Use of Hybrid Surrogate in Sampling Process for Design and Analysis of Computer Experiments (DACE) applied on Multivariable Problems

by

Odilon Rodrigues Filho

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Naval Architecture and Marine Engineering)
in the University of Michigan
2022

Doctoral Committee:

Professor Nickolas Vlahopolous, Chair
Associate Professor Mattew D. Collete
Professor Emeritus John Lee
Dr. Ricardo Sbragio
Professor David J. Singer

Odilon Rodrigues Filho
odilon@umich.edu
ORCID iD: 0000-0002-4047-0467

©Odilon Rodrigues Filho 2022

To my parents Odilon and Nevolanda, to my wife Ana Paula, Bella and my sisters Regina and Kelly.

ACKNOWLEDGMENTS

I would like to thank my advisor Professor Nickolas Vlahopoulos that provided me a singular and intense orientation in such manner that without his help, paciency and friendship, this work could not be possible. He guided me during the whole work performed with enough orientations, experiences and recommendations to solve aparent complex problems with simple ingredients.

I also acknowledge my sincere friend Ricardo Sbragio who provided me intense help and directions since when I arrived in Ann Arbor to start my MS Program in Naval Architecture and Marine Engineering Department (NAME) at University of Michigan (UM). He encouraged me and guided me to superate the challenges that I have been faced on since that time.

Similarly, I would like to thank to my dissertation committee Professor Matthew Collette, David Singer and John Lee who followed me on the development of this work over this time and provided me valorous advices. My gratitude is also addressed to Mrs Nathalie Fiveland, who helped me throughout this period in the department. I cannot forget to UM, in special to the Departments of Naval Architecture and Marine Engineering and Nuclear Engineering and Radiological Sciences (NERS), where I took my MS degrees and provided me a wonderful and challengeable moments in my life. It was a time that made me change for a better person.

Finally, my most sincere thanks to my wife Ana Paula and to my parents Odilon and Nevolanda who encouraged me to start this PhD program, even in a hard moment, and helped me during the development of this work. They always gave me hope to overcome whatever difficulties that could come.
Thank you!

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AK Adaptive Kriging

CK Conventional Kriging

cp control points

DACE Design and Analysis of Computer Experiments

dMSE derivative of MSE

dMSE1 derivative of MSE related to x_1

dMSE2 derivative of MSE related to x_2

DoE Design of Experiments

FEM Finite Element Method

GA Genetic Algorithm

HC Hybrid Code

IPSO Improved Particle Swarm Optimization

IWO Improved Whale Optimization

LHS Latin Hypercube Sampling

MARS Multivariate Adaptive Regression Splines

MSE Mean Squared Error

N_a number of points to be added

ngen number of generations

NNW Neural Network

OA Orthogonal Arrays

OLHS Optimum Latin Hypercube Sampling

PCA Principal Component Analysis

RBF Radial Basis Function

RND Random Sampling Design

SBGO Surrogate-Based Global Optimization

sp sample points

SVM Support Vector Machine

UNDEX UNDerwater EXplosion

ABSTRACT

This work focuses on the improvement of Kriging in Design and Analysis of Computer Experiments (DACE) for multivariable problems through a specific sampling process that couples two methods already known in the literature: Adaptive Kriging and Genetic Algorithm. The special integration of these methods to enhance the sampling process is the main contribution of this dissertation. This procedure optimizes the sampling process of the metamodel finding specific location in the design space with relative few samples, which brings higher fidelity and better performance than conventional methods based on pure Conventional Kriging. Mean squared error (MSE) in Adaptive Kriging and its coupling with MSE derivative (dMSE) in objective functions of Genetic Algorithm are the metric used for sampling improvement. A dedicated random mesh is implemented in the design space covering all variables of the problem. The Adaptive Kriging works as a first contribution for sampling improvement, which is followed by the Genetic Algorithm based on the NSGAII code. The Genetic operations are used to exploit such mesh in order to identify specific locations with high potential for enhancement of metamodel fidelity with relative low quantity of sample points. All of these steps configure the proposed method called Hybrid Code that couples the Adaptive Kriging with Genetic Algorithm.

Three analytical problems are considered for Hybrid Code validation: Branin function and Product Peak Integrand Family function for two and ten variables. Once the validation is performed, the hybrid code is used to solve two numerical representative problems of FEM engineering applications: cantilever beam and underwater explosion effect on a submerged stiffened plate clamped at its edges. In each case, different commercial FEM code have been used: ANSYS for the former and ABAQUS for the latter. In the cantilever beam

problem, the output analyzed for each individual sample point are the equivalent stress at the clamp location and vertical displacement at the free end of the beam. This problem contains ten variables defined as thickness of each section of the beam. For the underwater explosion problem by its turn, the output analyzed is the vertical displacement at the middle of the plate. This problem has six variables defined as the thickness of each structural element of the plate, such as planking, bar, flange, etc.

In all analyses, the metamodel performance obtained by the proposed method is compared with the one obtained from Conventional Kriging. The performance measurement parameters are based on non-dimensional error in comparison to the known response and the computational time required. The conclusion based on the results collected is that the Hybrid Code generates a higher fidelity metamodel in a faster manner than the Conventional Kriging for high-dimensional and/or more complex problems.

CHAPTER 1

Introduction

1.1 Motivation

As illustrated by Ye (2019) and Fernandez et al. (2019), in recent years, the evolution of trade relations around the world has pushed different developments searching for global optimization in different aspects such as time, performance and cost. These developments are based on optimization process itself, machine learning evolution, statistical methods, or by the processing capacity of computers. Considering the optimization process and machine learning evolution, specific techniques, named in the literature as Surrogate-Based Global Optimization (SBGO), rise up as a large and complex niche that needs to superate some obstacles.

In this scenario, the constraints that need to be overcomed can be identified as following:

- Accuracy of models to be used in the optimization process.
- Quantity of available known data to explore the design space searching for the global maximum/minimum solution.

The first obstacle is straightforward connected to metamodels that aims to speed up the optimization process allied to the challenge of reaching a reasonable accuracy related to the physical model that it represents.

In the second constraint, the availability of known data can configure a problem due to the long time or impossibility to produce all necessary quantity. This scenario can become worse when large amount of variables are involved. This situation imposes the consideration of sparse data, which also contributes to the low fidelity of metamodels that impacts

the performance faced on the first constraint.

A simple example that can illustrate both abovementioned aspects is shown in figure 1.1. The original function is represented by the blue line. Two different set of samples are shown as green marks. In the top curves, bad samples are located in bad regions of the design space, which produce a not accurate metamodel represented as red line. The opposite occurs in the bottom curves, which specific locations of samples make possible to improve the accuracy of the final metamodel.

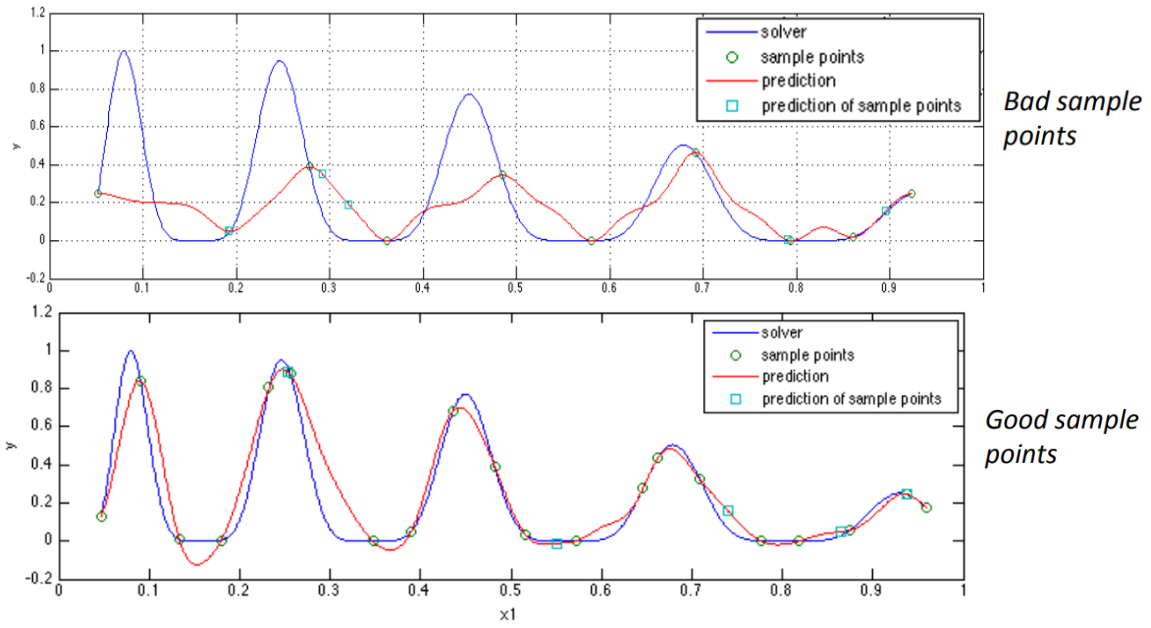


Figure 1.1: Examples of bad and good sample points influence on the final accuracy of metamodel for a single variable problem.

However, the most interesting scenario for investigation is when large quantity of variables are employed, which inflates the severity of two constraints previously mentioned and imposes a huge challenge on the achievement of an efficient metamodel within an acceptable computational time. Considering that, searching for a set of conditions with relative low quantity of samples that make possible this achievement configures a significant evolution in general optimization process as discussed in H. Liu et al. (2016). For fidelity issue, the combination of different metamodel methods has been the most recent evolution technique. Such mixing approach is named as Hybrid Method and aims to get the best advantages of each method to work symbioticaly on a single process. It addresses to the called hybrid metamodel. For the sampling process, the mixing of different sampling techniques such as Latin Hypercube Sampling (LHS) and Orthogonal Arrays (OA) is also

a relative recent evolution. Similarly to the previous mentioned technique, this aproach becomes the called *hybrid sampling*. As result, implementing such strategy in DACE context becomes a logic application due to its relative faster and cheaper analysis.

Hence, this research applies the *hybrid metamodel* concept in the *hybrid sampling* analysis in order to reach a faster and more accurate metamodel. All details of this work are described in next chapters.

1.2 Literature Review

1.2.1 DACE and DoE

The first aspect that is necessary to contextualize in the current work is the application addressed to Design and Analysis of Computer Experiments (DACE), which is different from the Design of Experiments (DoE) concept. Several authors explored DACE, such as Sacks et al. (1989), T. Simpson et al. (2002), Etman (1994) in general approach. In DoE, as classical definition, the manufacturing issues, human error, and other factors related to physical experiments work as random input and induce to a probabilistic approach of the problem treatment. Works that can be highlighted are Kleijnen et al. (2005) and Kleijnen et al. (1997). In the opposite sense, DACE considers numerical models also in a probabilistic approach, which has particular advantages over DoE such as being less expensive and more flexible due to its ability to involve the representativeness of complex physical models under large quantity of variables in a relatively easier manner. Also, this basic difference has a strong effect on the sampling process in the design space that allows DACE to explore it easier (Sacks et al., 1989 and T. Simpson et al., 2002). Other works such as T. W. Simpson et al. (2004) and Baco et al. (2019) make an interesting comparison between both techniques. In specific application of DoE and DACE, two works can be cited as representative of both approaches. In the first work, Thurman et al. (2020) explores a metamodel generation for finding side force coefficients on hydrodynamic bodies considering both DoE and DACE techniques and presents the comparison between them. The authors indicate DoE as an efficient way to produce a metamodel preserving the link with the physical phenomenon. Nevertheless, DACE is indicated by the authors as the potential approach to produce more accurate metamodels. In the second work, Joe (2017) addresses human factors to generate a metamodel in order to develop the modernization of control rooms for nuclear power plant. This particular problem illustrates how much complex and how many variables can be involved in a specific problem with relative difficulty to sample

the input data.

Figure 1.2 illustrates the relationship among physical problem, physical model and metamodel. DoE and DACE uses the same parameter concepts, such as sample points (x_i and y_i) and test points (x_t and y_t). However, DACE takes advantage of physical model that considers a unique correlation between input sample x_i and output y_i (also between the input test point x_t and its output y_t). Considering such relationship between input and output, the sample points are used to get the metamodel. Next, the performance is verified comparing the estimate output $\hat{y}_t$ obtained from the input test points x_t with the known response y_t. This comparison between y_t and $\hat{y}_t$ provides the error of the process ϵ.

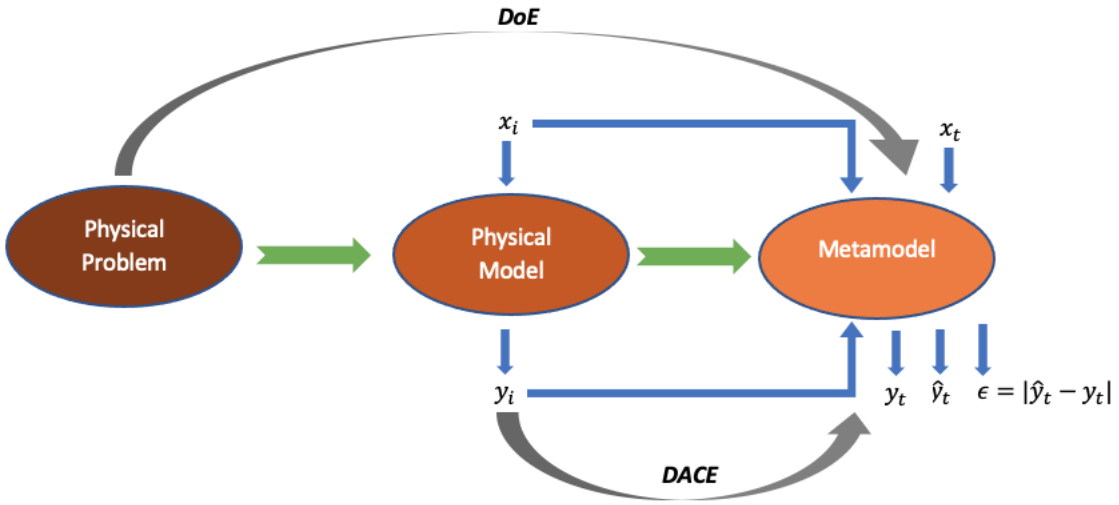


Figure 1.2: Relationship among physical problem, physical model and metamodel. y_i and x_i are considered as sample points and x_t and y_t are considered as test points.

Considering all aspects highlighted for DoE and DACE and specific characteristics of the later approach based on its flexibility, relative low resources needed and capability to investigate several different and complex problems in relatively short time, DACE has been chosen as the context for the current research.

1.2.2 Hybrid Metamodels

Another aspect investigated in the literature is the pursue for an accurate metamodel. In this context, Queipo et al. (2005), G. G. Wang et al. (2006), Forrester et al. (2009), Fernan-

dez et al. (2019) and Ye (2019) perform an overall review about this subject.

As G. G. Wang et al. (2006) describes in his study, an accurate metamodel can be reached by the use of different solvers such as Kriging, Radial Basis Function (RBF), Support Vector Machine (SVM) or Multivariate Adaptive Regression Splines (MARS) to generate surrogates able to mimitize the real behaviour of the physical model. In this context, the Kriging method rises up as a large employed solver that considers a deterministic response as random function from a given physical problem (Lophaven et al. (2002)). However, there is no conclusion that one single method is the best solver to reach a global precise surrogate. Each one has its own limitations and advantages depending on the type of problem involved. On the opposite way, what has been indicated in the last developments are the combination of different methods. As explained by G. G. Wang et al. (2006) and indicated by Fernandez et al. (2019), Ye (2019), K.-H. Lee et al. (2019) and Baquela et al. (2019), the definition of hybrid metamodels consists on the result of such combination collecting the best features that each single method has.

For example, Baquela et al. (2019) searches for a coupling between genetic algorithm and Kriging that updates the metamodel in each iteration searching for the best surrogate composition. Searching for the same goal, K.-H. Lee et al. (2019) combines different solvers reducing the design space. Similar approach was followed by Streltsov et al. (1995) that implemented sequence of partial random sampling in design space searching for global optimization. Gu et al. (2019) also worked with hybrid metamodels combining Kriging, RBF and quadratic functions starting from random sampling based on LHS. Other authors worked on adaptive hybrid approach such as Zhang et al. (2012), Zhang et al. (2013), Yu et al. (2014) and Zhou et al. (2016). In that sense, the adaptive characteristic is based on the change of internal parameters that play different participation of each solver in the pool where they are combined in the hybrid algorithm. Interesting to note that Zhang et al. (2013) compared three different sampling approaches (LHS, Sobol's quasi-random sequence, and Hammersley sequence), but only one is chosen as the best within the analysis performed. In that case, no mixing of sampling approach is adopted.

Nevertheless, good emphasis can be placed on Hacker et al. (2001) and L. Wang et al. (2019) that propose inovative hybrid methods for more efficient optimization. Hacker et al. (2001) considers the coupling of genetic algorithm with Sequential Linear Programming. L. Wang et al. (2019) proposes a new algorithm based on comprehensive learning strategy based on the combination of four different methods: Improved Particle Swarm Optimiza-

tion (IPSO) algorithm, Improved Whale Optimization (IWO) algorithm, improved archive mechanism and dual-population evolutionary mechanism.

What can be observed in the L. Wang et al. (2019) is that the hybrid technique not only is applied on surrogate models, but also for sampling, storage information and other steps of the whole process involved in the optimization operation. The next section explains the developments of hybrid methods applied on sampling processes.

1.2.3 Hybrid Sampling process

Hybrid sampling developments follow the same idea of hybrid metamodels, which is based on the combination of different methods to improve the efficiency of a given process. However, the main difference between hybrid metamodel and hybrid sampling approach is the availability for the latter on more options of test criteria to check the efficiency of an obtained sample. In such approach, the tests could be focused on internal sampling parameters, such as the maximum distance among the sampled points (Pan et al., 2014a).

In Pan et al. (2014b), a sequential optimization sampling based on extended radial basis functions is proposed. In this work, two different phases have been implemented, In the first one, the initial sample is obtained throught LHS. Then, the RBF takes place to generate a metamodel able to identify the extrema points of the response. From that step, density functions are computed in order to find sparse regions. The minimum points from these regions are added on the sampling as new points and this process continues until some finish criteria is satisfied. Interesting aspect in this work is the performance of the proposed hybrid sampling in comparison to traditional methods based on pure LHS and pure RND, which implements a sequential sampling getting information from the sample to foreseen where new sample can be added in the design space. The limiting aspect on this proposed algorithm is its parameterization, which is focused only on the maximum/minimum regions of the response surface.

Ran et al. (2018) also discusses the effect of an efficient sampling on the accuracy of the metamodel. Assuming one single stage of sampling is not enough to reach higher efficiency, Ran et al. (2018) considers two steps of sampling adopting the Optimum Latin Hypercube Sampling (OLHS) coupled with restriction of design space. The OLHS approach is a variance of LHS that is submited to an optimization process based on collumnwise-pairwise exchange process. In such method, parts of the global design space are removed

from the object of optimization sampling and focusing in regions of the global design space that have potential for sampling improvement. Yao et al. (2013) also considers the hybrid infill sample approach adopting the RBF neural network in two steps implementation. Additionally, Pan et al. (2014a) uses the LHS concept to relocate the sample point that is not well located from previous interation. In such processes, the sample does not change the size, but keeps its searching for a better performance. However, according to the results presented in all of these studies, some issues have been identified for high dimensional and/or complex problems.

1.2.4 Proposed Algorithm

Regarding the reviewed literature, each proposed method presents an effort to maximize the performance of the final metamodel. However, for high nonlinear and high dimension problems, it becomes a challenge. In this context, the proposed algorithm goes in the same hybrid approach direction considering the following:

- Spliting of design space, but not reducing it;

- Inserting the hybrid metamodel into the sampling process combining two different metamodel solvers;

- Collecting the best sample set considering online approach [1] in order to generate the final metamodel at once without the need to compare different metamodels to chose the best; and

- Searching for less computational time.

In this context, the proposed algorithm considers Adaptive Kriging (AK) and Genetic Algorithm (GA) as coupled solvers embedded in a code written in MATLAB. In next section, the main contribution of this work is indicated based on the abovementioned approach.

[1] Online approach consists in obtaining the sample and its response during the processing cycle. In the opposite, the offline method consists in getting all sample and finishing the whole process before obtaining the response of such points.

1.3 Dissertation Contribution

The main contribution of this dissertation is the use of hybrid metamodel concept in the sampling process generating a new hybrid sampling method. This smart sampling optimizes the own metamodel generation using a relatively low number of sample points in a faster computation time reaching high fidelity prediction. In order to pursue such main contribution, other developments have also been achieved as following:

1. Appling the NSGA II genetic algorithm from Deb et al. (2002) on the sampling process coupled with AK becomes an original way to build the HC for sampling application. It allows to obtain a high fidelity metamodel in a relative faster process for large quantity of variables;

2. Using a random mesh covering all variables of the problem instead of the traditional cartesian mesh is also another original way to map the global design space. This random approach, which is based on the centroid of cloud of points computing, is applied on a sparse mesh divided in different zones of design space and covering a large quantity of variables with relative low computational cost;

3. Implementing very sensible objective functions on NSGAII genetic algorithm based on MSE and its derivative of MSE (dMSE) imposes a severe selective method to get best individuals that optimize the sampling process; and

4. Speeding-up the GA convergence by monitoring the change of centroid location of the generated populations. Such change implies on a faster convergence reaching only few generations. This method has been chosen instead of checking directly how the Pareto frontier is filled due to the the basic characteristic of the sampling process, which is the main core of the proposed algorithm.

1.4 Dissertation Outline

Considering the elements pointed out so far, it is possible to describe the dissertation outline starting by chapter 1, which has been composed by four parts: motivation, liter-

ature review, dissertation contribution and dissertation structure. The first part described the motivation to develop a more efficient method to generate a surrogate model than the conventional ones available in literature (item 1.1). The second part presented a literature review on this field and the contribution of the previous works in metamodel and optimization developments (item 1.2). The third part indicated the new developments and the major contributions of this dissertation (item 1.3). Finally, the last part presented the structure of this dissertation (item 1.4).

Chapter 2 begins presenting the theoretical description and applications of the Kriging Metamodel, its key factor based on the variogram estimator, and its uses in optimization process. The theory described in this chapter is addressed to the method named on this work as Conventional Kriging (item 2.1). Item 2.2 describes the spatial improvement method of Conventional Kriging based on the Adaptive Kriging theory. The item 2.3 introduces the theory of Genetic Algorithm and how it processes possible solutions to generate optimum ones. It is also highlighted that the Genetic Algorithm considered in this research is related to sampling method instead of an optimization tool. Finally, the item 2.4 describes the theory of mixing algorithms generating the Hybrid Code.

Chapter 3 describes the overall architecture of the proposed algorithm beginning by the description of the Conventional Kriging numerical implementation (item 3.1) and its adaptive method considering an enhancement of the first one (Adaptive Kriging – item 3.2). Then, the Genetic Algorithm is described – item 3.3, and finally the complete scheme of the Hybrid Code algorithm is presented (item 3.4).

In chapter 4, analytical benchmarks available in the literature are applied to the Hybrid Code for its validation. The mean error and time cost are the global efficiency criteria used for comparison with the Conventional and Adaptive Kriging methods. In item 4.1, the Branin function is used for validation and surrogate model comparisons. Also, a deeper analysis and comparison on the sampling process applied for this bechmark is evaluated in an example scenario. In Item 4.2, the Peak Family function with 2 variables and in item 4.3, the same function with 10 variables are used for comparison, also based on the global efficiency criteria.

Chapter 5 presents engineering problems modeled by finite element method and their structural responses applied on the Hybrid Code. Their results are also compared with the ones obtained by Conventional Kriging method demonstrating the efficiency achieved by

the new algorithm. In this context, item 5.1 presents the computations of a cantilever beam and item 5.2 presents a more complex model, which is based on a representation of an underwater explosion load on an orthotropic steel plate.

Finally, chapter 6 overviews new developments and results achieved by this research (item 6.1) and lists the topics that can be addressed for future researches (item 6.2).

CHAPTER 2

Theoretical description of Surrogate Models and Genetic Algorithm

The first aspect that is important to highlight is the metamodel definition. In strict sense, metamodel means model of a model. As seen in figure 1.2, the metamodel considers the physical problem as a black box taking care only of its inputs and outputs. In optimization problems, which are necessary to exploit several different combination of inputs searching for an optimum solution, the metamodel becomes a feasible and cheap way to be implemented. The challenge is to generate cost-effective metamodels with high fidelity on physical models (e.g. minimum ϵ from figure 1.2).

This is the core of this research with focus on sampling process. Next topics will present the theory of the typical Kriging method, based on DACE and named here as CK, its improvement on the sampling process and known in the literature as AK, an approach normally considered in optimization problems that is based on GA, and finally a new approach based on the mixing of such methods and called by this work as HC.

