 09*(04), 367-382. doi:10.1142/s0218539302000901

Li, Y., Li, D., & Wang, D. (2012). Quantum-Behaved Particle Swarm Optimization Algorithm Based on Border Mutation and Chaos for Vehicle Routing Problem. *ICSI 2012, Part I, LNCS 7331*, (pp. 63-73).

Liefvendahl, M., & Stocki, R. (2006). A study on algorithms for optimization of Latin hypercubes. *Journal of Statistical Planning and Inference, 136*(9), 3231-3247. doi:10.1016/j.jspi.2005.01.007

Lim, W. H., & Isa, N. A. (2014). Particle swarm optimization with increasing topology connectivity. *Engineering Applications of Artificial Intelligence, 27*, 80-102.

Lin, C. D., & Kang, L. (2016). A General Construction for Space-filling Latin Hypercubes. *Statistica Sinica*. doi:10.5705/ss.202015.0019

Liu, J. B., Shen, Z. X., & Lu, Y. L. (2014). Optimal Antenna Design With QPSO–QN Optimization Strategy. *IEEE Transactions on Magnetics, 50*(2).

Ma, C.-X., Fang, K.-T., & Liski, E. (2000). A new approach in constructing orthogonal and nearly orthogonal arrays. *Metrika, 50*(3), 255-268.

McKay, M. D., Beckman, R. J., & Conover, W. J. (1979). Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. *Technometrics, 21*(2), 239-245. doi:10.1080/00401706.1979.10489755

Moon, H., Dean, A., & Santner, T. (2011). Algorithms for Generating Maximin Latin Hypercube and Orthogonal Designs. *Journal of Statistical Theory and Practice, 5*(1), 81-98. doi:10.1080/15598608.2011.10412052

Morris, M. D., & Mitchell, T. J. (1995). Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference, 43*(3), 381-402. doi:10.1016/0378-3758(94)00035-t

Mukerjee, R., & Wu, C. J. (2006). *A Modern Theory of Factorial Designs.* Springer New York. doi:10.1007/0-387-37344-6

Newman-Bennett, M., Zaman, M. S., Satish, M. G., & Islam, M. R. (2008). Modeling Of North Triumph Gas Reservoir For Carbon-dioxide Sequestration – A Case Study. *WIT Transactions on Engineering Sciences, 59*, 447-455. doi:10.2495/AFM080441

Oliver, D. S., Reynolds, A. C., & Liu, N. (2008). *Inverse Theory for Petroleum Reservoir Characterization and History Matching.* Cambridge University Press.

Owen, A. B. (1994). Controlling Correlations in Latin Hypercube Samples. *Journal of the American Statistical Association, 89*(428), 1517-1522. doi:10.1080/01621459.1994.10476891

Peng, H., Yan, H., & Zhang, W. (2013). The connection between cross-validation and Akaike information criterion in a semiparametric family. *Journal of Nonparametric Statistics, 25*(2), 475-485.

Pillonetto, G., & Nicolao, G. D. (2011). Kernel Selection in Linear System Identification, Part I: A Gaussian Process Perspective. *IEEE Conference on Decision and Control and European Control Conference (CDC-ECC)*, (pp. 4318-4325). Orlando, FL, USA.

Pintér, J. D., & Horváth, Z. (2013). Integrated experimental design and nonlinear optimization to handle computationally expensive models under resource constraints. *Journal of Global Optimization, 57*(1), 191-215.

Plevris, V., & Papadrakakis, M. (2011). A Hybrid Particle Swarm—Gradient Algorithm for Global Structural Optimization. *Computer-Aided Civil and Infrastructure Engineering, 26*(1), 1057-1071.

Powelson, D. K., & Gerba, C. P. (1994). Viral removal from sewage effluents during saturated and unsaturated flow through soil columns. *Water Research, 28*, 2175–2181.

Powelson, D. K., Gerba, C. P., & Yahya, M. T. (1993). Virus transport and removal in wastewater during aquifer recharge. *Water Resources, 27*, 583-590.

Powelson, D. K., Simpson, J. R., & Gerba, C. P. (1990). Virus transport and survival in saturated and unsaturated flow through soil columns. *J. Environ. Qual., 19*, 396–401.

Powelson, D. K., Simpson, J. R., & Gerba, C. P. (1991). Effect of organic matter on virus transport in unsaturated flow. *Appl. Environ. Microbiol., 57*, 2192–2196.

Pradhan, S., & Patra, D. (2015). RMI based non-rigid image registration using BF-QPSO optimization and P-spline. *International Journal of Electronics and Communications, 69*(3), 609–621.

Pronzato, L., & MÃ¼ller, W. G. (2011, apr). Design of computer experiments: space filling and beyond. *Statistics and Computing, 22*(3), 681-701. Retrieved from http://dx.doi.org/10.1007/s11222-011-9242-3

Ranganathan, P. (2011, January). From Microprocessors to Nanostores: Rethinking Data-Centric Systems. *Computer, 44*(1), 39-48.

Rasmussen, C. E., & Williams, C. K. (2006). *Gaussian Processes for Machine Learning.* the MIT Press. Retrieved from http://www.GaussianProcess.org/gpml

Ratha, D. N., Hari Prasad, K. S., & Ojha, C. S. (2009). Analysis of virus transport in groundwater and identification of transport parameters. *Pract. Period. Hazard. Toxic Radioact. Waste Manage., 13*(2), 98-109.

Redman, J. A., Grant, S. B., Olson, T. M., & Estes, M. K. (2001). Pathogen filtration, heterogeneity, and potable reuse of wastewater. *Environ. Sci. Technol., 35*, 1798-1805.

Rezaee Jordehi, A., & Jasni, J. (2012). Particle swarm optimisation for discrete optimisation problems: a review. *Artificial Intelligence Review*, 1-16.

Rikards, R., & Auzins, J. (2004, feb). Response surface method for solution of structural identification problems. *Inverse Problems in Science and Engineering, 12*(1), 59-70. Retrieved from http://dx.doi.org/10.1080/10682760310001597446

Robertson, W., Cherry, J., & Sudicky, E. (1991). Ground-water contamination from two small septic systems on sand aquifers. *Ground Water, 29*(1), 82-92.

Schutte, J. F., Reinbolt, J. A., Fregly, B. J., Haftka, R. T., & George, A. D. (2004). Parallel global optimization with the particle swarm algorithm. *International Journal for Numerical Methods in Engineering, 61*(13), 2296–2315.

Seethaa, N., Kumar, M. S., & Hassanizadeh, S. M. (2015). Modeling the co-transport of viruses and colloids in unsaturated porous media. *Journal of Contaminant Hydrology, 181*, 82–101.

Shamsuzzaman, M., Satish, M. G., & Pinter, J. D. (2015). Distance correlation - based nearly orthogonal space-filling experimental designs. *Int. J. Experimental Design and Process Optimisation, 4*(3/4), 216-233.

Shao, J. (1996). Bootstrap Model Selection. *Journal of the American Statistical Association, 91*(434), 655–665. doi: http://doi.org/10.2307/2291661

Sim, Y., & Chrysikopoulos, C. V. (1995). Analytical models for one-dimensional virus transport in saturated porous media. *Water Resour. Res. , 31*(5), 1429-1437.

Sim, Y., & Chrysikopoulos, C. V. (2000). Virus transport in unsaturated porous media. *Water Resour. Res., 36* (1), 173-179.

Simpson, T., Booker, A., Ghosh, D., Giunta, A., Koch, P., & Yang, R.-J. (2004, jun). Approximation methods in multidisciplinary analysis and optimization: a panel discussion. *Struct Multidisc Optim, 27*(5). Retrieved from http://dx.doi.org/10.1007/s00158-004-0389-9

Singh, M. R., & Mahapatra, S. (2016). A quantum behaved particle swarm optimization for flexible job shop scheduling. *Computers & Industrial Engineering, 93*, 36–44.

Sobol, I. M. (1967). Distribution of points in a cube and approximate evaluation of integrals. *U.S.S.R Comput. Maths. Math. Phys. (in English), 7*, 86–112.