2.1 Conventional Kriging (CK)

The Kriging method was introduced by Danie G. Krige and formalized later by G. Matheron. Such method is based on the spatial predicting process based on statistical analysis of minimum mean squared error that starts from known samples. The response for an unknown location in the design space is estimated through the spatial variance analysis of the sample called by variogram or correlatiogram (Cressie, 1993). There are different types of kriging, such as simple, ordinary and universal.

Simple kriging is based on the premise that the mean μ is known and constant in the entire design space. Mathematically, it is described by the following equation:

$$y(x) = \mu + z(x), \tag{2.1}$$

where $z(x)$ is considered as the stochastic residuals of the estimative.

Ordinary kriging has the same general equation described by 2.1, but it is based on the premise that the unknown response is a function of a constant and unknown mean plus the stochastic residuals.

Finally, the universal kriging, which is considered by Sacks et al. (1989) and Cressie (1993) following the DACE approach, assumes the existence of a trend on the response that works as a non stationary function. In other words, the mean is variable and depends on its location, which can be considered as a linear combination of known functions. Considering that, based on Vlahopoulos et al. (2007), Cressie (1993) and Lophaven et al. (2002), such approach has the following general equation:

$$y(x_0) = F(\beta, x_0) + z(x_0), \tag{2.2}$$

Where $F(\beta, x_0)$ becomes the regression model that is normally modeled by some polynomial interpolation, while β is assumed as the regression parameters. It can be constant, linear or quadratic according to Lophaven et al. (2002). This aspect will be further described. For the stochastic residual, it can be modeled by different correlation models that will also be presented in next subsection.

This research is based on the last kriging approach (universal kriging), which is called by this dissertation as CK. The following topics focus on the description of such model.

2.1.1 Modeling of Universal Kriging

Based on Lophaven et al., 2002, the eq. 2.2 assumes the random process $z(x)$ with zero mean and covariance $R(x_s)$ for the known points and $r(x_0)$ for the unknown response. Also, it is assumed that the covariance R depends only on the distance between the sample points $h(x_s)$ (which responses are known) and not on the location of those points (x_s). Thus, considering m as the quantity of sample points, let the following regression matrix

F be described as:

$$F = [f(x_{s1})f(x_{s2}) \cdots f(x_{sm})] \tag{2.3}$$

And:

$$\hat{y}(x_0) = c^T Y, \tag{2.4}$$

Where, Y is the known response matrix and c is a matrix of coefficients. In this context, the error between the predicted response ($\hat{y}$) and the true value (y) can be derived as following:

$$\begin{aligned} &\hat{y}(x_0) - y(x_0) = \\ &= c^T Y - y(x_0) = \\ &= c^T(F\beta + Z(x_s)) - (f(x_0)^T\beta + z(x_0)) = \\ &= c^T Z(x_s) - z(x_0) + (F^T c - f(x_0))^T\beta \end{aligned} \tag{2.5}$$

Complying with unbiased condition:

$$\begin{aligned} &F^T c - f(x_0) = 0 \\ &f(x_0) = F^T c \end{aligned} \tag{2.6}$$

Considering the eq. 2.5 and the unbiased condition (eq. 2.6), the MSE named as δ can be derived as:

$$\begin{aligned} &\delta = E[(\hat{y}(x_0) - y(x_0))^2] = \\ &= E[(c^T Z(x_s) - z(x_0))^2] = \\ &= E[z(x_0)^2 + c^T Z(x_s) Z^T(x_s) c - 2c^T Z(x_s) z(x_0)] = \\ &= \sigma^2(1 + c^T R c - 2c^T r), \end{aligned} \tag{2.7}$$

Where σ^2 is the variance of z.

Minimizing δ as a function of c and respecting the unbiased condition (eq. 2.6), the lagrange multipliers (L, λ) can be applied in order to minimize δ as shown next:

$$
\begin{aligned}
L(c,\lambda) &= \sigma^2(1 + c^T R c - 2c^T r) - \lambda^T(F^T c - f) \\
\frac{\partial L}{\partial c}(c,\lambda) &= 2\sigma^2(Rc - r) - \lambda F
\end{aligned}
\tag{2.8}
$$

$$
\begin{bmatrix} R & F \\ F^T & 0 \end{bmatrix} \cdot \begin{bmatrix} c \\ -\frac{\lambda}{2\sigma^2} \end{bmatrix} = \begin{bmatrix} r \\ f \end{bmatrix} \tag{2.9}
$$

Considering $\tilde{\lambda}$ equal to $\frac{\lambda}{2\sigma^2}$, the following solution can be obtained:

$$
\begin{aligned}
\tilde{\lambda} &= (F^T R^{-1} F)^{-1}(F^T R^{-1} r - f) \\
c &= R^{-1}(r - F\tilde{\lambda})
\end{aligned}
\tag{2.10}
$$

Applying 2.10 into eq. 2.5:

$$
\begin{aligned}
&\hat{y}(x_0) = (r - F\tilde{\lambda})^T R^{-1} Y = \\
&= r^T R^{-1} Y - (F^T R^{-1} r - f)^T (F^T R^{-1} F)^{-1} F^T R^{-1} Y = \\
&= f(x_0)^T \beta' + r^T R^{-1}(Y - F\beta'),
\end{aligned}
\tag{2.11}
$$

Where β' is $(F^T R^{-1} F)^{-1} F^T R^{-1} Y$.

Also, the eq. 2.11 can be rewriten as shown next:

$$
\hat{y}(x_0) = f(x_0)^T \beta' + r R^T \gamma \tag{2.12}
$$

Where γ is the correlation matrix:

$$
R\gamma = Y - F\beta' \tag{2.13}
$$

Hence, the general equation of Universal Kriging stated by eq. 2.2 can be translated by the eq. 2.12. The possible regression models that compose F and the alternatives for the correlation function R in eq. 2.13 are described next.

2.1.2 Regression models

According to Lophaven et al. (2002), three different regression models can be used:

Constant

$$f_1(x) = 1 \tag{2.14}$$

Linear

$$\begin{aligned} &f_1(x) = 1 \\ &f_2(x) = x_1 \\ &\cdots \\ &f_{n+1}(x) = x_n \end{aligned} \tag{2.15}$$

Quadratic

$$\begin{aligned} &f_1(x) = 1 \\ &f_2(x) = x_1, \cdots, f_{n+1}(x) = x_n \\ &f_{n+2}(x) = x_1^2, \cdots, f_{2n+1}(x) = x_1 x_n \\ &f_{2n+2}(x) = x_2^2, \cdots, f_{3n}(x) = x_2 x_n \\ &\cdots \\ &f_p(x) = x_n^2, \end{aligned} \tag{2.16}$$

Where $p = \frac{1}{2}(n+1)(n+2)$ and n as the number of variables.

In this research, the quadratic regression model is the one considered for kriging equations.

2.1.3 Correlation models

The correlation function $R(\theta, h)$ relates the covariance of a given response with its spatial location. However, the spatial dependence prescribed by such correlation is based only on a defined vector h, which represents the distance among the points. The parameter θ is an internal parameter computed from the following optimization solution:

$$\theta \equiv |R|^{\frac{1}{n}} \sigma^2 \tag{2.17}$$

Then, according to Cressie (1993) and Lophaven et al. (2002), the correlation models are expressed by equations 2.18 to 2.23 and illustrated in fig. 2.1.

Linear function

$$R(\theta, h) = max\{0,\ 1 - \theta|h_i|\} \tag{2.18}$$

Cubic function

$$R(\theta, h) = 1 - 3\xi_i^2 + 2\xi_i^3, \xi_i = min\{1, \theta|h_i|\} \tag{2.19}$$

Spline function

$$\begin{aligned} &R(\theta, h) = 1 - 15\xi_i^2 + 30\xi_i^3,\ 0 \leq \xi_i\ \leq 0.2 \\ &R(\theta, h) = 1.25(1 - \xi_i)^3,\ 0.2 < \xi_i\ < 1 \\ &R(\theta, h) = 0,\ \xi_i\ \geq 1 \\ &\xi = \theta|h_i| \end{aligned} \tag{2.20}$$

Spherical function

$$R(\theta, h) = 1 - 1.5\xi_i + 0.5\xi_i^3, \xi = min\{1, \theta|h_i|\} \tag{2.21}$$

Exponential function

$$R(\theta, h) = exp(-\theta|h_i|) \tag{2.22}$$

Gaussian function

$$R(\theta, h) = exp(-\theta h_i^2) \tag{2.23}$$

Although there is no conclusive approach on which correlation model is better (Bossong et al., 1999, Ryu et al., 2002, and Cressie, 1993), the gaussian function was chosen in this research as it fits better the metamodels used in the validation of the method described in chapter 4.

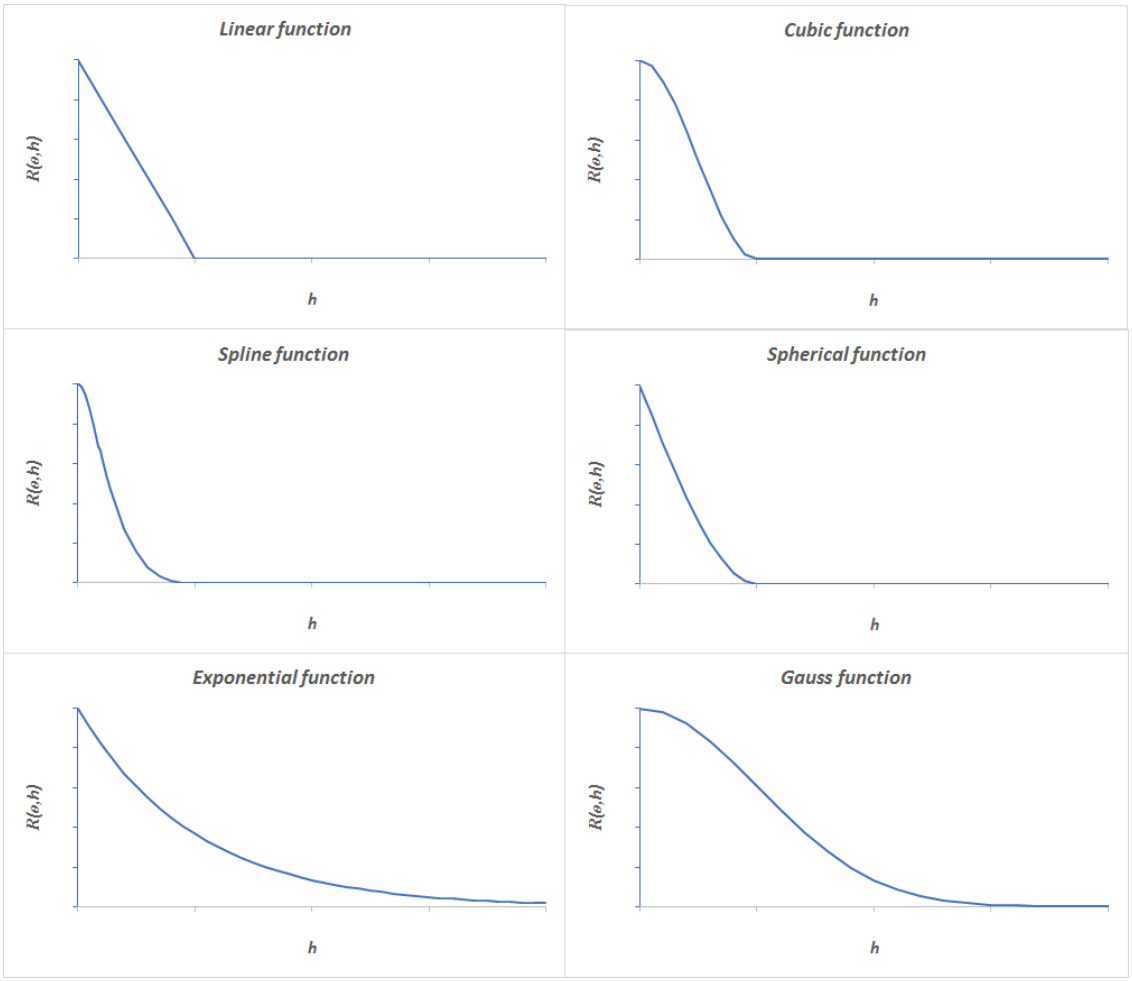


Figure 2.1: Example of different correlation functions

2.2 Adaptive Kriging (AK)

The key aspect of AK is to overcome the pure random approach of the Conventional Kriging through a sistematic sampling based on specific metrics that allow the improvement of final efficience on CK. There are different types of Adaptive Kriging methods that differ on the particular criterion for sampling. Bouhlel et al., 2017 considered the gradient of response for problems with large number of variables while Jones et al. (1998) assumed the MSE of the design space. H. Liu et al. (2017) considered by its turn the maximum error estimative. The common aspect on these approaches is the need to perform a first Conventional Kriging loop to generate a basis metamodel before applying the criterion to choose specific locations in the design space for new samplings.

It is highlighted that the adaptive approach considered in this research is based on the MSE criterion, which is determined throughout the design space. The maximum MSE provides an idea about the error of the metamodel obtained from the first CK loop. In this context, the overall MSE values are mapped in the design space considering the third group of points named as control points (cp) (remember that there are other two set of points named as sample and test points according to figure 1.2). The cp set consists on a mesh distributed along the entire design space with the same variables of the sample and test points. Details about control points mesh generation will be provided in chapter 3.

Therefore, after the first loop of Conventional Kriging, the large MSE locations are identified and grouped in a smaller set of points. Then, the cp located on these regions are ramdomly selected, added to the original sample and used to generate a new metamodel using the same Kriging approach. Such process reduces the pure random aspect of CK described by eq. 2.2. In this context, the new equation of the universal kriging for the second loop becomes:

$$y(x_0) = F(\beta, x_0^{MSE}) + z(x_0^{MSE}) \tag{2.24}$$

Although the CK performance can be improved by such adaptive approach, this method still faces some limitation due to the control points mesh size. If the number of cp is small, the precision on the high MSE location decreases. If the opposite occurs in relation to the cp mesh, the computation time is impacted due to the severe refinment of such mesh. Considering that, the need to exploit other complementary methods that are able to minimize the overall error of the estimation with reasonable computational time becomes justified. The complementary approach used in this research is the Genetic Algorithm, which is described in next topic. Chapter 4 will also show the comparative efficiency between the Adaptive Kriging and its coupling with Genetic algorithm.

2.3 Genetic Algorithm (GA)

Also known as Evolutionary method, the GA applied in this research has been based on the NSGAII code developed by Deb et al. (2002) and revisited by several other authors, such as Yusoff et al. (2011), Eby et al. (1998) and Baquela et al. (2019). In general aspect, this method mimics the mechanism of biologic species evolution through gene operations

searching for better individuals after some population generations. Normally, such algorithm is used for optimization process searching for best global or local solution, or even for best metamodels to predict the desired response. Nevertheless, for the context of this research, this method is applied on the improvement of a key step for metamodel generation based on the sampling process, similarly to the Adaptive Kriging approach.

Therefore, this topic will explain the main aspects of this process based on the NSGAII algorithm and adapted to this current implementation covering the objective functions, gene operation, and the elitism process.

2.3.1 Objective Functions

In GA, the sample points described in previous chapters about Conventional and Adaptive Kriging are called in this topic as individuals and their set is named as population. They are ranked according to best scores using some metrics defined by the objective functions. These functions are the first aspect that represents an adaptation from the original NSGAII to the current work.

The development performed has similar approach to Maki et al. (2012), which established a top and secondary objective functions. The initial population, which are the sample points (sp), are used to provide enough elements in Kriging process to compute the MSE values (eq. 2.7) for each individual of cp mesh. Taking into account these MSE values, the first and top objective function is defined for each individual of such population defined in the design space. The creation of cp mesh will be explained in detail on chapter 3.

Once the MSE value is computed for each cp, its respective $dMSE_v(x_i)$ derived from eq. 2.7 and related to each variable of the problem is also calculated. This parameter indicates the location of maximum or minimum MSE. Then, it corresponds to the second and third objective functions, or more if the problem has more than 2 variables. They work as a secondary objective function.

Therefore, considering both types of objective functions (MSE and dMSE), the total quantity of metrics becomes equivalent to the total number of variables plus one ($v + 1$):

$$
\begin{aligned}
&fo_1 = MSE \; or \; fo_1 = \delta \\
&fo_m = dMSE_v(x_i) \; or \; fo_m = \partial\delta/\partial x_i,
\end{aligned}
\tag{2.25}
$$

Where fo_m is the objective function for $m = \{2, 3, ..., v+1\}$ and $\delta = MSE$. Following the eq. 2.7, these objective functions can be rewritten as following (eq. 2.26):

$$
\begin{aligned}
&fo_1 = max\{MSE\} = max\{E[(\hat{y}(x_0) - y(x_0))^2]\} = max\{\sigma^2(1 + c^T Rc - 2c^T r)\} \\
&fo_m = min\{dMSE\} = min\{2\sigma^2((R - r)c' - r'c)\}
\end{aligned}
\tag{2.26}
$$

However, as mentioned before, the minimum dMSE can indicate two options: location close to the maximum MSE or close to the minimum MSE. The first possibility is trivial and desired. The second possibility can generate a problem due to the lack of contribution on a better sampling performance. The way to solve this issue is the Paretto frontier method already incorporated in NSGAII approach (Deb et al., 2002). After some generations, the convergence of GA is achieved through the reaching on the best solutions in terms of compromise between maximum MSE and minimum dMSE for each population. This aspect will better described in the Elitism Process. Moreover, in addition to that theoretical process, for the current HC algorithm, an additional numerical feature has been considered in this process, which is based on a severe selection criteria applied on the objective functions. This feature will be detailed in the topic 3.3.1.

2.3.2 Gene Operations

Once collected the metrics from each individuals of a given population, they are submitted to two gene operations known as crossover and mutation. In fact, according to Deb et al. (2002), half population are chosen as parents and they are submitted on such process: 90% of this set takes part on the crossover and the 10% complementary part participates on the mutation process (figure 2.2).

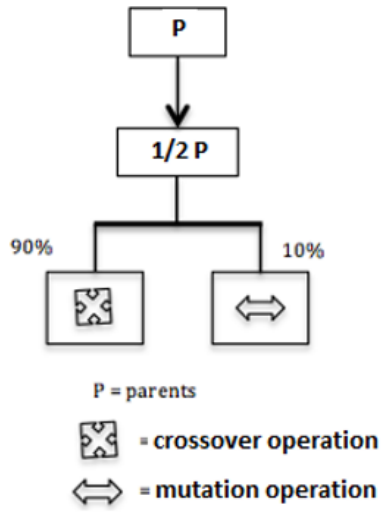


Figure 2.2: Genetic operations performed by NSGAII (Deb et al., 2002)

Basically, according to Deb et al. (1995), the crossover operation conserves the characteristics of the parents to generate (or select) new individuals from the design space. Eq. 2.27 shows how this crossing can be processed.

$$
\begin{aligned}
b_j &= (2u_j)^{\frac{1}{\rho+1}}, for\ u_j \leq 0.5\ and\ \rho = 20 \\
b_j &= \left(\frac{1}{2(1-u_j)}\right)^{\frac{1}{\rho+1}}, for\ u_j > 0.5\ and\ \rho = 20 \\
children_{1j} &= \frac{1}{2}\left((1+b)parent_{1j} + (1-b)parent_{2j}\right) \\
children_{2j} &= \frac{1}{2}\left((1-b)parent_{1j} + (1+b)parent_{2j}\right),
\end{aligned}
\tag{2.27}
$$

Where $j \in \{1, ..., v\}$ and indicates a traceability with the respective parents. v corresponds to the number of variables of the design space, ρ value is arbitrarily given by Deb et al. (2002) and u_j is a random number between 0 and 1.

Inversally to the crossover, the mutation needs only one single parent and its expression is presented next as described in Deb et al. (1995):

$$\Delta_j = (2u_j)^{\frac{1}{\kappa+1}}, for\ u_j < 0.5\ and\ \kappa = 20$$
$$\Delta_j = 1 - (2(1-u_j))^{\frac{1}{\kappa+1}}, for\ u_j \geq 0.5\ and\ \kappa = 20 \quad (2.28)$$
$$children_{3j} = parent_j + \Delta_j,$$

Where $j \in \{1, ..., v\}$ and indicates a traceability with the respective parents. v corresponds to the number of variables of the design space, κ is arbitrarily chosen by Deb et al. (2002) and u_j is a random number between 0 and 1.

Note that each children obtained by the crossover or even by the mutation represents a new particular location x_i in the design space for the context of this research and new loop of objective functions computation need to be performed.

2.3.3 Elitism Process

In this process, each individual of the population (parents + children) is scored considering the overall performance achieved by the objective functions and forming a group of individuals named as *non-dominated solutions* (smallest rank). This comparative process collects optimum data related to the objective functions using the Paretto frontier. By doing that, each rank is computed allowing to choose the best individuals with better scores as part of the next generation. NSGAII has a particular feature based on the crowding distance considering the obtained metrics from the objective functions. This aspect imposes some distance between the scores placed at the Paretto frontier, which implies on the adding of some diversity on the next population. Such distance is normalized by the maximum distance computed from each rank. After one single loop, the final set of chromossomes configuration considering N individuals for a given population takes the form presented in figure 2.3.

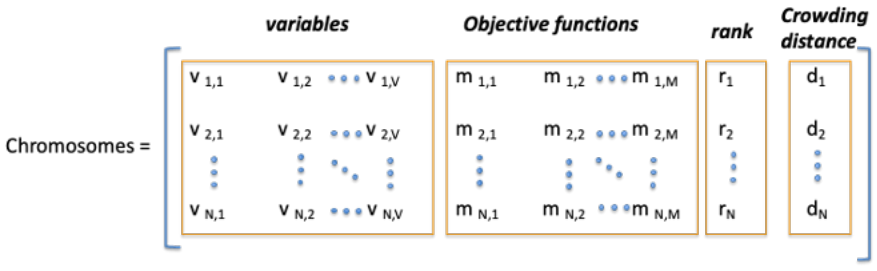


Figure 2.3: Chromossomes configuration computed by NSGAII (Deb et al., 2002)

Through these generation loops, several new populations can be created searching for even better scores from the objective functions. This process normally stops when its convergence is reached. In the proposed algorithm, the method to define and to track such convergence will be explained in chapter 3.

Figure 2.4 illustrates the relationship among the objective functions and how the optimum individuals are identified in relation to the other ones. NSGAII identify the Paretto frontier and the best individual can be collected from this set. The numeric method about that specific selection is also explained in detail on chapter 3.

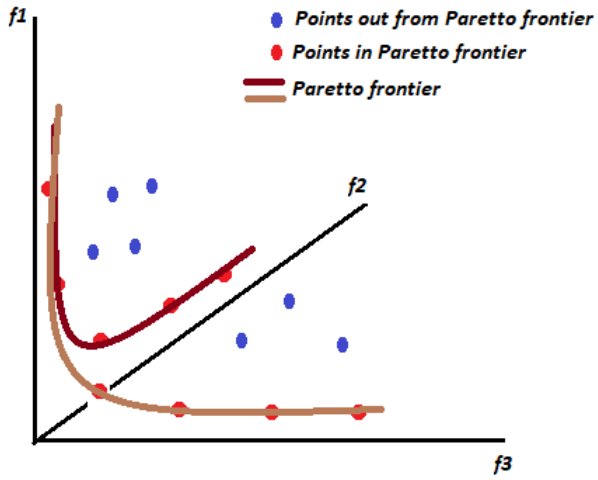


Figure 2.4: Theoretical Paretto frontier obtained from the relationship among involved variables and objective functions. In this graph, only three objective functions (problem with two variables) are presented for easy visualization.

2.4 Hybrid Methodology

Finally, as expected, the hybrid methodology joins all these approaches in a rational method taking profit of the best advantages of each one in a single algorithm and achieving the best optimization performance. Some researches that follows such method are Bouhlel et al. (2016), Baquela et al. (2019), K. Lee et al. (2019), Menz et al. (2019), Kleijnen (2017), Dellino et al. (2009), Jeong et al. (2005), Y. Liu et al. (2014), Zhu et al. (2013) and Tarek et al. (2006). In particular, the work done by Baquela et al. (2019) also tries to combine the Kriging and the genetic algorithm methods. However, the key difference is that the method proposed by Baquela et al. (2019) searches for the optimum metamodel

through the computation of several ones using the GA and selecting the best ones from that set. References Y. Liu et al. (2014) and Zhu et al. (2013) also have some similarity, as it uses the GA application to update the metamodels achieving better performance. Differently from that works, the current research focuses on the straightforward adaptation for this hybrid method on the sampling process inputting an adaptive process in the overall code named here as Hybrid Code and generating a single metamodel with high fidelity and relative low computational cost.

Hence, the proposed approach works with three processing layers getting a single and relative efficient metamodel as shown in figure 2.5.

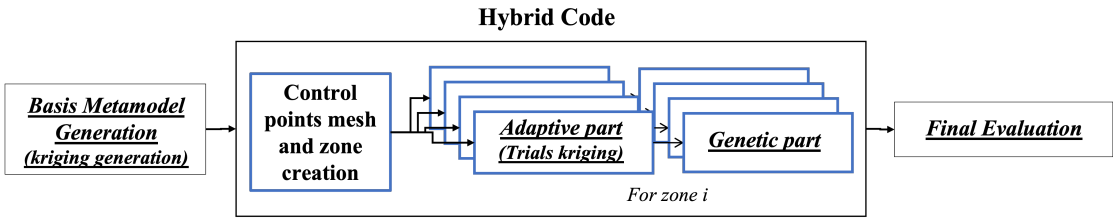


Figure 2.5: Global Hybrid Code flowchart.

Considering the Universal kriging or DACE as the starting task (Conventional Kriging), the new cp mesh covering the entire design space in i zones takes into account all v variables works as the first processing HC layer. Then, the Adaptive kriging is implemented as second layer considering MSE as criterion selection. Finally, the Genetic Algorithm works as the third layer with $v + 1$ objective functions (MSE and $dMSE_v$). This process is able to reach even better new sample points (individuals) through gene operations from cp mesh and their children. Next chapter will explain how these layers are computationally implemented and how they communicate with each other.

CHAPTER 3

Numerical Implementation

This chapter describes how each method is individually implemented and how they are linked together in the Hybrid Code. All algorithms have been written in MATLAB and for engineering applications, two finite element commercial softwares have been used: ANSYS and ABAQUS. Moreover, the machine used to process all of them is intel(R) core(TM) i7-2720QM 2.20 GHz and 8Gb for RAM memory.

Before continuing in the numeric description of each method, two main aspects have been considered in all algorithm implementations and described in this section: the test points and one metric chosen to evaluate the performace of the obtained metamodel, which has been based on the normalized mean error.

- Test points

For all generated metamodel, it is necessary to evaluate its prediction performance taking into account new points sampled from the design space. These points are called *test points* and they are sampled by RND approach employed in all the performed simulations (see figure 1.2). For simplicity, 200 points (n_t) are used in all of these cases.

- Normalized mean error

The metric chosen to evaluate the fidelity of each metamodel when compared to the real function is the mean error $(\bar{\epsilon})$ computed from the difference between the response prediction of the test point and its actual response, normalized by the maximum response known from the set of test points (y_t^{max}). Therefore, such evaluation metric is computed as following:

$$\bar{\epsilon} = \frac{1}{n_t} \sum_{1}^{n_t} \frac{|\hat{y}_t - y_t|}{y_t^{max}}, \tag{3.1}$$

An additional metric used for global performance evaluation of the metamodel is the consumed time related to its computing (e.g. learning stage) and trial performance (e.g. trial stage) processes considering the due correlation with the mean error achieved. Both metrics are presented in chapters 4 and 5.

3.1 Conventional Kriging

The first step of this algorithm is to collect the input data x_i and y_i from a given problem. These points are sampled using the RND method. As mentioned before, this part is known as Learning Stage and important parameters are obtained from this set of points, such as the correlogram, regression model and internal optimum parameter θ as described by eq. 2.17. Once these information are obtained and used to generate the metamodel, the test points are sampled and submitted to the metamodel for the final evaluation. This part is known as Trial stage.

It is also important to highlight that, as mentioned in chapter 2, the interpolation model and correlogram are quadratic and Gauss function, respectively. The following algorithm 1 presents an overview of the steps taken to get the metamodel and the measurement of its performance. This algorithm is the basis of HC, which is described in section 3.4.

Algorithm 1: Conventional Kriging (CK) algortihm

Result: Metamodel obtained from CK and its mean error performance

Begin
- **Learning Stage** *routine*:
  - sampling x_i and y_i through RND method
  - searching the minimum θ parameter for the correlogram function
  - Getting the metamodel from the interpolation and correlogram functions
- **Trial Stage** *routine*:
  - **for** *i=1 to t* **do**
    - Applying the metamodel in each test point obtaining the $\hat{y}_t$ and MSE_t
  - computing the maximum response of the test points, y_t^{max}
  - computing the normalized error for each test point, $\epsilon_t = \frac{|\hat{y}_t - y_t|}{y_t^{max}}$
  - computing the mean error performance, $\bar{\epsilon}$, and the total time consumed, T

3.2 Adaptive Kriging

The Adaptive Kriging, as described in Chapter 2, aims to provide a new sampling in the design space from the results obtained in previous Conventional Kriging metamodel in order to improve the accuracy of this estimator. In this context, two important features are implemented in such algorithm: the control points mesh and its relation with specific criteria (high MSE values). Both features are used to select new areas in the design space to sample new points. They are described hereafter.

3.2.1 Control points mesh

In order to estimate the accuracy of the estimator in the entire design space, it is necessary to generate a cloud of points called *control points*.

The generation of the control points mesh for a cartesian distribution is straightforward. The control points are equidistributed along the coordinates. The MSE values at these points are mapped and the positions with the highest MSE (as seen in the red color regions of figure 3.1) are the potential candidate areas for new additional points.

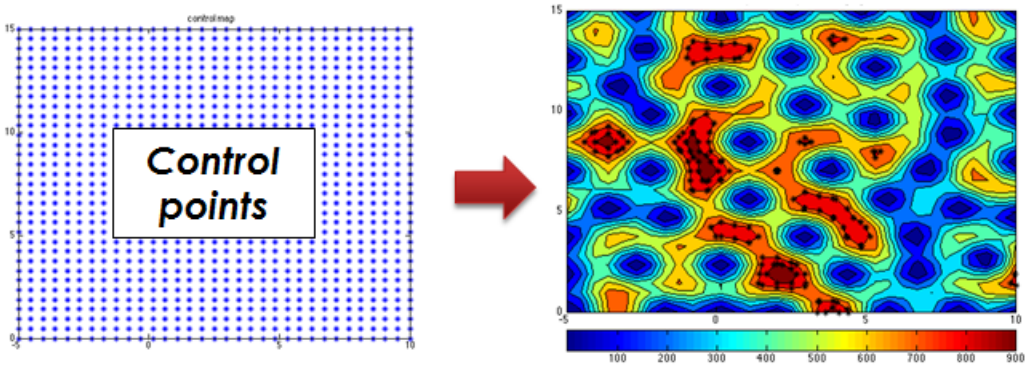


Figure 3.1: Cartesian mesh of control points for 2 variables problem generating MSE map of the design space.

However, for problems with more variables, e.g. 3, 5, 10, ... , the computational time involved to generate such mesh grows exponentially, which is proportional to the number of points raised to the power of quantity of variables (N^v), impacting the computational time.

The solution to overcome this obstacle is using a random or nearly random control point distribution instead of equidistant coordinates. The method adopted is based on k-means clustering (eq. 3.2), which means that the algorithm separates the cloud of points into several different subclouds, called by *zones*. They are located in subspace regions of the design space considering the euclidean distance of the points (x_i) to the zone centroids (c_i). Such computation is performed in several iterations until its convergence (figure 3.2). For this current work, 300 maximum iterations have been assumed as limit. This technique is based on Lloyd (1982) and Arthur et al. (2007) and chosen for this mesh process due to the influence of the distribution of the points on the topology of the design space. It has been implemented in MATLAB according to reference *MATLAB Manual - k-means Clustering* (2018).

$$
\begin{aligned}
d(x_i, c_i) &= (x_i - c_i)(x_i - c_i)' \\
c_i &= \frac{\sum x_i}{n_{points}}
\end{aligned}
\tag{3.2}
$$

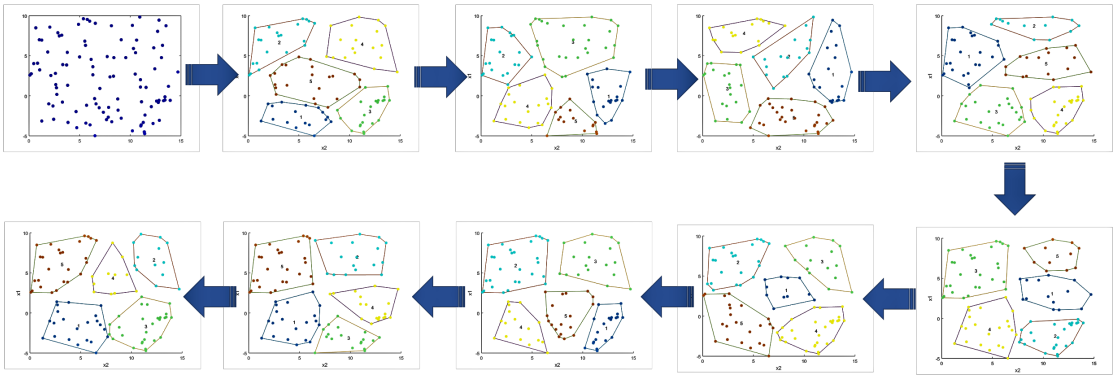


Figure 3.2: Kmeans iterations for zones convergence.

In this context, each zone becomes a small part of the complete domain that is filled by one cluster of control points considering all variables of the problem. Figure 3.3 illustrates the resulting control point mesh. The performance of this type of mesh for higher number of variable problems is proportional to the number of points times the quantity of variables ($N \cdot v$), being much more efficient than the cartesian mesh. This meshing methodology has been directly implemented in the code.

3.2.2 Control points mesh and MSE criterion

The MSE criterion for each zone in the control points mesh obeys the following rules:

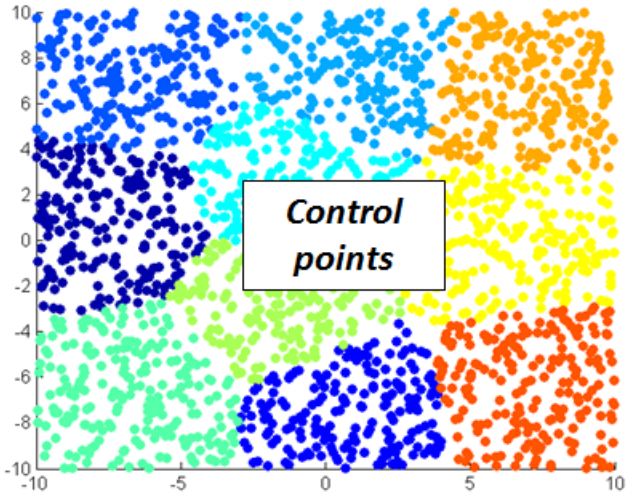


Figure 3.3: Random control points mesh based on subspace regions - each color represents one zone.

- For variables less and equal to 4

It is assumed an initial sample of 3000 control points for each zone. Applying the MSE criterion, this quantity reduces to 10 control points at each zone with the highest MSE values.

- For variables larger than 4

It is assumed an initial sample of 7000 control points for each zone, which is reduced by the MSE criterion to 50 control points at zone with the highest MSE values.

Following this approach, problems with large number of variables can be managed by this Adaptive Kriging algorithm without huge impact on the computational time. For a smaller number of variables, the number of control points is minimized while preserving the overall performance of the predictor.

3.2.3 Algorithm overview

Applying the concepts presented in the previous items, the Adaptive Kriging is described in the following algorithm 2:

Algorithm 2: Adaptive Kriging (AK) algortihm

Begin
  Conventional Kriging *Algorithm 1*:
    sampling x_i and y_i through RND method;
    searching the minimum θ parameter for the correlogram function;
    Getting the metamodel from the interpolation and correlogram functions;

  Random mesh *routine*:
    Generating the random control points mesh;
    Running the *kmeans* method identifing zones in the control points mesh;

  Adaptive Kriging *routine*:
    for *i=1 to cp* **do**
      Applying the metamodel in each control point for each zone i obtaining the MSE_{cp};
    Identifying the maximum MSE locations for each zone (x_{cp});
    Computing the real response of these maximum MSE locations (y_{cp});
    Generating a new Kriging Metamodel considering new sample points ($x_i + x_{cp}$; $y_i + y_{cp}$);

  Trial Stage *routine*:
    for *i=1 to t* **do**
      Applying the metamodel in each test point obtaining the $\hat{y}_t$;
    computing the maximum response of the test points, y_t^{max};
    computing the normalized error for each test point, $\epsilon_t = \frac{|\hat{y}_t - y_t|}{y_t^{max}}$;
    computing the mean error performance, $\bar{\epsilon}$, and the total time consumed, T;

3.3 Genetic Algorithm

The NSGA II Deb et al. (2002) is the Genetic Algorithm used in this research. Few adaptations have been included in order to adjust it for the sampling purpose. The algorithm overview and its main aspects are presented next.

3.3.1 Objective functions - MSE prioritization

According to topic 2.3.1, some additional particularities have been complemented in the numeric implementation in order to improve the efficiency of the algorithm. In fact, both objective functions (maximum MSE and minimum dMSE - eq. 2.26) used simultaneously may produce a drawback. If low dMSE is tracked, individuals placed at high and low

MSE are selected. The same happens for high MSE location, where individuals can have high and low dMSE.

In order to overcome this issue, besides of the normal Pareto frontier process considered by NSGAII, the maximum MSE has been received a priority in relation to the lowest dMSE. It means that for each population in each zone, the individuals that have MSE values lower than the maximum one have their evaluation metric arbitarily changed: $1e-11$ for MSE and $1e10$ for dMSE. As result, such procedure selects only one individual that corresponds to the maximum MSE as the best for a given population and for each generation in each zone. The dMSE lowest values collected are only the ones correlated with the maximum MSE individuals selected and then analysed together for best selection individuals. This severe selective process forces to create a group of individuals composed by only the best representatives for each generation and allows to the second objective function to be more effective in further population generations. Thus, performing such process, all objective functions are computed from each individual allowing to compose each chromossome following Deb et al. (2002). After some generations, until convergence, these individuals take part on the Pareto frontier.

3.3.2 Convergence rate

This particular aspect has been discussed in the literature such as Ming et al. (2006), Abu-Lebdeh et al. (2002), He et al. (2016), Louis et al. (1992), Sahin et al. (2010) and Stark et al. (2002). These references focus in the optimization process, especially in the Pareto frontier process, aiming to overcome the challenge to identify the global instead of the local optimum solution.

For the sampling procedure adopted in this research, it is necessary to identify when the best solution is achieved after some generations. This issue is solved by analysing the location of the best individual selected from each generation obtained through the procedure described in the previous topic (3.3.1). In summary, the convergence solution implemented is based on the geographic location in the design space. It is established when the centroid of each cloud of individuals obtained after each generation and normalized by complete range of the domain changes less than 0.1% in comparison to the centroid achieved by previous generation. Figure 3.4 illustrates this process. As consequence, convergence of this GA is relatively faster in comparison with traditional genetic algorithms applied in optimization problems, as it needs only few generations in the optimum search.

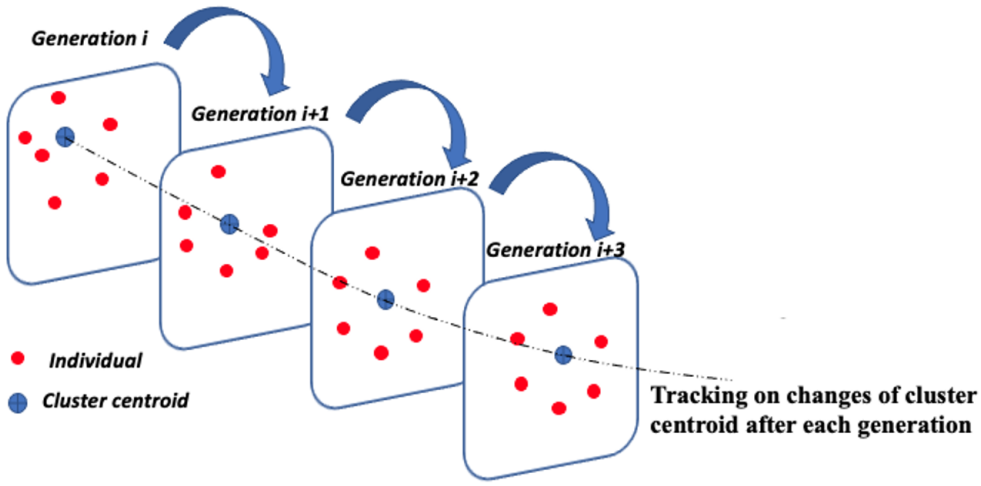


Figure 3.4: Convergence process adopted for Genetic Algoritm based on the geographic cloud centroid location.

3.3.3 Limit Boundaries of Design Space

Another relevant aspect added on NSGA II algorithm is the enforcement of boundaries in the design space. The gene operations described by the topic 2.3.2 may result in individuals located out of the design space. If such result is obtained, these individuals are neglected, not taking part on the Pareto frontier evaluation.

3.3.4 Final selection from Pareto Frontier

Once the convergence has been reached and the Pareto frontier has been evaluated, the best individuals of each zone configure a set that joints both maximum MSE and minimum dMSE characteristics. From this set, the best individual is randomly chosen in order to be added as a new point in the original sample points set.

3.3.5 Algorithm overview

After the particular descriptions previously presented, it is adequate to show the overview of the whole algorithm. As already mentioned, it is fully based on Deb et al. (2002) with specific adjustments described before. In order to make easier its understanding, the algorithm can be split in two parts: Introductory and Evolutionary part. The first part is focused on the first generation and aims to compute the objective functions for each individual from the population 0. The second part is related to computation of next generations, which includes objective functions, ranking, placing individuals on the Pareto frontier, selecting

best individual of a given generation, and convergence monitoring.

Algorithm 3: Genetic Algorithm (GA) based on Deb et al. (2002) - Introductory part

Result: Computing the Objective Functions MSE and dMSE for the first generation of individuals.

Begin
- collecting the x_i by each individual for the first population;
- computing the objective functions (MSE and $dMSE_v$) for each individual;
- setting the chromossome composition for each individual;
- performing the elitism process through the rank of each individual for the first population based on the fitness of their chromossome;
- Assembling the final chromosome format;

Algorithm 4: Genetic Algorithm (GA) based on Deb et al. (2002) - Evolutionary part

Result: Obtaining the best individual after some generations.

Begin
- **LOOP** *N=1 to number of generations (ngen)*
  - selecting the best individuals of given population to generate their children;
  - performing the gene operations on the parents in order to get the children;
  - computing the overall population as parents + children;
  - Making sure that all individuals are inside of design space;
  - ranking all individuals;
  - sorting the individuals based on their rank;
  - computing the individual crowding distances;
  - grouping the best individuals in a pool through Pareto frontier;
  - computing the centroid of best individuals cloud in order to check the GA convergence;
  - `// To be continued ...`

```
18 Main Loop
19     LOOP continuation N=1 to ngen
20         if convergence achieved then
21             selecting the best individual from the Pareto frontier pool for a given
                generation and consider as output;
22             stop the loop;
23         else
24             selecting the best individuals from the Pareto frontier pool for a given
                generation and create next generation with new loop;
```

3.4 Hybrid Code

As already explained in topic 2.4 and illustrated on figure 2.5, HC combines all previous algorithms in a single code searching for the best sampling. This process aims the achieving of a better metamodel able to generate a more accurate prediction. Basically, the first part is composed by a random mesh able to cover the entire design space including all variables involved on the problem (topic 3.2.1). In the second part, the AK described in topic 3.2 takes place performing the first candidates selection from the control points mesh. In sequence, such pre-selection becomes input to be considered by GA described in topic 3.3, which performs severe selection for each population generation achieving the best point for each zone of the design space that complies with both objective function types described in topic 2.3.1 and eq. 2.26. Figures 3.5 and 3.6 show a detailed flowchart of the overall code.

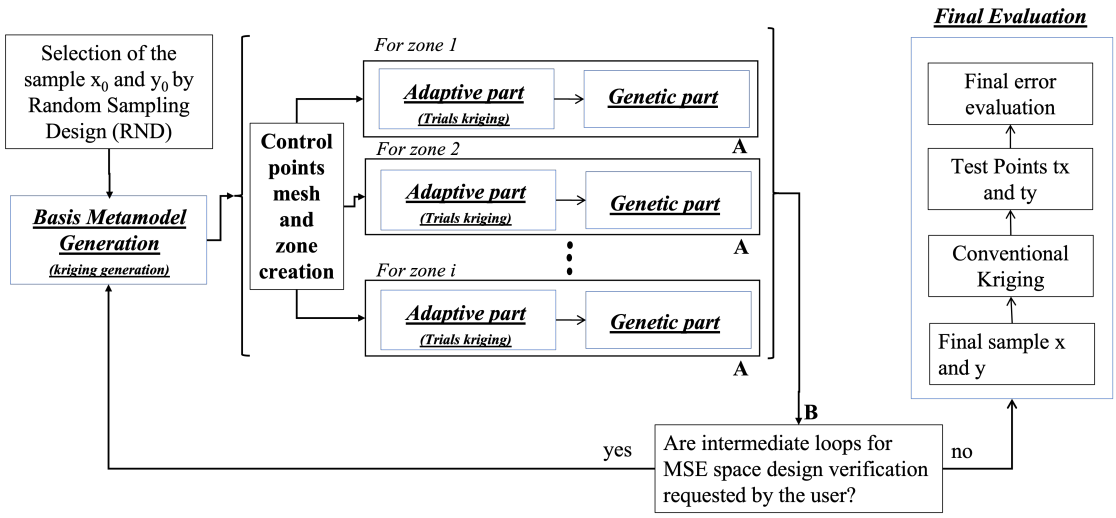


Figure 3.5: Flowchart of the Hybrid Code - Overview.

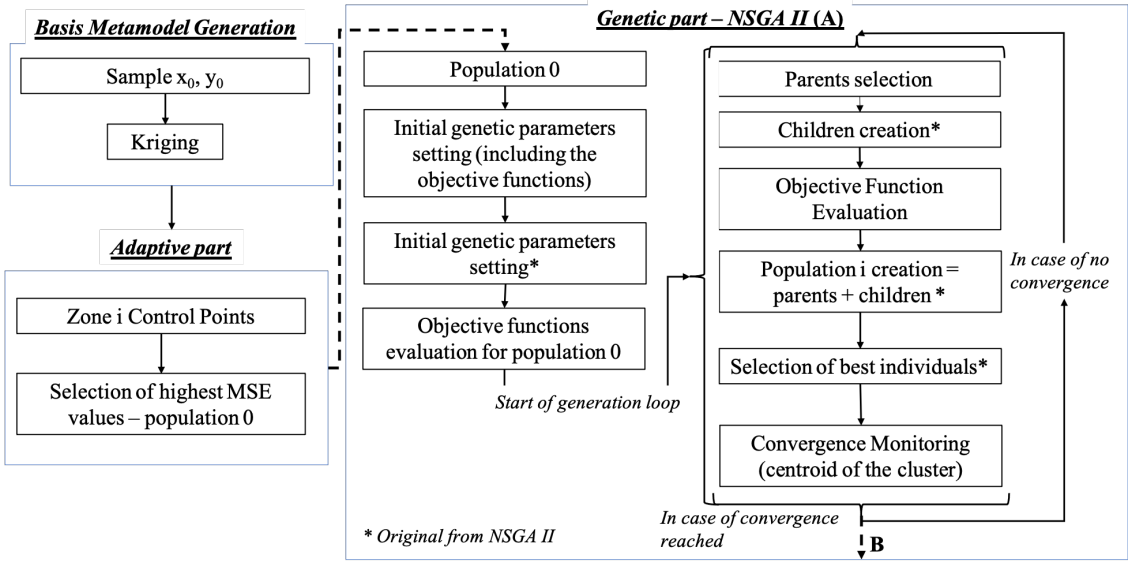


Figure 3.6: Flowchart of the Hybrid Code - Adaptive Kriging and Genetic parts details.

In the first part, once the initial sample is selected by RND (x_0 and y_0), the control points mesh is generated as shown in figure 3.3. Important to mention that the quantity of zones to be considered for the control points mesh is an input by the user. In summary, as it is presented in this chapter and in the chapter 2, the number of zones will depend on the quantity of points desired to be added on the original sample by the HC since each zone outputs one new point to be added on the sample. For example, if it is desirable to add 10 points on the sample (*named as N_a on the algorithm*), 10 zones are created in the mesh. Then, the first part is the generation of the first kriging metamodel based on the

known sample points (*Basis Metamodel Generation*). After that stage, the random mesh takes place dividing the design space into zones. From the AK in *Adaptive part*, the MSE for control points in each zone are computed. It selects the highest MSE locations for each zone following the scheme described by the topic 3.2.2 and generates the first set of points (*population 0*) to be the input for the GA. In the *Genetic part*, the code runs for each zone as described in topic 3.3.

Another relevant aspect is the possibility to update the sample considering intermediate MSE evaluations. Such MSE updating consists on the division of desired points to be added by HC into groups. After selecting the points of the first group, the *Basis Metamodel* is updated and a new MSE map is evaluated in the whole control points mesh. From these new results, new loop of AK and GA run for all zones and selecting the next group of points to be added on the original sample. It repeats until completing the quantity of desired points to be added in the original sample. This process is a tentative to improve the final performance of HC and has been called in this dissertation as *Basis Metamodel loop*. However, one important aspect needs to be observed: spliting the additional points in group of loops means that the cp mesh will be splitted in the same fashion, but still covering the overall design space. In this sense, the same geographic zones are considered in all of these loops. For example, if the idea is to add 20 points in the original sample considering 2 loops, 10 zones covering the whole domain with 10 new additional points are created in both scenarios. This process indicates if the quantity of zones also has some influence in the final performance. As it is shown in the chapters 4 and 5, this additional process does not bring a significant improvement considering the balance between the computational time and mean error prediction.