Spitzner, D. J. (2007). Constructive Cross-Validation in Linear Prediction. *Communications in Statistics - Theory and Methods, 36*(5), 939-953.

Suganthan, P. N. (1999). Particle swarm optimizer with neighborhood operator. *Proceedings of the 1999 Congress on Evolutionary Computation, CEC'99*, (pp. 1958–1961). Wasington, DC, USA,.

Suganthan, P., Hansen, N., Liang, J., Deb, K., Chen, Y., Auger, A., & Tiwari, S. (2005). *Problem Definitions and Evaluation Criteria for the CEC2005, Special Session on Real Parameter Optimization.*

Sun, J., Fang, W., Palade, V. W., & Xu, W. (2011). Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point. *Applied Mathematics and Computation, 218*, 3763–3775.

Sun, J., Fang, W., Palade, V., Wu, X., & Xu, W. (2011). Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point. *Applied Mathematics and Computation, 218*, 3763 - 3775.

Sun, J., Fang, W., Wu, X., Palade, V., & Xu, W. (2012). Quantum-behaved Particle Swarm Optimization: Analysis of Individual Particle Behavior and Parameter Selection. *Evol. Comput., 20*(3), 349--393.

Sun, J., Feng, B., & Xu, W. (2004). Particle swarm optimization with particles having quantum behavior. *Congress on Evolutionary Computation CEC2004*, *1*, pp. 325-331.

Sun, J., Xu, W., & Fang, W. (2006). Quantum-behaved particle swarm optimization with a hybrid probability distribution. *PRICAI 2006: Trends in Artificial Intelligence*, (pp. 737-746).

Sun, J., Xu, W., & Fang, W. (2006a). Quantum-Behaved Particle Swarm Optimization Algorithm with Controlled Diversity. *LNCS. 3993*, pp. 847 - 854. Springer.

Sun, J., Xu, W., & Fang, W. (2006b). Enhancing Global Search Ability of Quantum-Behaved Particle Swarm Optimization by Maintaining Diversity of the Swarm. *LNAI*, *4259*, pp. 736 - 745.

Sun, J., Xu, W., & Feng, B. (2004). A global search strategy of quantum-behaved particle swarm optimization. *IEEE Conference on Cybernetics and Intelligent Systems*, *1*, pp. 111-116.

Sun, J., Xu, W., & Liu, J. (2005). Parameter Selection of Quantum-Behaved Particle Swarm Optimization. In L. Wang, K. Chen, & Y. Ong (Eds.), *Advances in Natural Computation* (Vol. 3612, pp. 543-552). Springer.

Székely, G. J., & Rizzo, M. L. (2009). Brownian distance covariance. *Annals of Applied Statistics, 3*(4), 1233-1303. doi:10.1214/09-aoas312rej

Székely, G. J., Rizzo, M. L., & Bakirov, N. K. (2007). Measuring and testing dependence by correlation of distances. *Annals of Statistics, 35*(6), 2769-2794. doi:10.1214/009053607000000505

The MathWorks Inc. (2014). Retrieved December 20, 2014, from Matlab Documentation: http://www.mathworks.com/help/matlab/ref/pdepe.html

Tim, U. S., & Mostaghimi, S. (1991). Model for Predicting virus movement through soils. *Ground Water, 29*(2), 251-259.

van Dam, E. R., Husslage, B., den Hertog, D., & Melissen, H. (2007, feb). Maximin Latin Hypercube Designs in Two Dimensions. *Operations Research, 55*(1), 158-169. Retrieved from http://dx.doi.org/10.1287/opre.1060.0317

van Dam, E. R., Rennen, G., & Husslage, B. (2009, jun). Bounds for Maximin Latin Hypercube Designs. *Operations Research, 57*(3), 595-608. Retrieved from http://dx.doi.org/10.1287/opre.1080.0604

van den Bergh, F. (2001). *An Analysis of Particle Swarm Optimizers.* PhD Thesis, University of Pretoria.

van den Bergh, F., & Engelbrecht, A. P. (2006). A study of particle swarm optimization particle trajectories. *Information Sciences, 176*, 937 - 971.