The Final Evaluation takes place with the complete sample, which is used to generate the final Kriging metamodel followed by its performance evaluation on test points. In order to follow the same format of other methods presented, the HC is described by the following algorithm 5, which is validated in chapter 4 through its comparison with Conventional and Adaptive Kriging.

Algorithm 5: Hybrid Code (HC) algorithm

```
Begin
    Conventional Kriging Algorithm 1:
        sampling x_i and y_i through RND method
        searching the minimum θ parameter for the correlogram function
        Getting the metamodel from the interpolation and correlogram functions

    Random mesh Algorithm 2:
        Considering number of points to be added number of points to be
          added (N_a) for the original sample
        Generating the random control points mesh
        Running the kmeans method identifing N_a zones in the control points mesh

    Adaptive Kriging Algorithm 2:
        for i=1 to cp do
            Applying the metamodel in each control point for each zone i obtaining
              the MSE_cp
        Identifying the maximum MSE locations from each zone (x_cp) in order to
          get the population 0

    Genetic Algorithm Introductory part - Algorithm 3:
        collecting the x_i by each individual for the first population
        computing the objective functions (MSE and dMSE_v) for each individual
        setting the chromossome composition for each individual
        performing the elitism process through the rank of each individual for the
          first population based on the fitness of their chromossome
        Assembling the final chromosome format

    Genetic Algorithm Evolutionary part - Algorithm 4:
        selecting the best individuals of given population to generate their children
        performing the gene operations on the parents in order to get the children
        computing the overall population as parents + children
        Making sure that all individuals are inside of design space
        ranking all individuals
        sorting the individuals based on their rank
        computing the individual crowding distances
        Selecting the best individuals through Pareto frontier to make part on the
          next generation
        Computing the centroid of best individuals cloud to check the GA
          convergence
    // To be continued ...
```

Main Loop `// continuing...`

- **Genetic Algorithm** *Evolutionary part - Algorithm 4*:
  - **if** *convergence achieved* **then**
    - Selecting the best individual and consider it as output
    - Stop the loop
  - **else**
    - Creating the next generation and perform a new loop

- **Basis Metamodel loop** *updating MSE map*:
  - **if** *YES for New Basis Metamodel loop* **then**
    - Reinitializing the overall run considering new MSE control points mesh values

- **Final Evaluation** *Trial Stage*:
  - **else**
    - Jointing the final set of new points to the original sample
    - Generating the final Kriging metamodel
    - Applying the final Kriging metamodel on the test points
    - computing the mean error performance, $\bar{\epsilon}$, and the total time consumed, T

CHAPTER 4

Hybrid Code Validation

In this chapter, the HC validation is demonstrated using three benchmark analytical problems: Branin function for 2 variables and Product Peak Integrand Family function for 2 and 10 variables according to Surjanovic et al. (2020). For each of them, the performance of HC is presented and compared with CK. In addition, a sensitive analysis also compares the performance of HC with the code operating only with AK. In particular for such analysis, the pure AK considered also the points with low dMSE in complement with the ones with high MSE in order to be consistent when compared with HC. In that case, once these points are grouped, only one is randomly selected as a representative of the best ones for each cp mesh zone, similarly as it occurs in the HC process.

4.1 Branin Function

4.1.1 Function description

According to Surjanovic et al. (2020), Branin function has two variables and three global minima. Its general definition is expressed by the following equation:

$$f(x) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1-t) \tag{4.1}$$

Considering the constants as $a = 1$, $b = \frac{5}{4\pi^2}$, $c = \frac{5}{\pi}$, $r = 6$, $s = 10$, and $t = \frac{1}{8\pi}$, in the range $-5 \leq x_1 \leq 10$ and $0 \leq x_2 \leq 15$, the formulation and shape of Branin function becomes as indicated in eq. 4.2 and presented in figure 4.1:

$$y = \left(x_2 - \frac{5}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10. \tag{4.2}$$

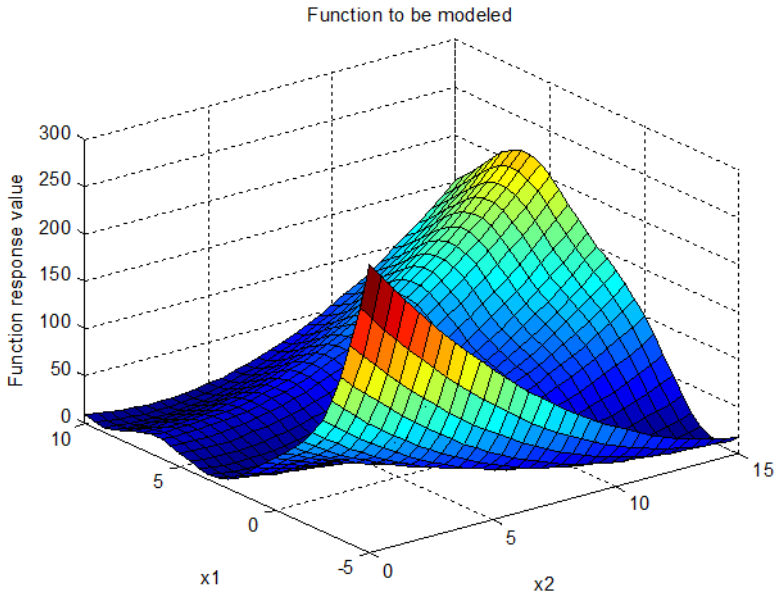


Figure 4.1: Branin function according to Surjanovic et al. (2020).

4.1.2 Sampling Process

One key aspect described in chapter 3 is the partition of design space into zones. Thus, following the figure 3.3, the domain of Branin function has been divided in zones, according to the different collors shown in figure 4.2.

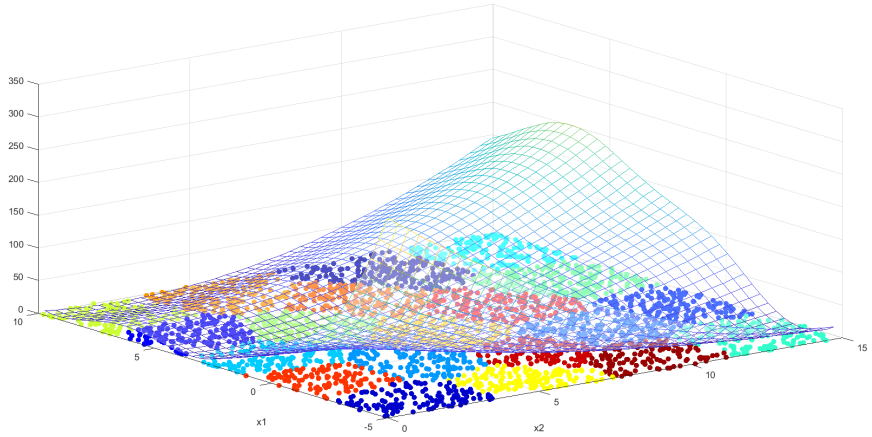


Figure 4.2: Branin function domain divided by zones.

The sampling process applying adaptive and genetic algorithms is exemplified by the

use of a Branin Function considering 20 initial sample points addressed for Conventional Kriging. The use of HC supplies another 20 additional sample points.

In the first step, the initial sample points are obtained by RND procedure. These points are highlighted in green in figure 4.3 and are in zones 3, 5, 6, 7, 10, 12, 14, 15, 16, 17 and 20.

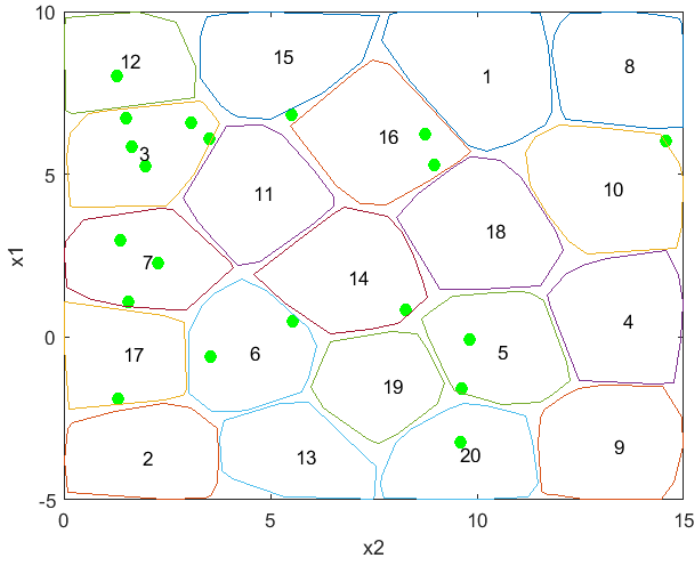


Figure 4.3: Zone location of the initial sample obtained by pure RND.

The genetic process applied on each zone follows the algorithm 5 described in chapter 3. The generations and the search for Pareto frontier are shown in Table 4.1 for each objective function. In reality, such table presents four figures applicable for each zone. On north, two figures are based on the objective function results. Considering the fact that the Branin function has only two variables, three objective functions are defined in the GA: MSE, derivative of MSE related to x_1 (dMSE1) and derivative of MSE related to x_2 (dMSE2). They have been plotted on these figures with the MSE as the y-axis and dMSE1 and dMSE2 as x-axis. The values obtained from the last generation are identified by filled marks. The values of each objective function correspond to the results before the change imposed by *MSE prioritization* described in the topic 3.3.1. On the southwest, the cp mesh of respective zone is identified over the Branin function for easy visualization. Finally, on the southeast figure, the final sampling result obtained from such processing is showed considering red points as identification of new samples reached by HC. In this figure, the green points are the initial sample obtained from RND, which are the same ones presented in figure 4.3, and

the light blue circles are the overall cp mesh. For each red point sampled, the respective zone indicated in figure 4.3 is also presented.

Table 4.1: Generations convergence - Pareto frontier on each zone

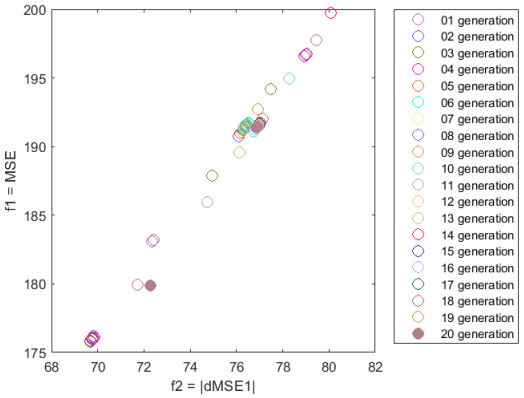
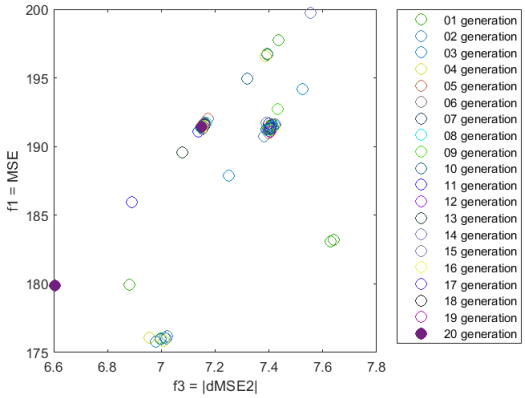
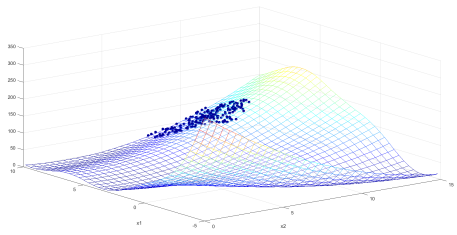
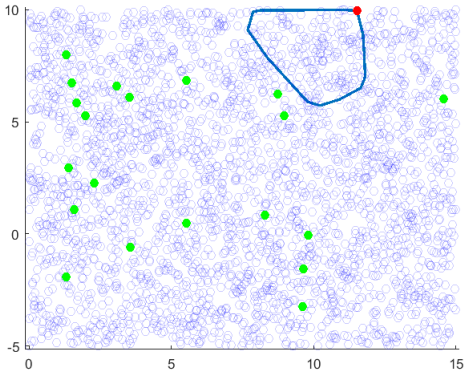
Zone 1	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

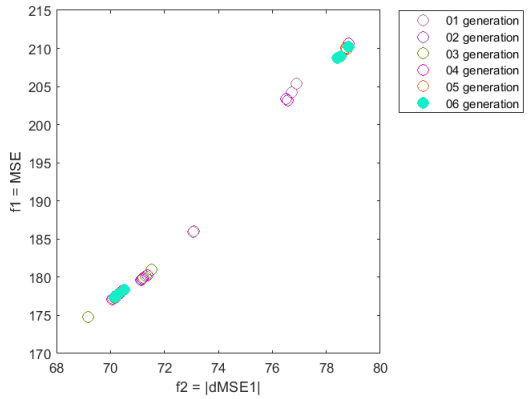
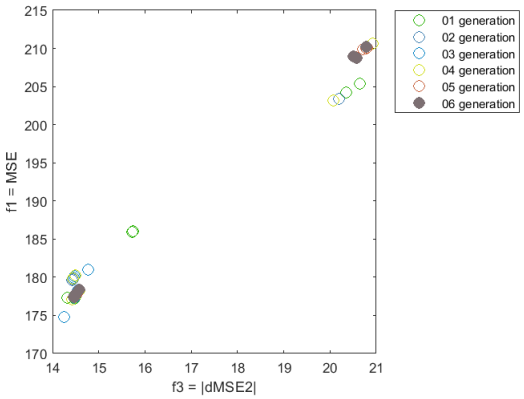
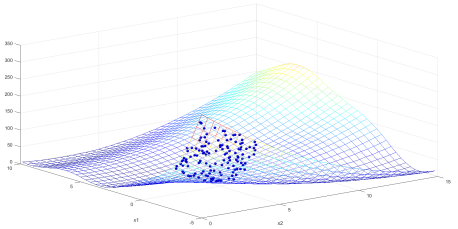
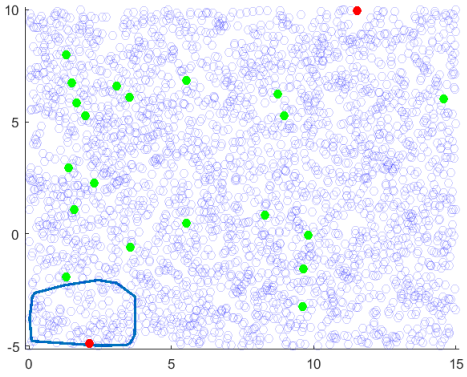
Zone 2	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

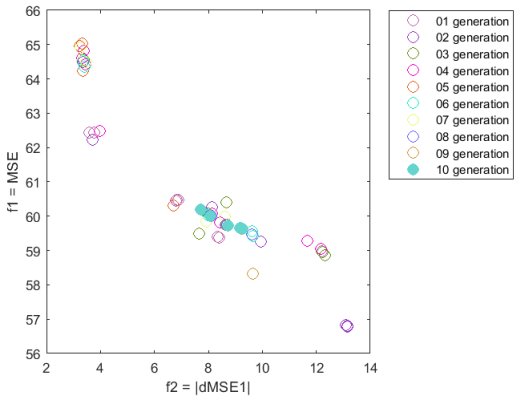
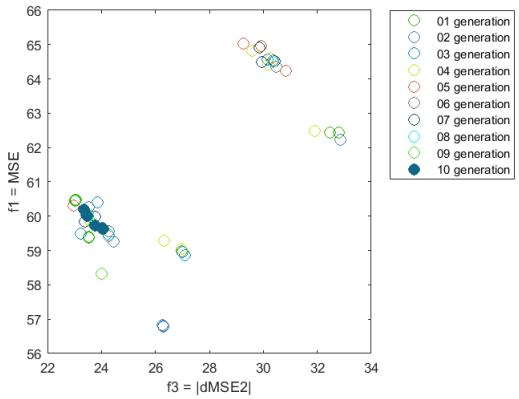
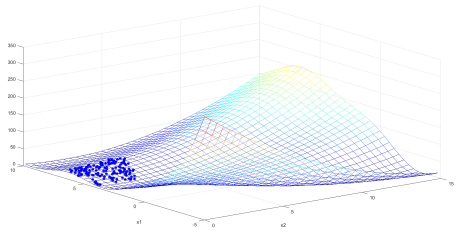
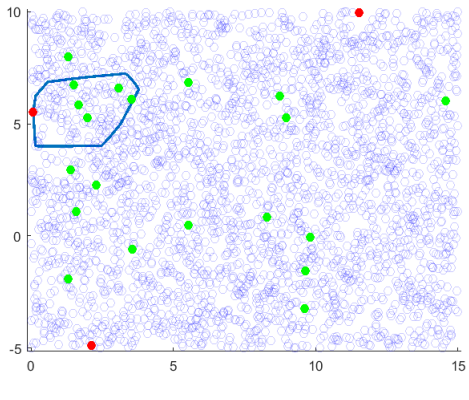
Zone 3	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

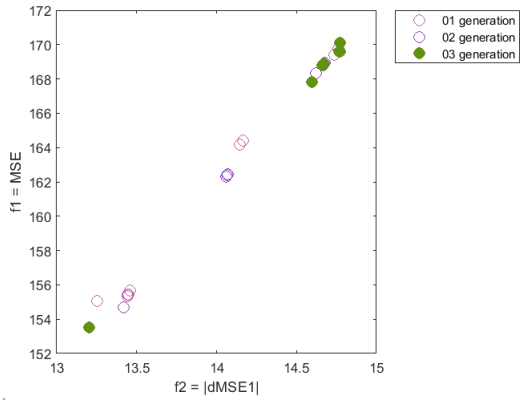
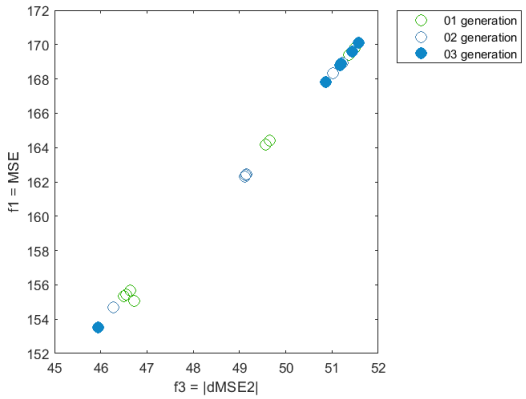
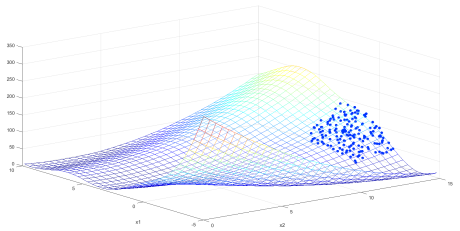
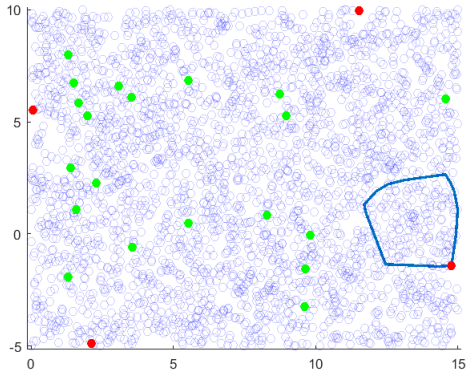
Zone 4	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

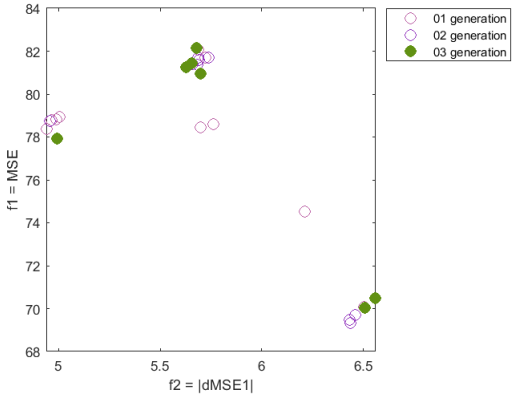
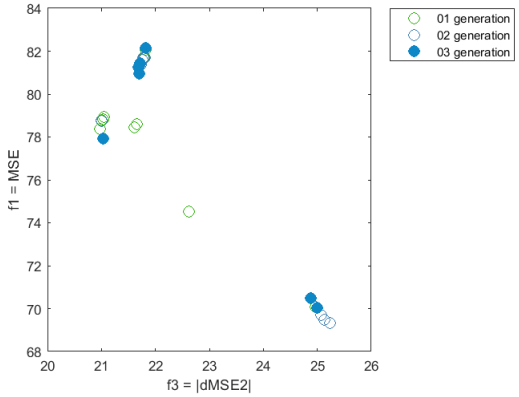
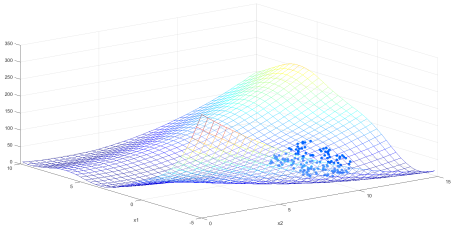
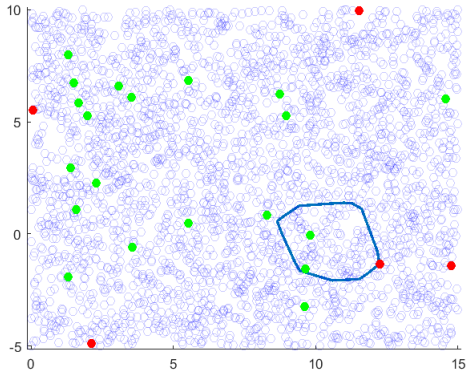
Zone 5	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

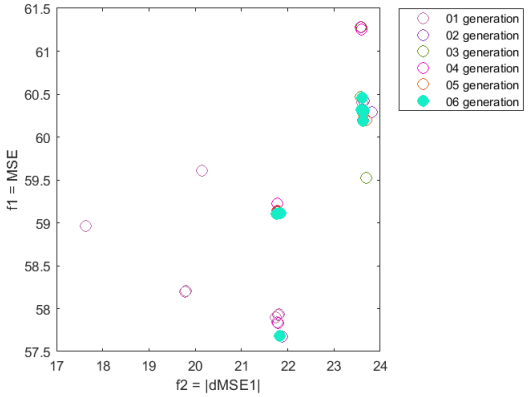
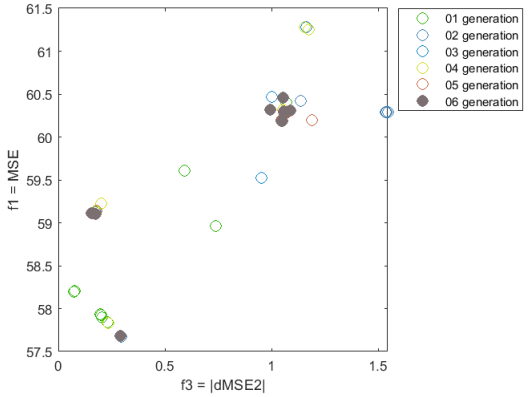
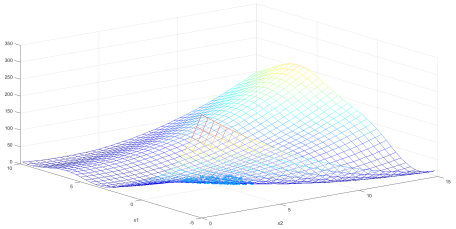
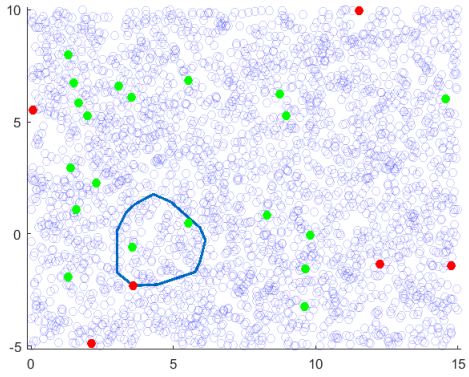
Zone 6	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

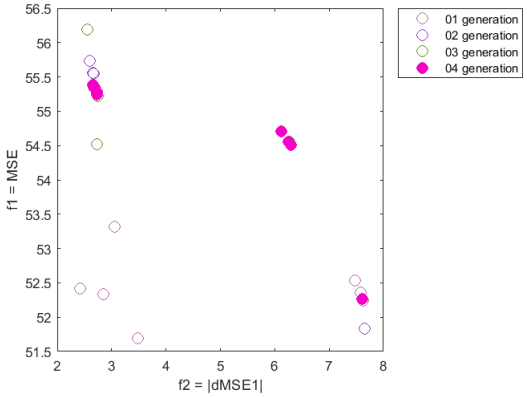
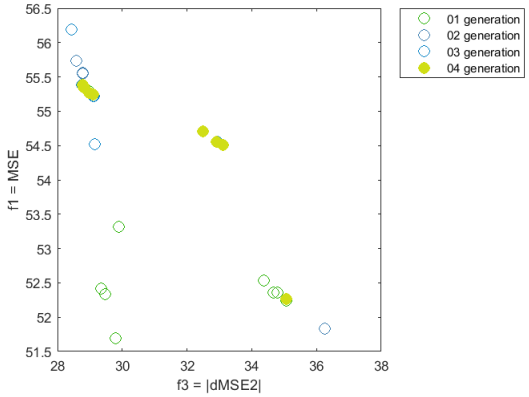
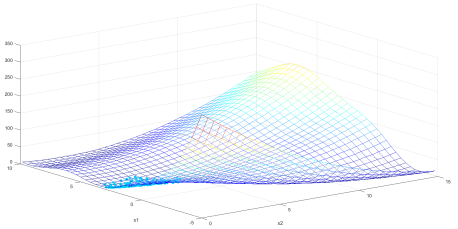
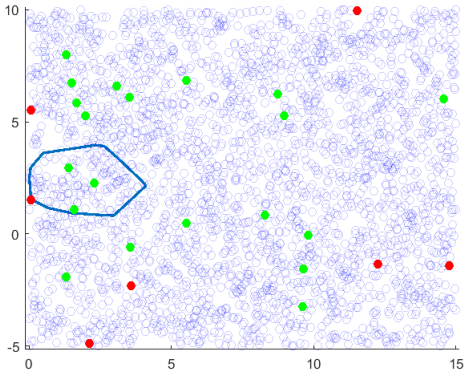
Zone 7	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

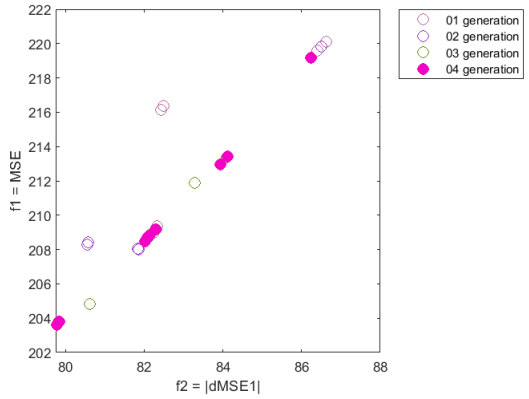
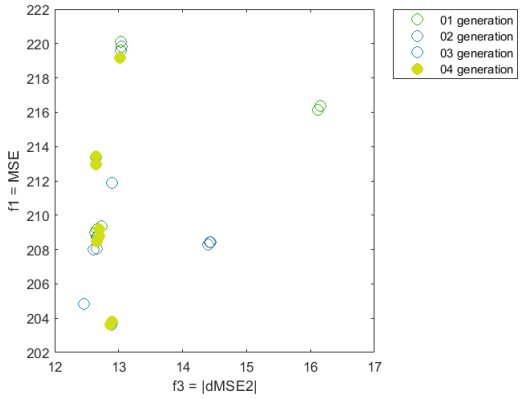
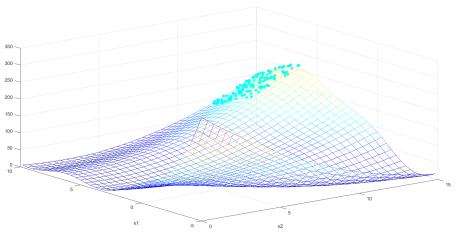
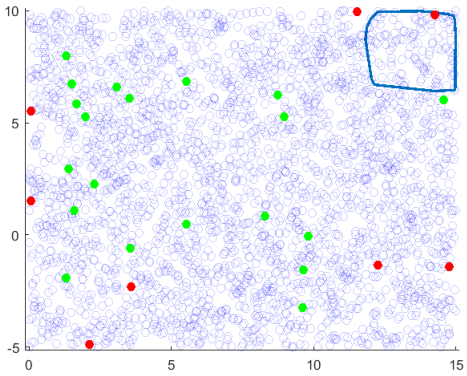
Zone 8	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

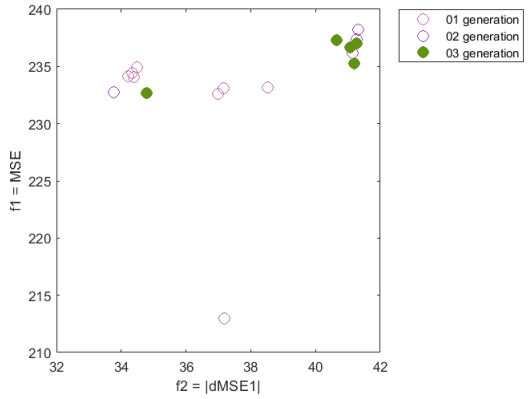
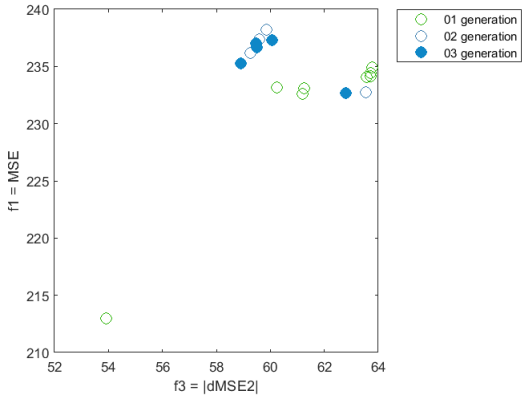
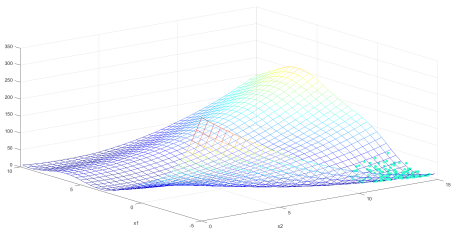
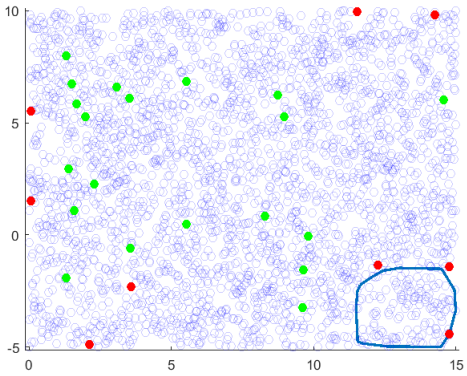
Zone 9	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

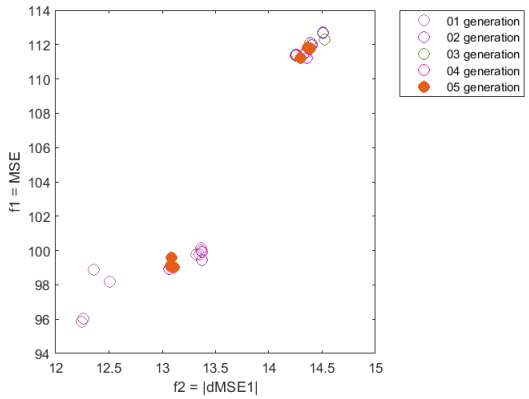
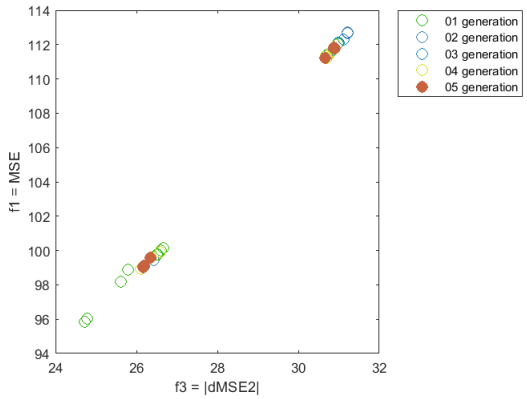
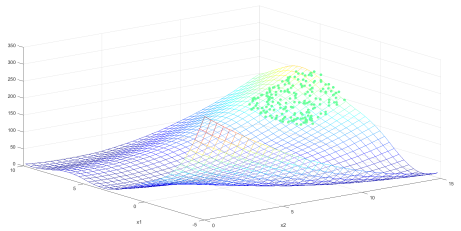
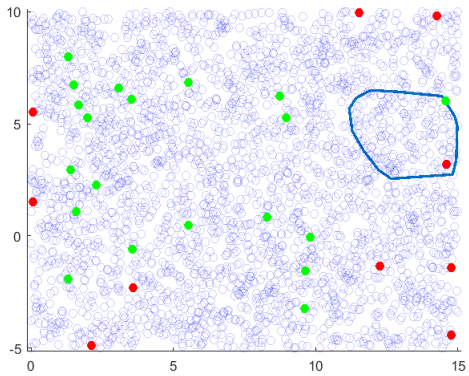
Zone 10	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

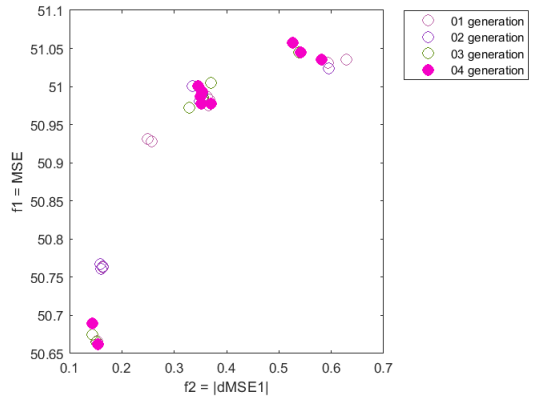
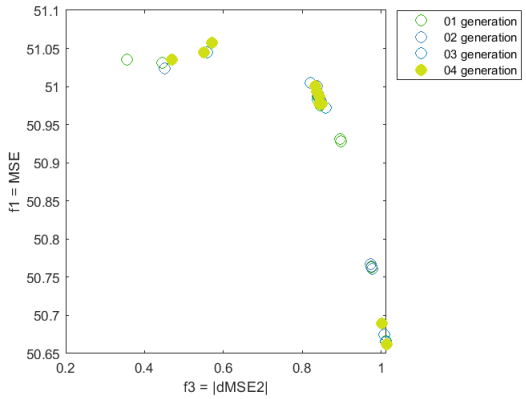
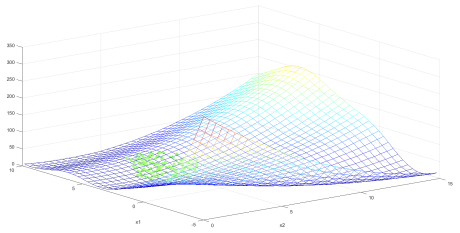
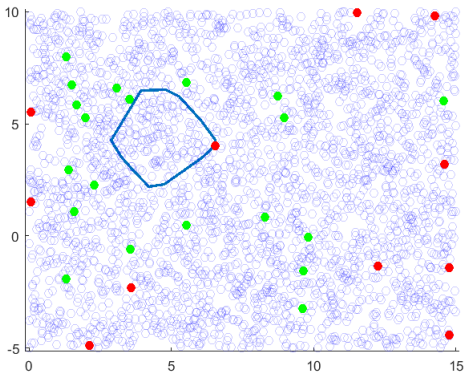
Zone 11	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

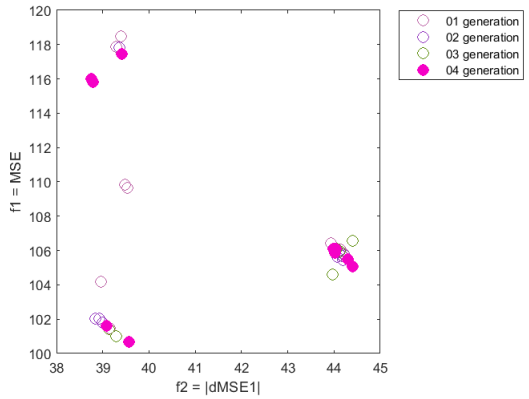
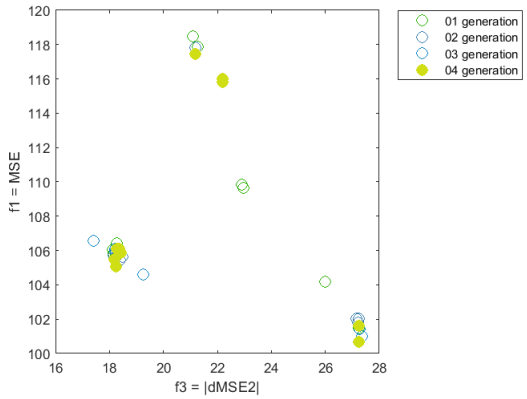
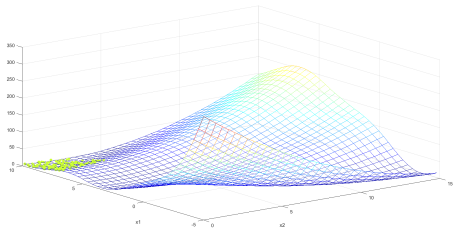
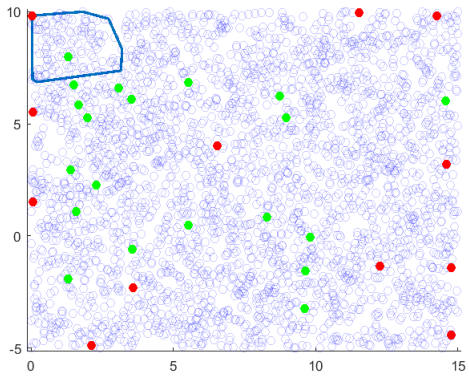
Zone 12	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

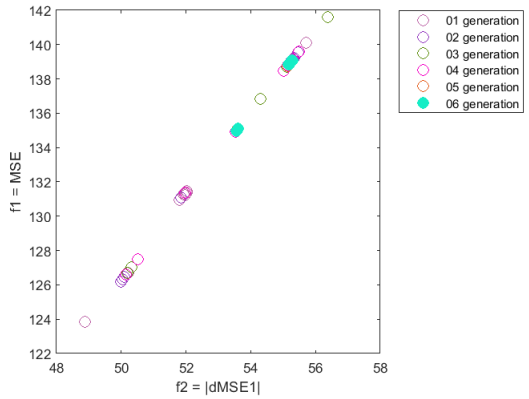
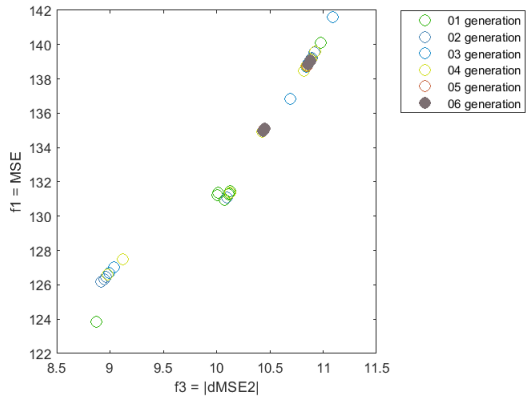
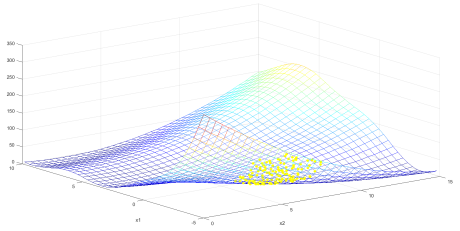
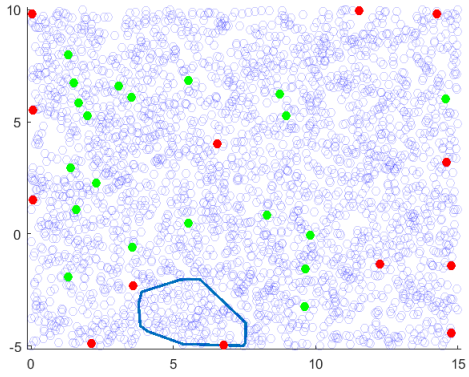
Zone 13	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

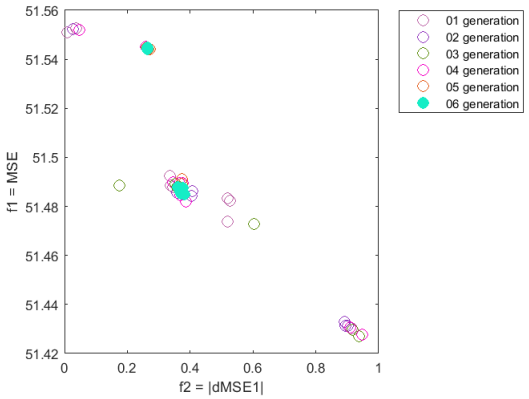
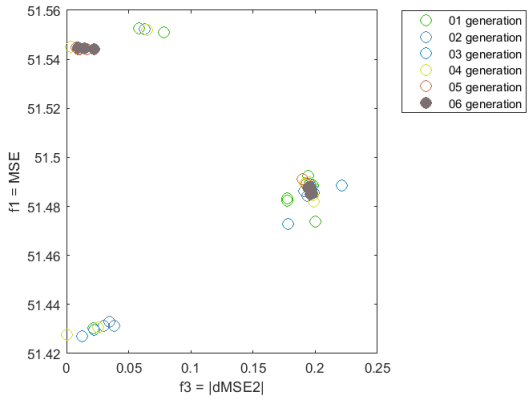
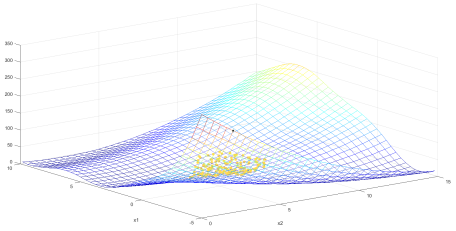
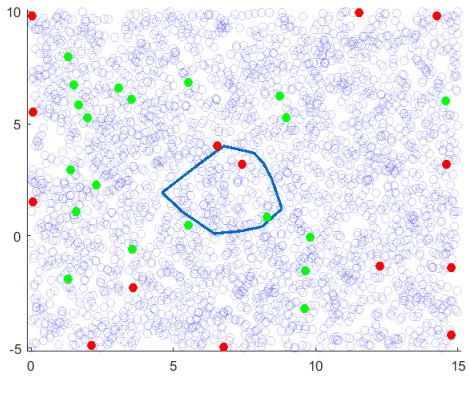
Zone 14	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

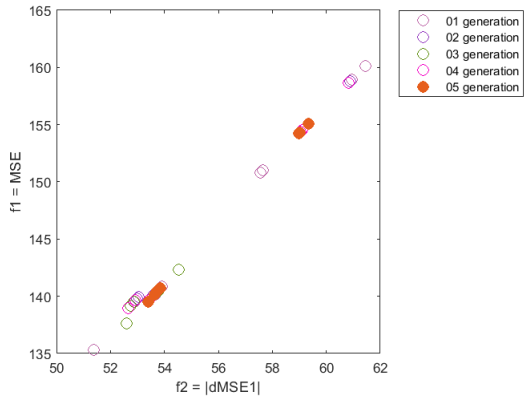
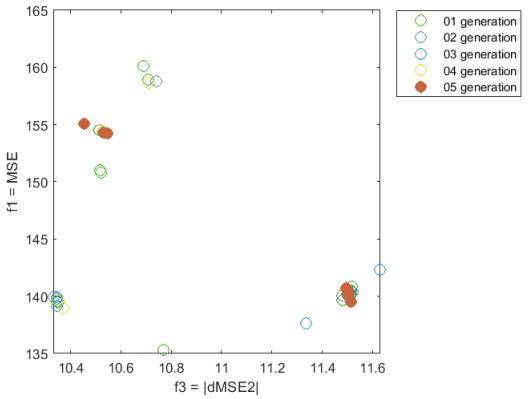
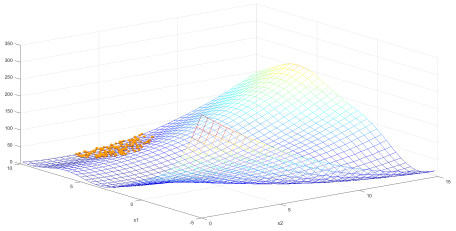
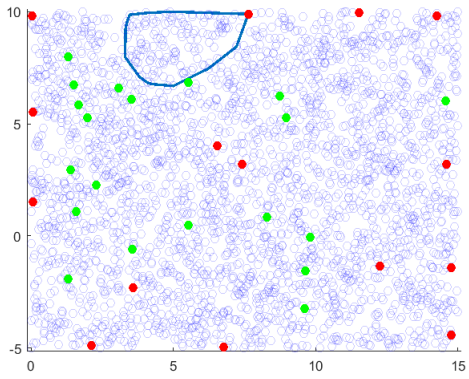
Zone 15	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

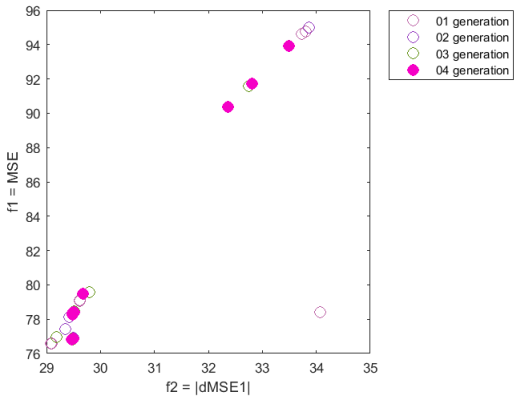
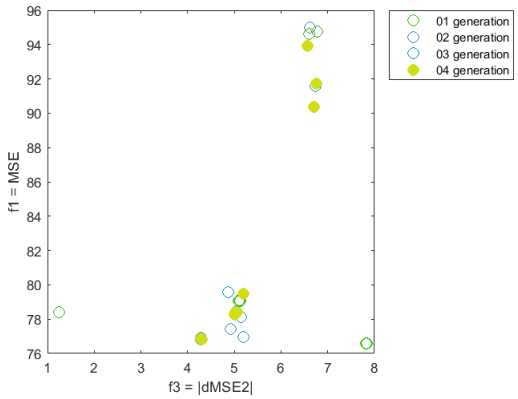
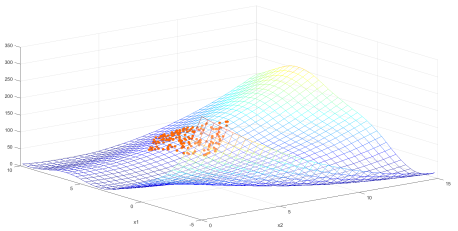
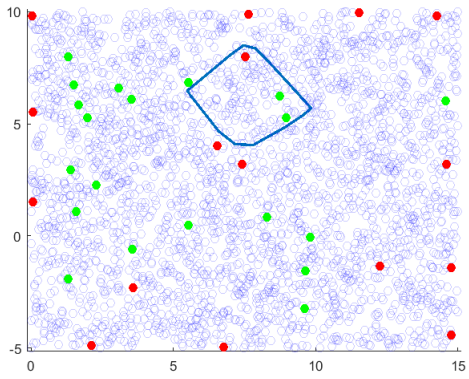
Zone 16	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

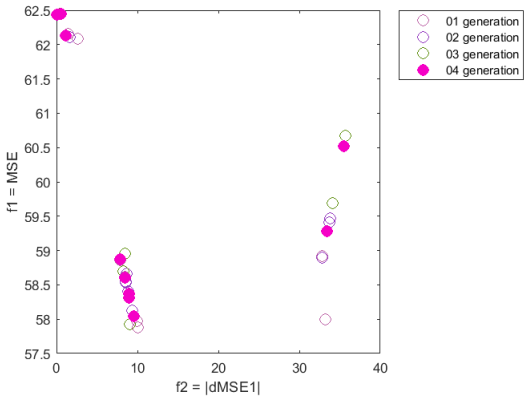
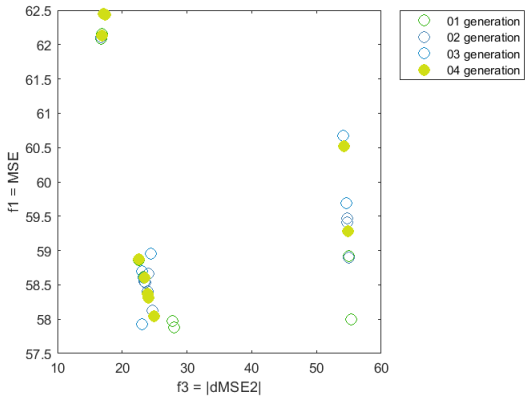
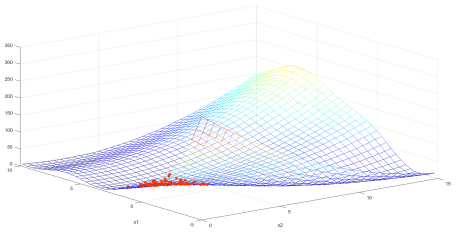
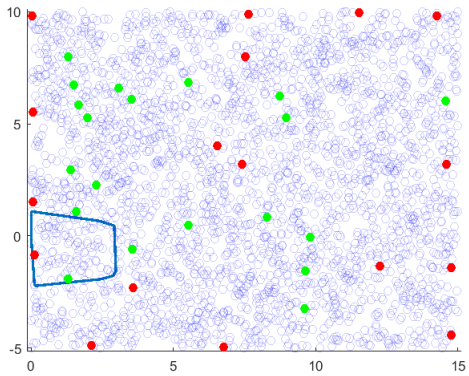
Zone 17	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