van Genuchten, M. T., & Alves, W. J. (1982). *Analytical Solutions of the One-Dimensional Convective-Dispersive Solute Transport Equation.* Technical Bulletin 1661, Agricultural Research Service, United States Department of

Agriculture. Retrieved May 01, 2015, from http://ars.usda.gov/sp2UserFiles/Place/20360500/pdf_pubs/P0753.pdf

Viana, F. A., Venter, G., & Balabanov, V. (2009). An algorithm for fast optimal Latin hypercube design of experiments. *International Journal for Numerical Methods in Engineering*, n/a--n/a. Retrieved from http://dx.doi.org/10.1002/nme.2750

Vilker, V., Frommhagen, L., Kamda, R., & Sundaram, S. (1978). Application of ion exchange/adsorption models to virus transprt in percolating beds. *AIChE Symposium Series, 74*(178), 84-89.

Wang, H., Sun, H., Li, C., Rahnamayan, S., & Pan, J.-s. (2013). Diversity enhanced particle swarm optimization with neighborhood search. *Information Sciences, 223*, 119 - 135.

Wilson, A. G., & Adams, R. P. (2013). Gaussian Process Kernels for Pattern Discovery and Extrapolation. *preprint: arXiv:1302.4245v3*.

Wolf, A., Henes, D., Bogdanski, S., Lutz, T., & Krämer, E. (2013). Statistical analysis of parameter variations using the Taguchi method. *Management and Minimisation of Uncertainties and Errors in Numerical Aerodynamics, 122*, pp. 247–264.

Wong, T.-T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. *Pattern Recognition, 48*, 2839-2846.

Xavier, C. R., Santos, E. P., Vieira, V. d., & Santos, R. W. (2013). Genetic Algorithm for the History Matching Problem. *Procedia Computer Science, 18* , 946–955.

Yang, X.-S., Huyck, C., Karamanoglu, M., & Khan, N. (2013). True global optimality of the pressure vessel design problem: a benchmark for bio-inspired optimisation algorithms. *Int. J. Bio-Inspired Computation, 5*(6), 329-335.

Yang, Z.-L., Wu, A., & Min, H.-Q. (2015). An Improved Quantum-Behaved Particle Swarm Optimization Algorithm with Elitist Breeding for Unconstrained Optimization. *Computational Intelligence and Neuroscience, 2015*. doi:10.1155/2015/326431

Yates, M. V., & Jury, W. A. (1988). Modeling microbial fate in the subsurface environment. *CRC Crit. Rev. Environ. Control, 17*, 307–344.

Yates, M. V., & Yates, S. R. (1988). Modeling microbial fate in the subsurface environment. *Crit. Rev. Environ. Contr., 17*(4), 307-344.

Yates, M. V., Gerba, C. P., & Kelley, L. M. (1985). Virus persistence in groundwater. *Appl. Environ. Microbiol., 49*(4), 778-781.

Ye, K. Q. (1998, dec). Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments. *Journal of the American Statistical Association, 93*(444), 1430-1439. Retrieved from http://dx.doi.org/10.1080/01621459.1998.10473803

Yeten, B., Durlofsky, L. J., & Aziz, K. (2003, September 1). Optimization of Nonconventional Well Type, Location, and Trajectory. doi:10.2118/86880-PA

Zaydullin, R., Voskov, D. V., James, S. C., Henley, H., & Lucia, A. (2014). Fully compositional and thermal reservoir simulation. *Computers & Chemical Engineering, 63*, 51–65.

Zhang, Y., & Yang, Y. (2015). Cross-validation for selecting a model selection procedure. *Journal of Econometrics, 187*, 95-112.

Zhou, K., & Yang, S. (2016). Understanding household energy consumption behavior: The contribution of energy big data analytics. *Renewable and Sustainable Energy Reviews, 56*, 810–819.

Ziomek-Moroz, M. (2012). Environmentally Assisted Cracking of Drill Pipes in Deep Drilling Oil and Natural Gas Wells. *Journal of Materials Engineering and Performance, 21*(6), 1061-1069.