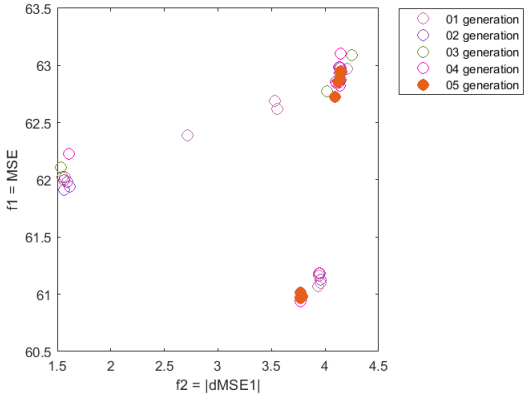
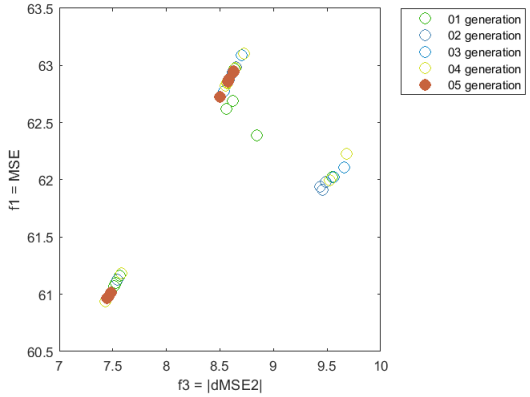
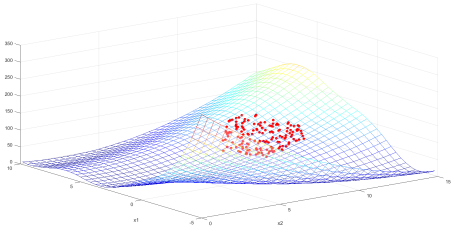
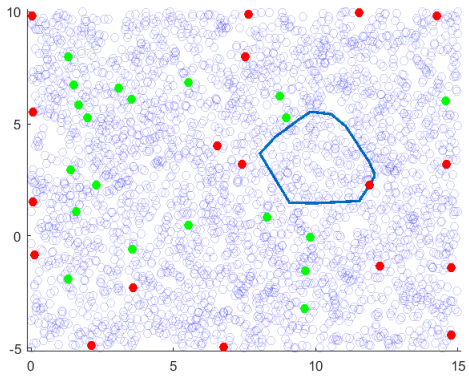
Zone 18	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

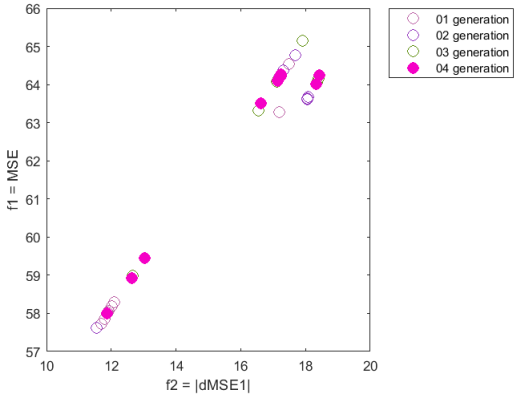
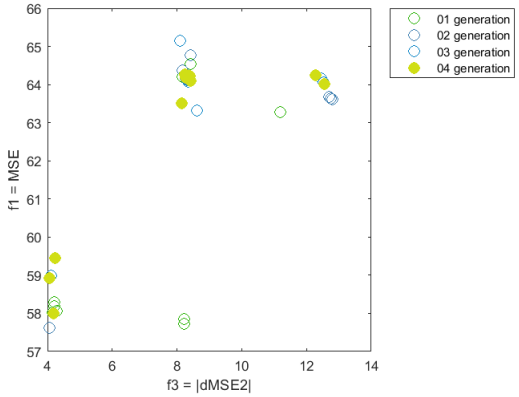
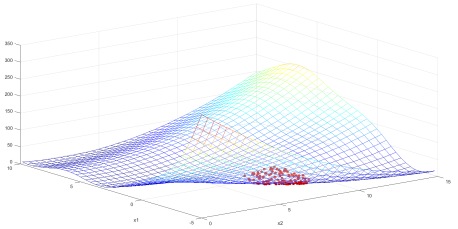
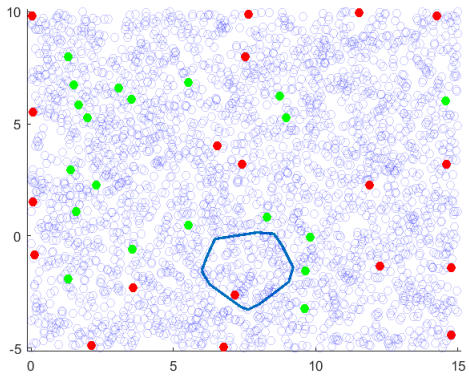
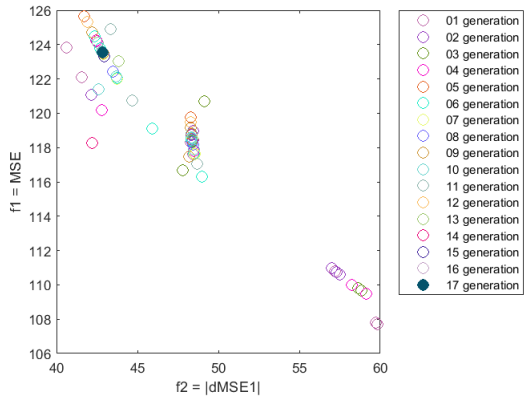
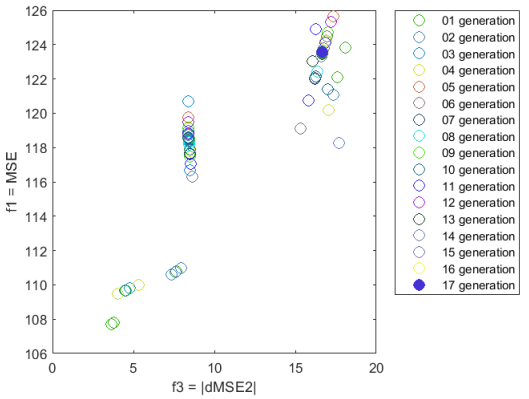
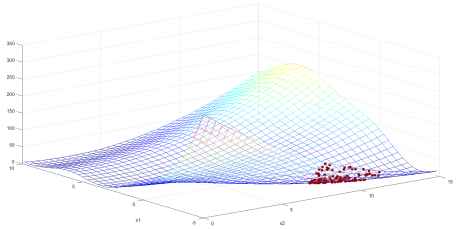
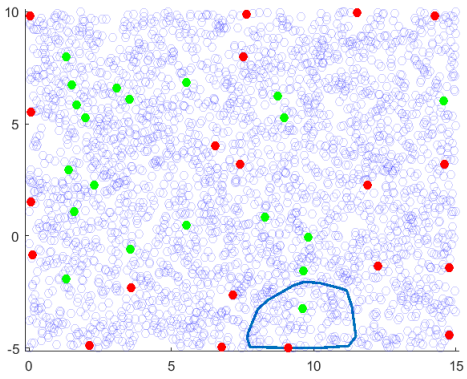
Zone 19	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

Table 4.1: (continued)

Zone 20	
Genetic objective functions	
	
ZONE	ADDED RED POINT
	

In general, there is no straightforward correlation between the objective functions behaviour and the respective zone where they are evaluated. However, according to the figures, it is noticed that the objective function values are influenciated by the presence of initial sample indicated by green points. Such relationship is due to the fact that these sample points are used to compute all MSE and dMSE values for the cp mesh through CK and then, close to green points, the MSE values shall be relativey small. Also, it is important to mention that in a single computation loop (see topic 3.4), one red point from a given zone have no influence on the localization of next red point in next zone.

Therefore, it is possible to observe that zones with large relative distance from the initial sample points have relative high MSE and, as consequence, few generations computed from GA. Such characteristic can be applied on zones 2, 4, 8, 9, 10, 13 and 15. In special, zones 2, 4, and 13, demonstrate a linear relationship among the objective functions. In opposite sense, the zones 1, 3, 5, 6, 7, 11, 12, 14 and 20 present high non-linear Pareto frontier with relative higher GA participation.

Table 4.2 shows the final sampling obtained from GA and its comparison with the ones obtained through the CK and AK methods. Such results also considers the sampling for the Branin fuction related to an evolution from 20 to 80 points considering the already existing 20 initial sample points (green points). These initial points do not have necessarily the same distribution due to their random selection. However, as already explained about CK, the sampling is relatively weak, as it uses only the RND procedure. In AK, the sampling improves, following the process described in the chapter 3 and in the beginning of this chapter. However, it still fails to distribute the sample in a homogeneous way creating some voids and aleatory configuration in the design space. Finally, the HC tries to uniformize the sample distribution into the design space. The next topic shows the efficiency of metamodels generated by each sample, demonstrating the improvement achieved by HC.

Table 4.2: Comparison of final sampling for CK, AK and HC

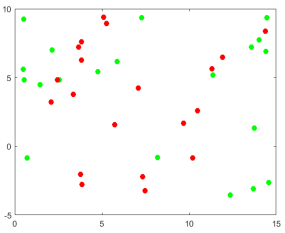
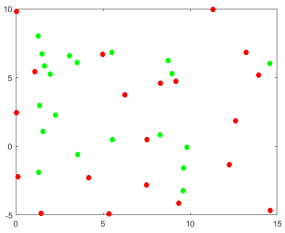
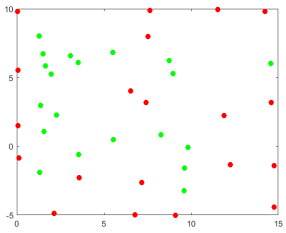
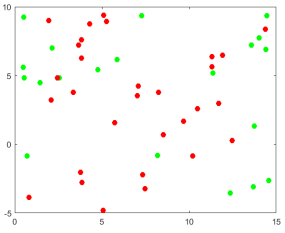
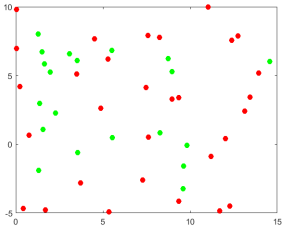
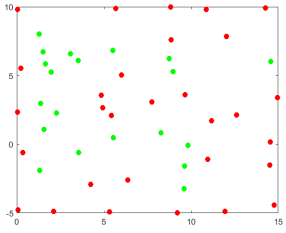
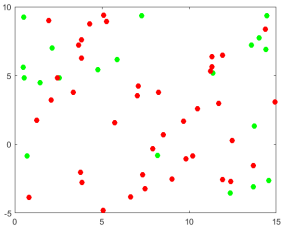
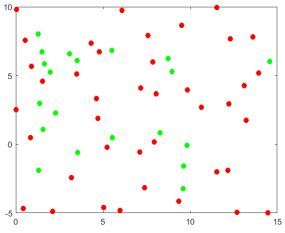
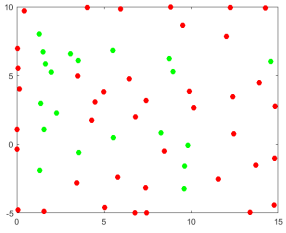
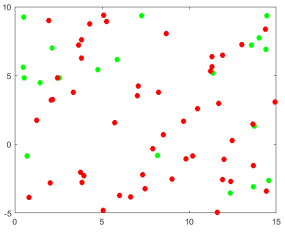
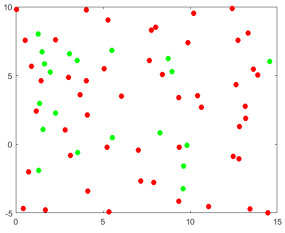
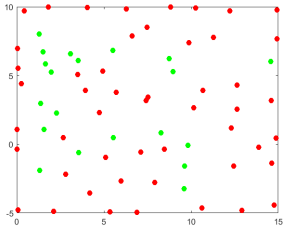
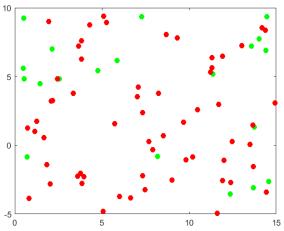
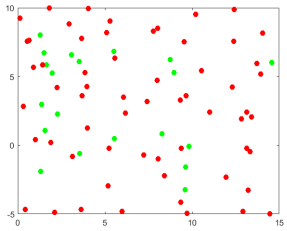
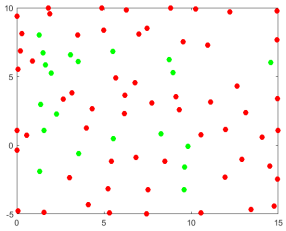
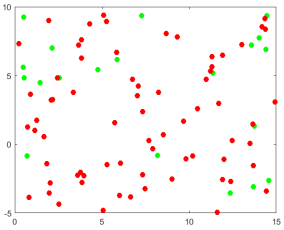
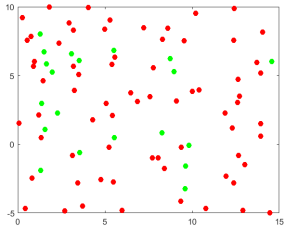
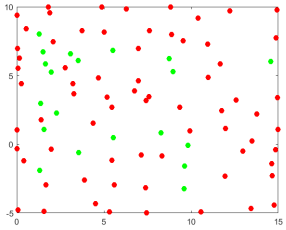
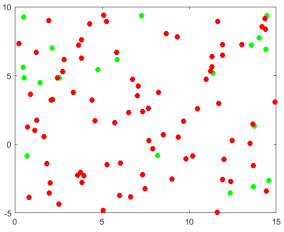
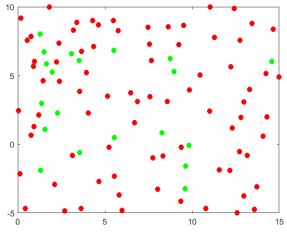
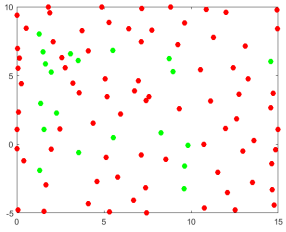
CONVENTIONAL KRIGING	ADAPTIVE KRIGING	HYBRID CODE
20 added points		
		
30 added points		
		
40 added points		
		
50 added points		
		

Table 4.2: (continued)

CONVENTIONAL KRIGING	ADAPTIVE KRIGING	HYBRID CODE
60 added points		
		
70 added points		
		
80 added points		
		

4.1.3 Validation

Focusing on performance measurement, the mean errors obtained in CK, AK and HC are compared as indicated in figure 4.4.

The lower abciss axis indicates the total size of sample points while the upper axis indicates the quantity of new points added on the sample by AK and HC. The range analyzed for Branin is between 40 and 100 points considering a set of 20 and 80 new added points on the sample. It means that AK and HC uses 20 fixed sample points to generate the Basis Metamodel before the application of their respective method.

Therefore, it is possible to observe that CK presents the worst performance and AK demonstrates an intermitent improvement. The best performance indicated in figure 4.4 for

the range of the analyzed sample size is obtained by HC.

Figure 4.5 shows the bahavior of the HC after applying 3 loops for the Basis Metamodel obtained by CK through the initial 20 fixed points, according to the concept explained in topic 3.4. In this result, regardless the slight improvement achieved by the 3 loops case, it does not represent a significant improvement in comparison to the one obtained by the single loop.

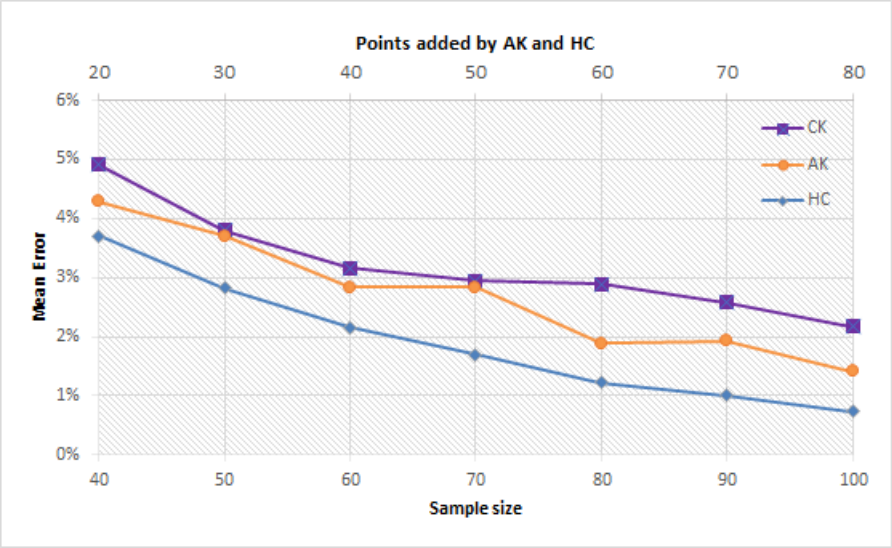


Figure 4.4: Mean error comparison among CK, AK and HC methods for Branin function.

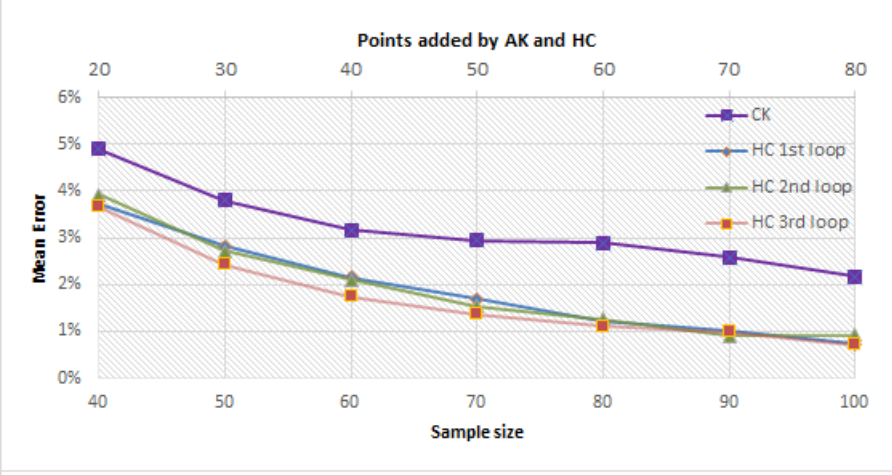


Figure 4.5: Mean error analysis for CK and Basis Metamodel loops in HC method for Branin function.

Regarding the computational time related to CK and HC methods, figure 4.6 shows the difference among the methods using 1, 2 and 3 Basis Metamodel loops. As the Branin function is represented by a simple analytical equation, there is a large difference in time scale between CK and HC, as the HC requires many more processing steps than CK. However, as it is shown in topic 4.2.3, the HC has advantages when dealing with complex problems that involves time consuming computations for obtaining the sample point results.

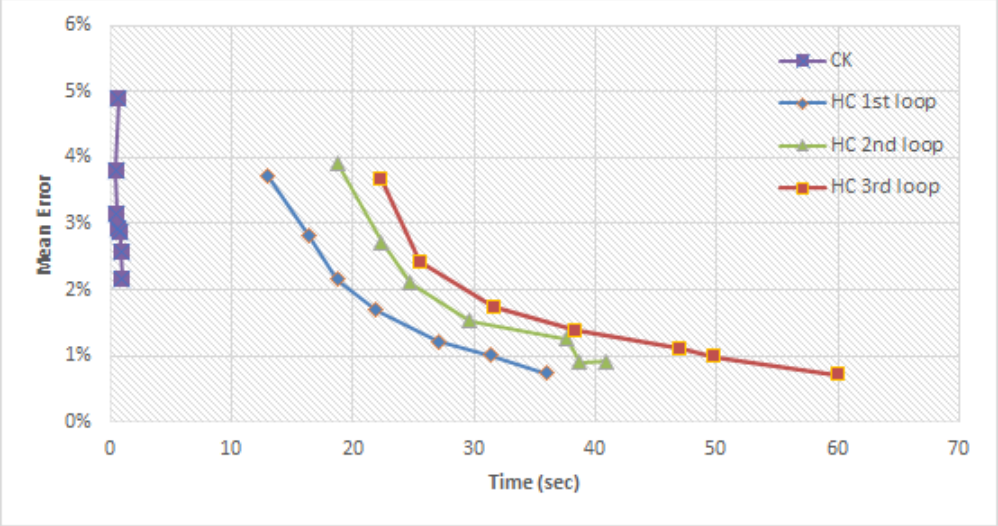


Figure 4.6: Computation time comparison among CK and HC with 1, 2 and 3 Basis Metamodel loops for Branin function.

4.2 Product Peak Integrand Family Function

4.2.1 Function description

According to Surjanovic et al. (2020), the Product Integrand Family function is n-dimensional and is described by eq. 4.3. The domain for x_i is [0,1]. The parameters u_i and a_i are 0.5 and 5, respectively. Figure 4.7 shows the function for 2 variables.

$$f(x) = \prod_{i=1}^{n} \frac{1}{a_i^{-2} + (x_i - u_i)^2} \tag{4.3}$$

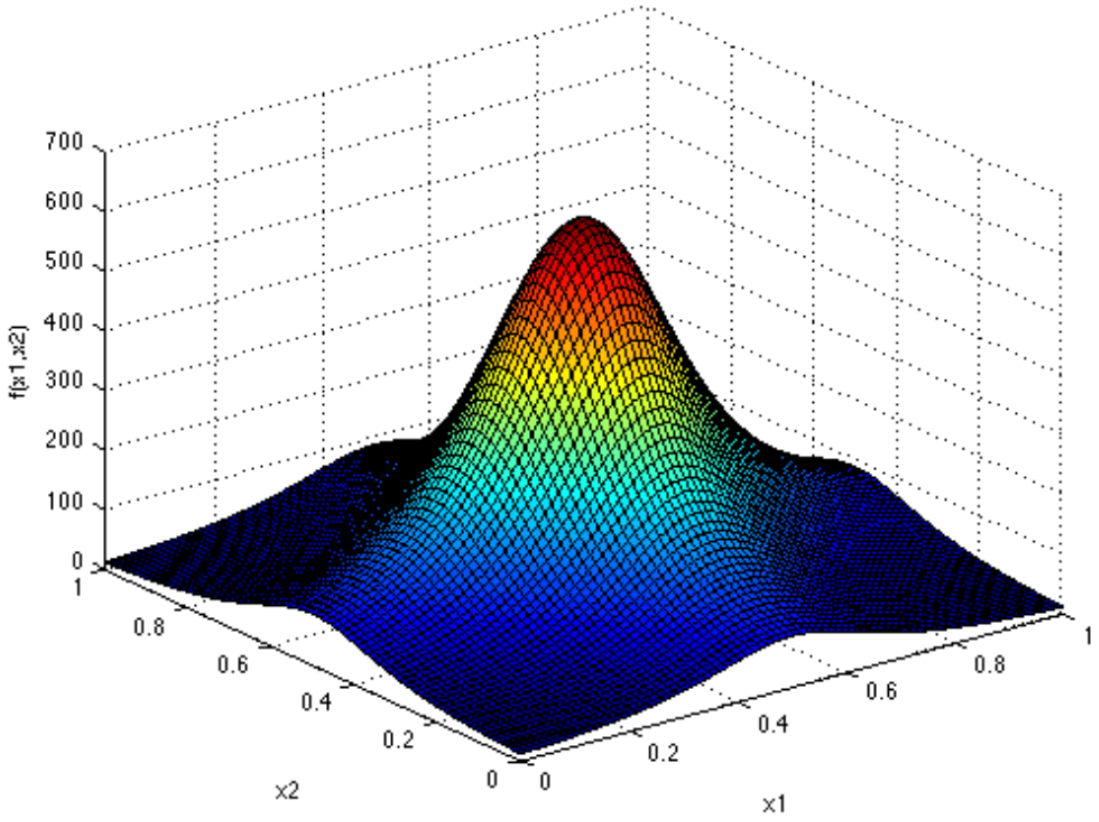


Figure 4.7: Product Peak Integrand Family function for 2 variables according to Surjanovic et al. (2020).

4.2.2 Validation for 2 variables

Similarly to the Branin function, the performance measurement analysis is determined by the mean error comparison among CK, AK and HC presented in figure 4.8.

The lower axis shows the total number of sample points, which varies from 40 to 130 points. The number of new points, that are added to the initial sample by AK and HC, is shown in the upper axis and varies from 30 to 120 points. Thus, the AK and HC uses 10 initial fixed sample points to generate the Basis Metamodel before the application of the method.

The results show again the worst performance achieved by CK and an intermitent improvement obtained by AK. Even considering a singular better performance of AK over HC on 70 sample size scenario, the last one shows the best overall performance.

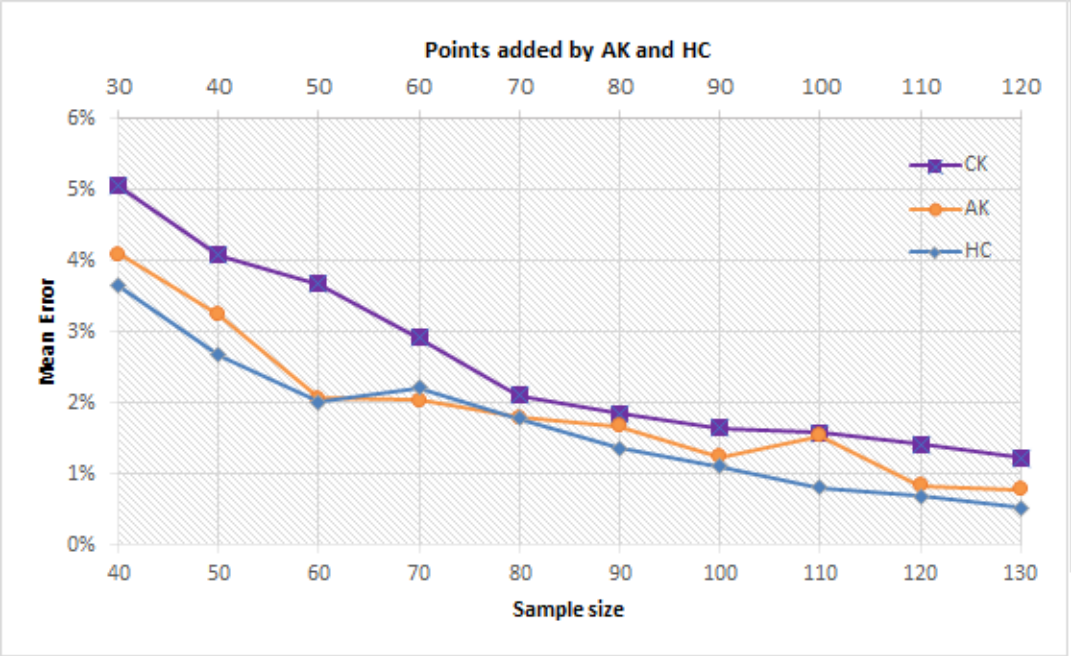


Figure 4.8: Mean error comparison among CK, AK and HC methods for Product Peak Integrand Family function of 2 variables.

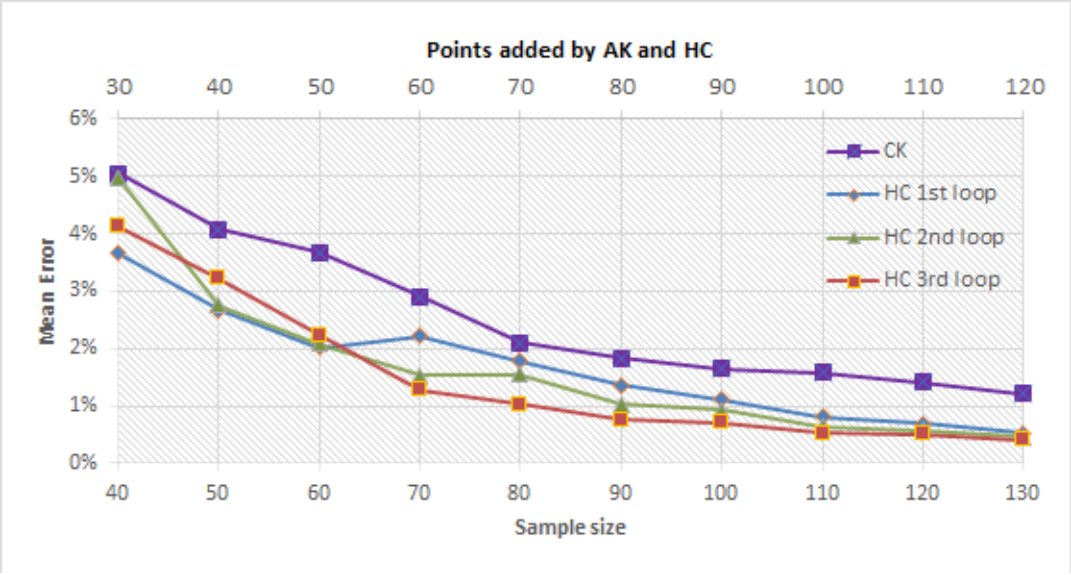


Figure 4.9: Mean error analysis for CK and Basis Metamodel loops in HC method for Product Peak Integrand Family function of 2 variables.

Considering again the Basis Metamodel generated by CK through the fixed 10 points,

figure 4.9 shows the behavior of HC considering 3 loops. Differently than the Branin function, the 3 loops case shows a relative better performance for total sample size scenario larger than 70 points.

Figure 4.10 compares the computational time between CK and HC for 1, 2 and 3 Basis Metamodel loops. Similar short computation time have been captured for CK in comparison to HC as presented in Branin function. However, focusing on the Basis Metamodel loops, sometimes more generations from GA can be involved in HC computations for a specific loop than other different loop scenario. It happens for the cases shown in figure 4.10, which leads to some fluctuations. However, no significant improvement of updating loops for Basis Metamodel has been identified considering such results.

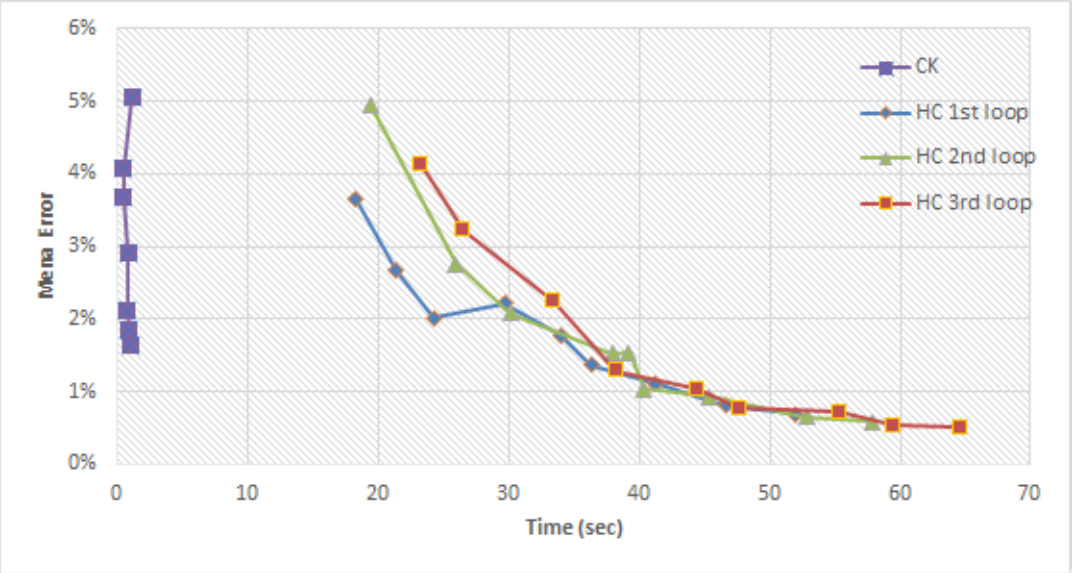


Figure 4.10: Computation time comparison between CK and HC with 1, 2 and 3 Basis Metamodel loops for Product Peak Integrand Family function of 2 variables.

4.2.3 Validation for 10 variables

The validation procedure for the 10 variables case compares the mean errors from the CK, AK and HC, which are shown in the graph of figure 4.11. The lower axis indicates the total size of sample points with a range from 120 to 165 points. The upper axis shows the quantity of new points added on the sample by AK and HC, which is between 50 and 95 points. Thus, the AK and HC uses the 70 sample points for CK to generate the Basis Metamodel before the application of the methods.

Similarly than before, the worst performance is indicated by CK, while AK shows an intermitent improvement. HC shows the best performance in the overall range of the sample analyzed.

For Basis Metamodel generated by CK through fixed 70 points, figure 4.12 shows the behavior of HC considering 3 loops. Similarly than before, no signficant difference is observed among these three analyzed loops.

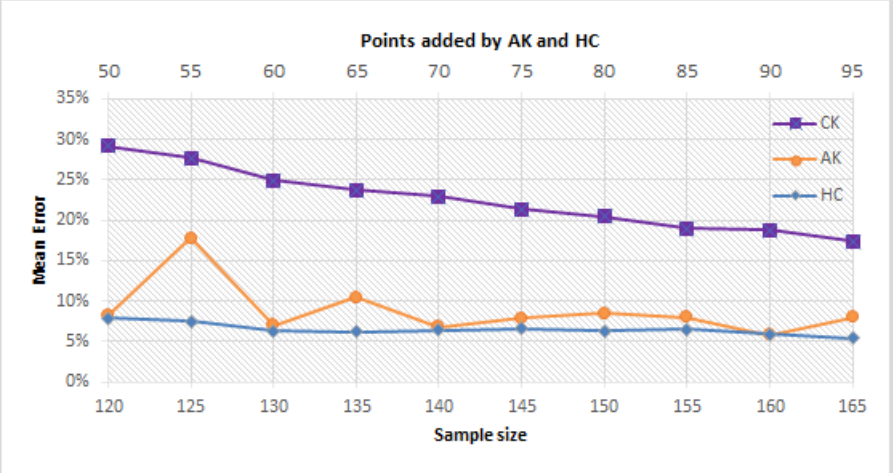


Figure 4.11: Mean error comparison among CK, AK and HC methods for Product Peak Integrand Family function of 10 variables.

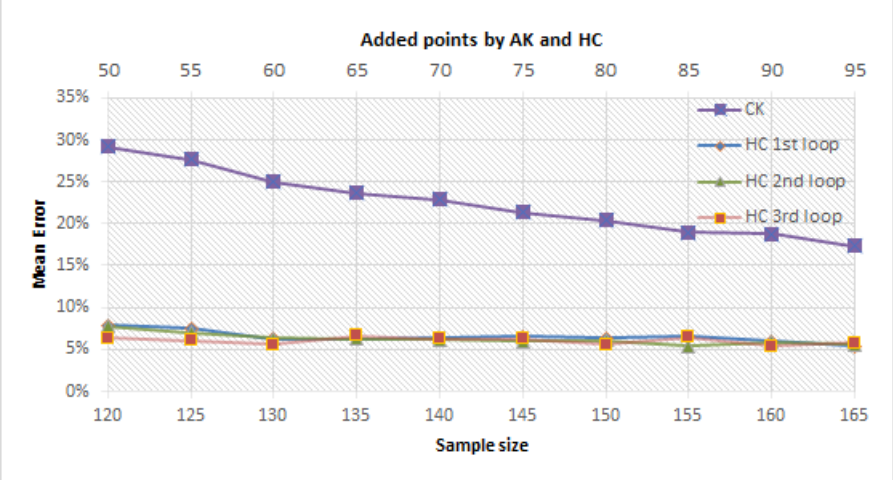


Figure 4.12: Mean error analysis for CK and HC method with Basis Metamodel loops for Product Peak Integrand Family function of 10 variables.

Regarding the computation time, figure 4.13 shows the time difference between CK and HC for 1, 2 and 3 Basis Metamodel loops.

It is also observed in figures 4.11 and 4.12 a small decrease of the mean error with the increase of the sample size for both methods CK and HC.

Regarding the computational time, figure 4.13 provides the overall time performence between the methods, which has time and error difference scale between CK and HC. However, figure 4.14 shows the comparison of the computational time obtained by processing additional points in CK method such that it reaches the same mean error of HC method. The number of additional points added to CK are between 300 and 1100 points. Although the final time scale shows the same order of magnitude between the methods with similar accuracy (mean error), CK needs around 25 times more points than HC. Such representativeness for much more points for CK in relation to HC can make a large difference when dealing with more complex problems.

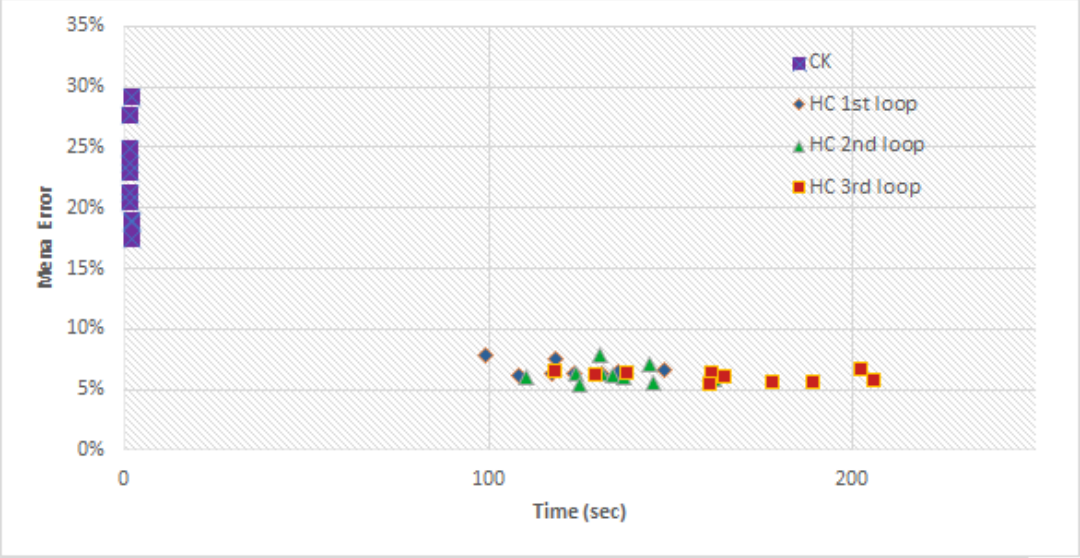


Figure 4.13: Computation time comparison among CK and HC with 1, 2 and 3 Basis Metamodel loops for Product Peak Integrand Family function of 10 variables.

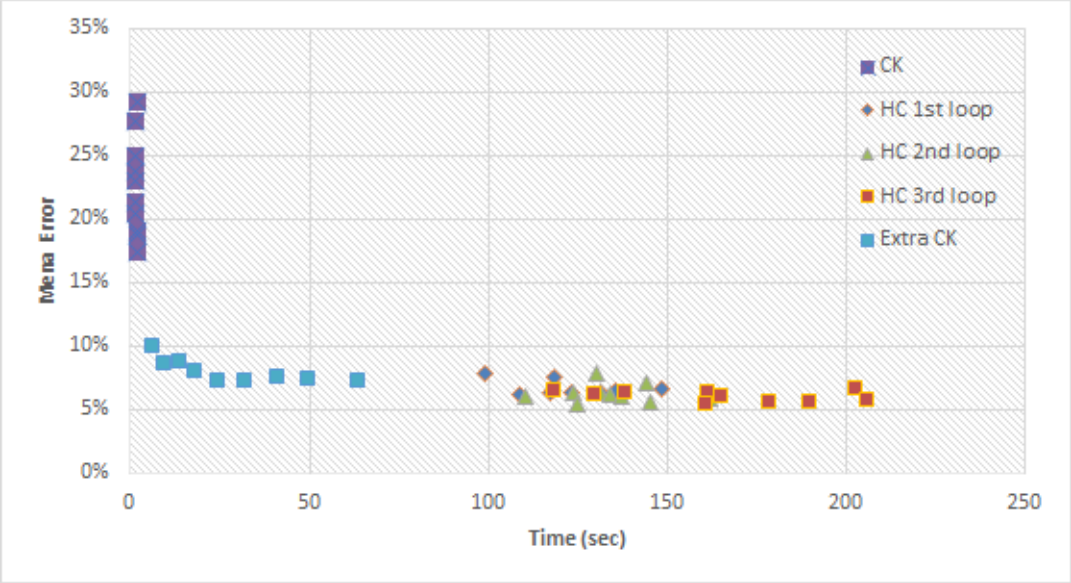


Figure 4.14: Computation time comparison between CK and HC with 1, 2 and 3 Basis Metamodel loops for Product Peak Integrand Family function of 10 variables considering extra sample points (300 to 1100) for CK.

4.3 General comments

As presented and discussed in this chapter, HC has been validated considering two analytical functions: Branin and Product Peak Integrand Family for 2 and 10 variables. In all cases, HC shows better mean error performance than CK and AK. Specifically for the last method, its comparison with HC indicates that only the AK coupled with cp mesh that brings the adaptative approach is not enough to obtain a robust metamodel with relative high fidelity as HC does. GA complements the AK with cp mesh through gene operations abling to exploit deeper the design space with new points to be added in the sample.

Similar mean error results achieved by using different number of loops is explained by the high efficiency of the MSE parameter in identifying the new sample points. This is verified by checking that the original MSE obtained from the single Basis Metamodel loop at each zone of the domain does not have a significant change in the following loops. In other words, the updated Basis Metamodel has not significant difference from its original configuration. This behavior can be generalized for any type of problem. It also demonstrates a weak influence on the quantity of zones to be meshed with the final sampling performance. This aspect is confirmed in the next chapter with the simulation of engineering aplication problems.

Regarding the computation time, the results obtained from the Product Peak Integrand Family function for 10 variables indicate that the time efficiency of HC increases with the increasing of the quantity of variables. Such correlation also occurs due to the increase of the general complexity in the problems calculation. Chapter 5 confirms these conclusions.

CHAPTER 5

Application of the Hybrid Code on engineering problems

Once HC has been validated in chapter 4, its application in engineering problems through the use of FEM commercial codes configures the next step of analysis, which is presented in this chapter. Considering that, two problems are analyzed: the cantilever beam simulated in ANSYS and a more complex problem related to UNDEX scenario simulated in ABAQUS.

5.1 Cantilever Beam

5.1.1 FEM Model

The cantilever beam has 10 sections with cross section dimensions equal to 0.5m x 0.5m. Each section has 1m length and its thickness varies as in figure 5.1. The thickness is constrained such that it is either equal to its predecessor or decreases towards the free end. This constraint is imposed only for the CK, cp mesh and test points. For simplicity, it is not applied on the points generated by GA in HC.

Figure 5.1: Cantilever beam model.

The geometry is shown in figure 5.1. The mesh uses SHELL281 quadratic shell elements according to *Elements Reference - ANSYS Manual* (2007). The boundary conditions are clamped in one end and a local vertical force applied in the other end (figure 5.2).

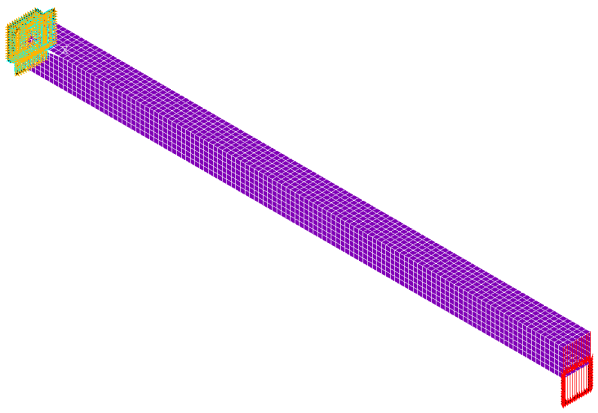


Figure 5.2: Boundary conditions applied on the cantilever beam.

The material is common steel with linear mechanical properties (5.1).

Table 5.1: Material Properties

PROPERTY/PARAMETER	MODEL INPUT VALUE
Elasticity Modulus	200 GPa
Poisson ratio	0.3

The maximum stress at the clamp section and the maximum displacement at the free end are the outputs, as presented in an example of the stress results over the beam sections shown in figure 5.3.

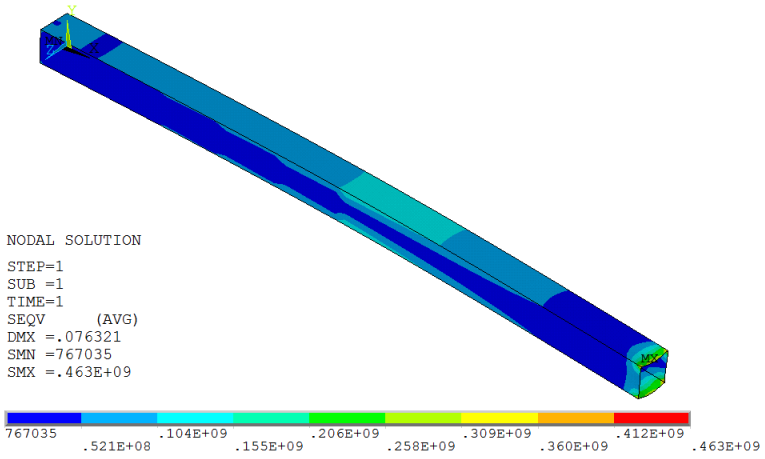


Figure 5.3: Example of stress results applied on the cantilever beam.

5.1.2 Displacement Results

The mean error for the displacement of the free end of the cantilever beam, obtained from metamodels generated by HC and CK, is shown in figure 5.4. The total number of sample points varies from 100 to 180 and the quantity of new points added by HC varies from 10 to 90 points. The HC uses a metamodel generated by CK based on fixed 90 points (Basis Metamodel). The result achieved by HC considering three Basis Metamodel loops is shown in figure 5.5.

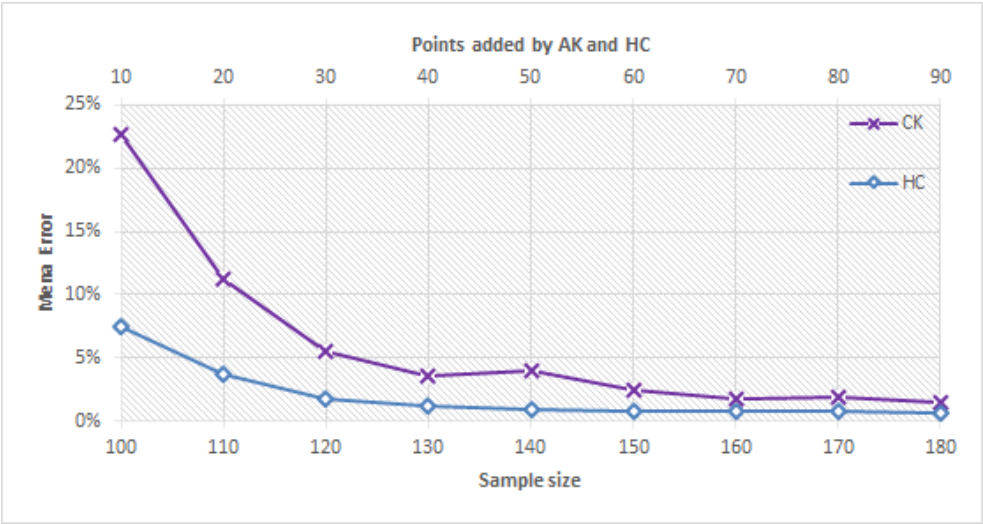


Figure 5.4: Mean error comparison betweeen CK and HC methods for the displacement results of the cantilever beam.

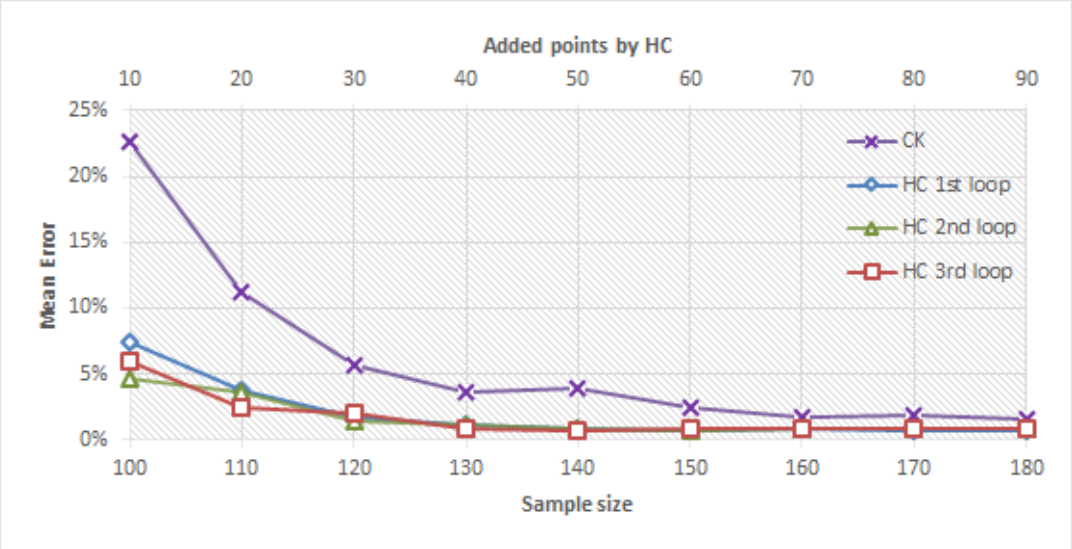


Figure 5.5: Mean error comparison betweeen CK and Basis Metamodel loops of HC for the displacement results of the cantilever beam.

From the results shown in figures 5.4 and 5.5, the mean error from HC is approximately 4 times smaller than the mean error of CK for the first sample size scenario (100 total sample points). This difference reduces with the increase of the number of sample points. Also, the use of Basis Metamodel loops does not improve considerably the mean error performance of HC. From this result, it is possible to infer that, for problems of large complexity, the need of a smaller number of sample points for obtaining the same mean error, by the HC procedure, can be an advantage regarding the computational time.

Figures 5.6 and 5.7 presents the same results of the computation time, but in different zoom. The results confirm the faster processing of the HC to obtain the same level of error of the CK for more complex problems, as mentioned in topic 4.2.3. Figure 5.8 shows an example of the time difference between the procedures for a 2% mean error.

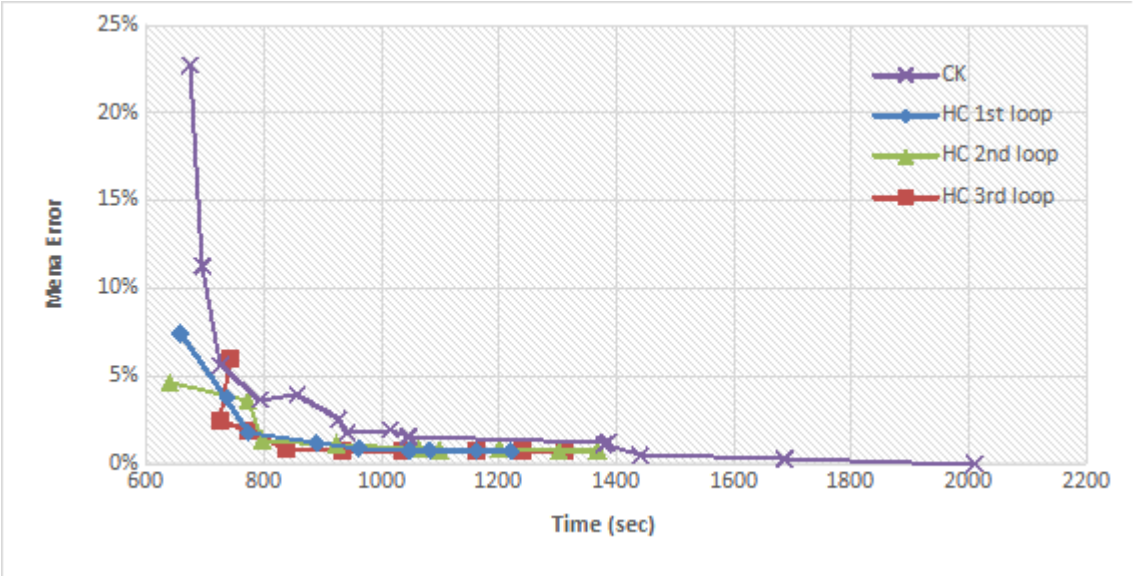


Figure 5.6: Computation time comparison between CK and HC for 1, 2 and 3 Basis Metamodel loops related to Cantilever Beam in displacement results.

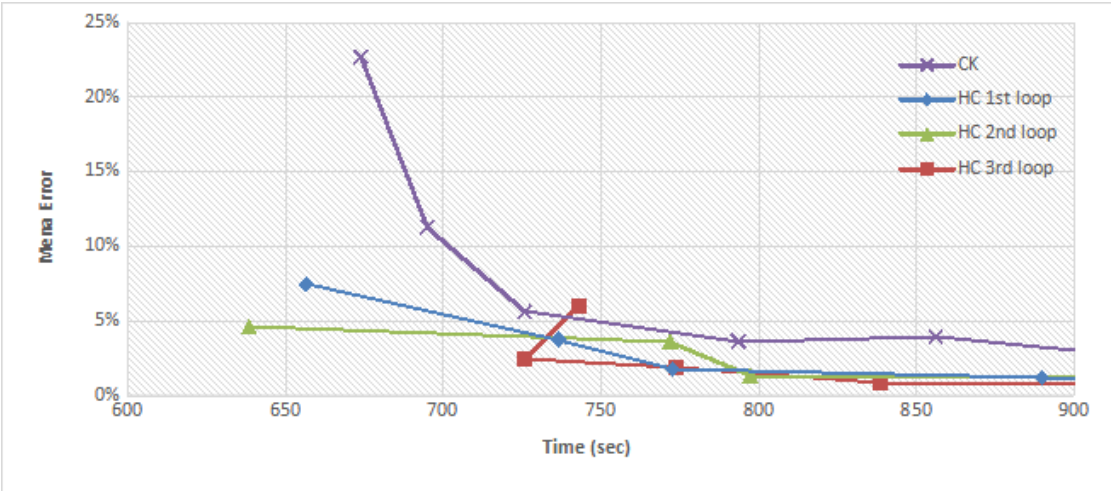


Figure 5.7: Computation time comparison, in a different time scale, between CK and HC for 1, 2 and 3 Basis Metamodel loops related to Cantilever Beam in displacement results.

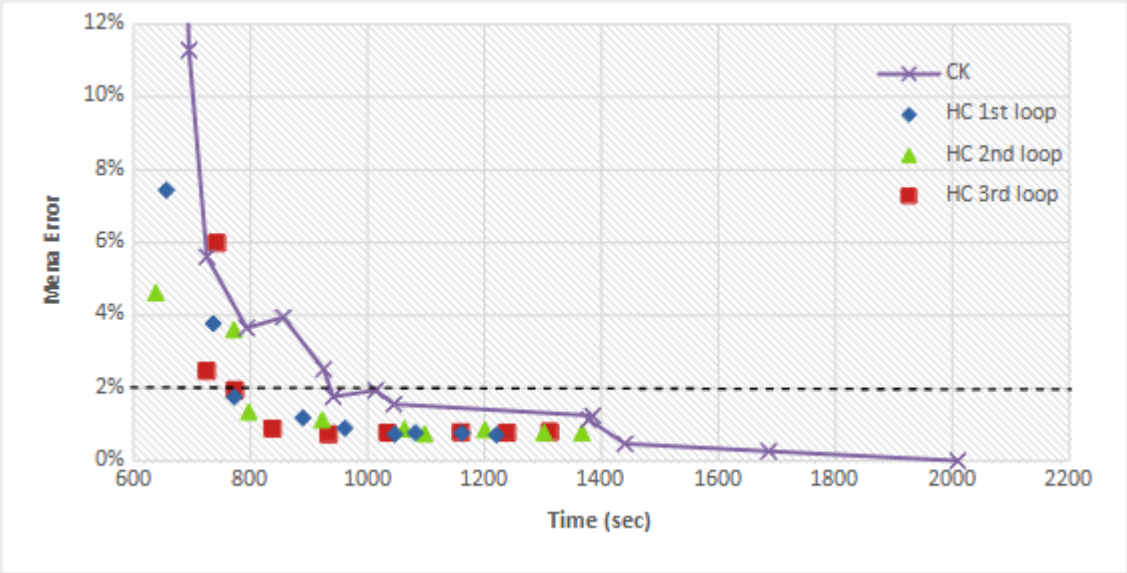


Figure 5.8: Computation time analysis for a 2% mean error between CK and HC for 1, 2 and 3 Basis Metamodel loops related to Cantilever Beam in displacement results.

5.1.3 Stress Results

The mean error performance for the maximum stress at the clamped end of the beam, achieved by HC and CK, is presented in figure 5.9. The total number of the sample points are between 100 and 180 and the quantity of new points added by HC is the same number of points used for the maximum displacement case (between 10 and 90 points). Figure 5.10 compares the mean error of the CK metamodel with the mean error of the HC, which is approximately half the mean error achieved by CK for a small sample size. The mean error performance of HC estabilizes for larger sample sizes. Regarding the number of Basis Metamodel loops, no considerable difference in the mean error performance is noticed.

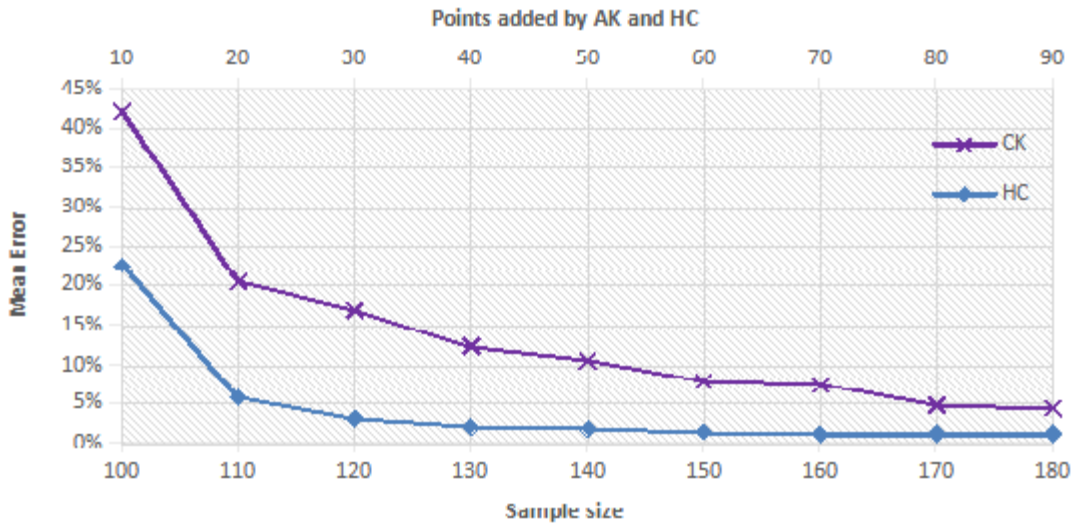


Figure 5.9: Mean error comparison between CK and HC methods for Cantilever Beam in stress results.

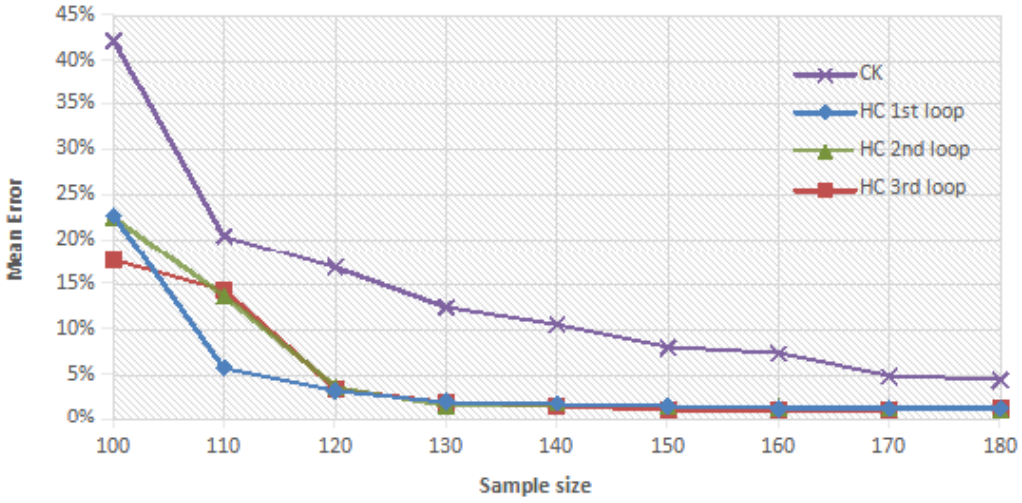


Figure 5.10: Mean error analysis for CK and Basis Metamodel loops for HC method related to Cantilever Beam in stress results.

Figures 5.11 and 5.12 compare the computational time of HC and CK for different sample size scales. These results confirm again the efficiency of HC for large quantity of variables in comparison to the CK method. Establishing the same 1% mean error for both methods in figure 5.13, it is noticed a computational time of the HC about 20% of the computational time of the CK. In this analysis, the difference about computation time spent by CK to achieve similar mean error performence than HC becomes huge (around

5000 sec for CK against 1000 sec for HC).

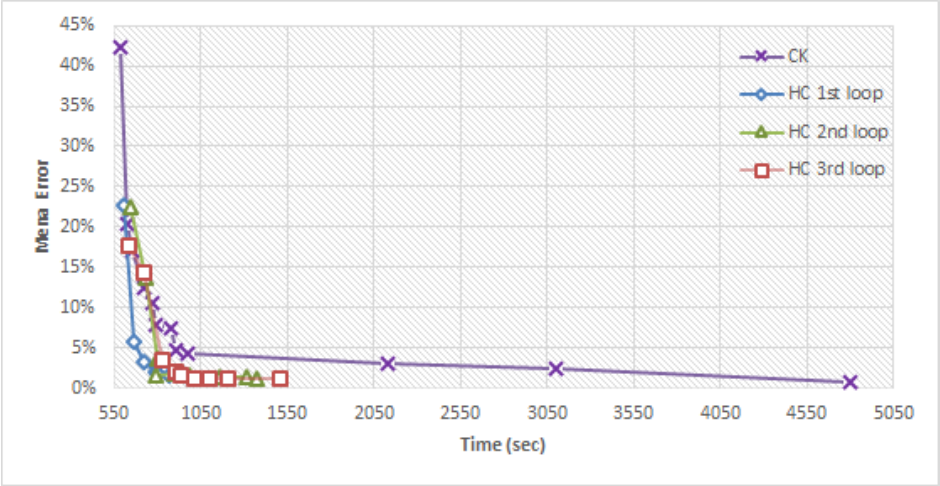


Figure 5.11: Computation time comparison between CK and HC for 1, 2 and 3 Basis Metamodel loops related to Cantilever Beam in stress results.

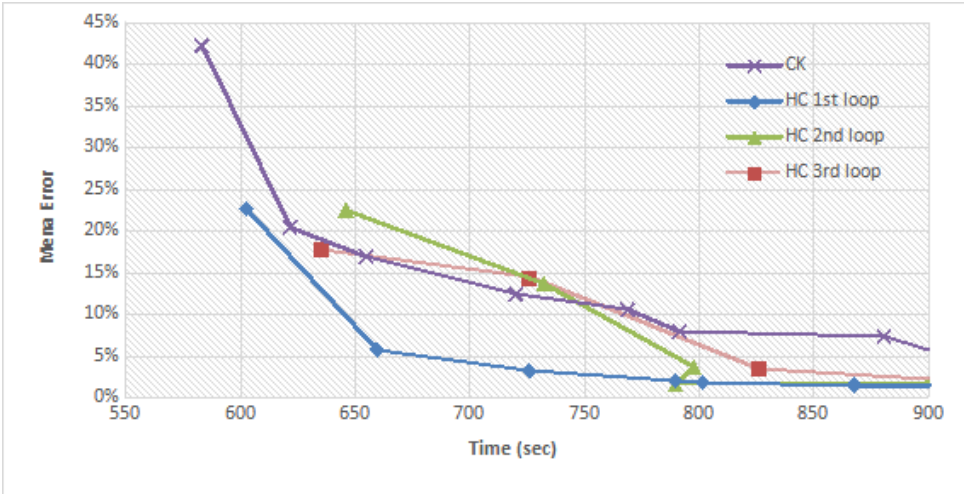


Figure 5.12: Computation time comparison ibetween CK and HC for 1, 2 and 3 Basis Mtamodel loops related to Cantilever Beam in stress results (time scale from 550s to 900s).

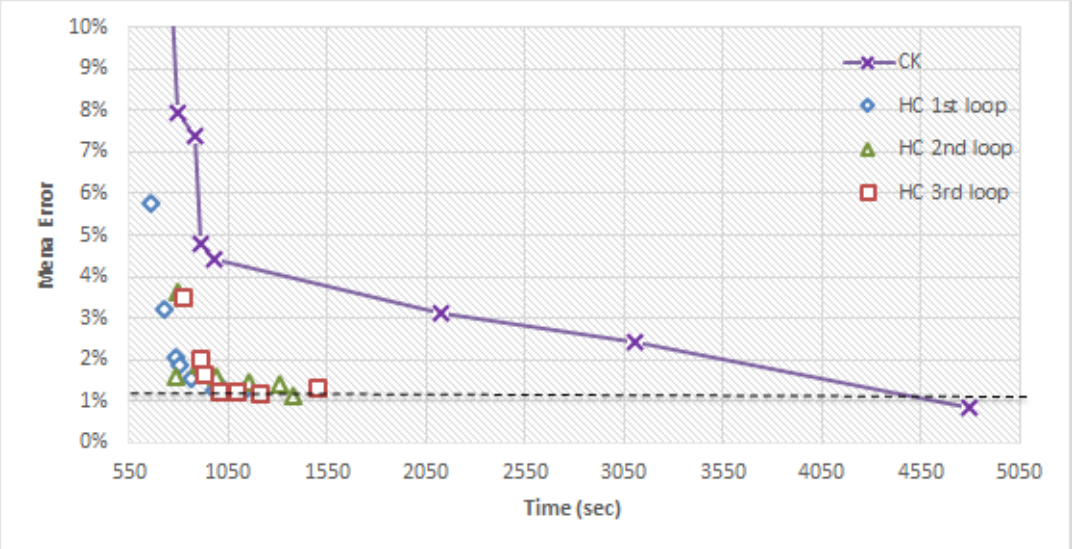


Figure 5.13: Computation time for a mean error of 1% between CK and HC for 1, 2 and 3 Basis Meamodel loops, related to Cantilever Beam in stress results.

5.2 Underwater Explosion

The UNDEX problem represents a specific class of engineering application since it involves a very high dynamic problem in a short physical time frame (order of miliseconds). In the following subsections, the FEM model is described and the HC performance is analyzed. The FEM model is simulated in ABAQUS solver.

5.2.1 FEM Model

In this FEM model, the explosion is modeled as a bubble of gas with high energy following the Geers et al. (2005) and Geers et al. (2002) works described in Hibbitt et al. (2018a), Hibbitt et al. (2018b) and eq. 5.1.

$$\ddot{V}(t) = \frac{4\pi a_c}{\rho_f} P_c[0.8251exp(-\frac{1.338t}{T_c} + 0.1749exp(-\frac{0.1805t}{T_c})], \tag{5.1}$$

Where:
$\ddot{V}(t)$ is the volume acceleration during the shock wave;
$T_c = km_c^{1/3}(a_c/m_c^{1/3})^B$;
$P_c = K(m_c^{1/3}/a_c)^{1+A}$;
$m_c,\ a_c$ are the mass and radius of the explosive charge;
$K,\ k,\ A\ and\ B$ are constants for the charge material;

ρ_f is the mass density of the fluid;
t is time.

Such bubble of gas produces an incident pressure wave in the fluid as modeled by eq. 5.2.

$$p_I(x_j, t) \equiv p_t(t) p_x(x_j), \tag{5.2}$$

Where:
$p_x(x_j) = \frac{1}{R_j}$;
$p_t(t) = \frac{\rho_f}{4\pi} \left(\frac{a_c}{R_j}\right)^A \ddot{V} \left(\left(\frac{a_c}{R_j}\right)^B t \right)$, for the shokwave phase assumed as $t_I < 7T_c$;
$p_t(t) = \frac{\rho_f}{4\pi} \ddot{V} = \rho_f (a^2 \ddot{a} + 2a\dot{a})$, for the oscillation phase assumed as $t_I \geq 7T_c$;
$R_j \equiv \| x_s - x_j \|$;
x_j - location in the domain where the load is applied;
x_s - location in the domain for the source point;
x_0 - standoff reference point, which is located inside of the computational domain and incorporates the time-history of the incident wave.

Integrating these equations through fourth-order Runge-Kutta integrator with variable time steps, the loading over some structure can be obtained. For this context, the present work considers such loading on a stiffened submerged plate as described in the references Woyak (2002) and Jen et al. (2010) and following the scheme presented in figure 5.14 considering a plate 4m x 4m clamped at its edges.

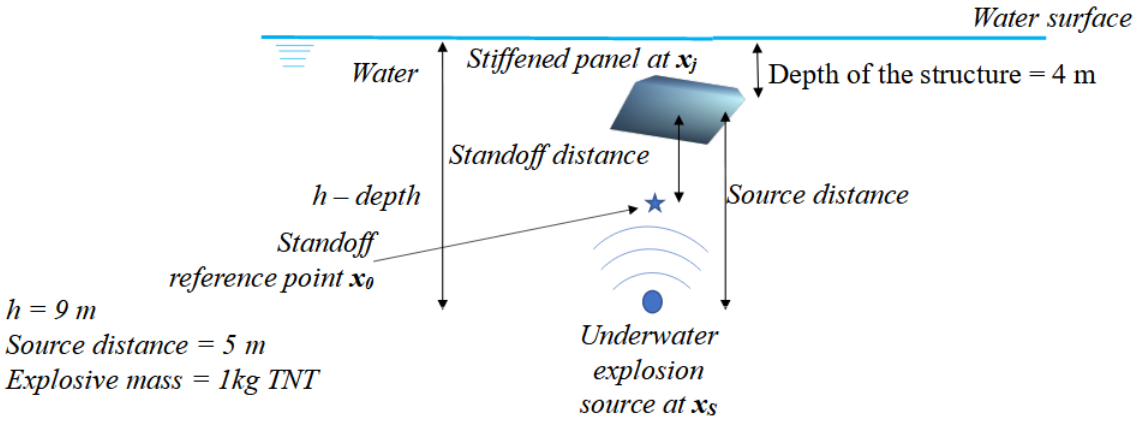


Figure 5.14: Global scheme of the UNDEX loading on a stiffened submerged plate.

The geometry (figure 5.15) is modeled in ABAQUS with two different FEM mesh elements: acoustic mesh for the fluid assuming 4-node linear acoustic tetrahedron element (AC3D4) and lagrangean mesh for the stiffined plate assuming 4-node doubly curved thin or thick shell, reduced integration, hourglass control and finite membrane strains (S4R), according to Hibbitt et al. (2018b). Both are indicated in figure 5.16.

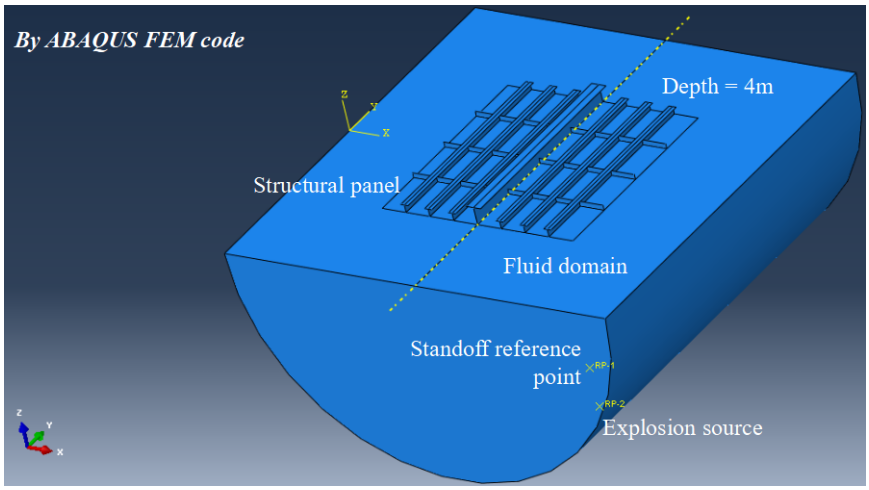


Figure 5.15: ABAQUS FEM model for UNDEX problem.

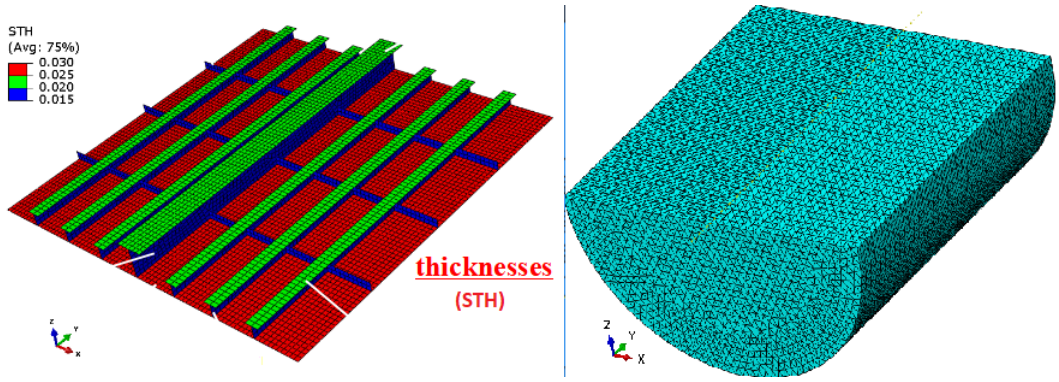


Figure 5.16: FEM mesh for the fluid and plate.

The model parameters are listed in table 5.2.

Table 5.2: FEM Model Parameters

PROPERTY/PARAMETER	MODEL INPUT VALUE
Fluid Properties	
Density	1025 kg/m^3
Bulk modulus	2140.4 MPa
Structure Properties	
Density	7850kg/m^3
Elasticity Modulus	200 GPa
Poisson ratio	0.3
Yield stress	270 MPa
Rupture Stress	480 MPa
Plastic strain limit	0.36
Explosion model Parameters	
Type of explosive charge	TNT
Speed of sound	1500 m/sec
Gas specific heat ratio	1.27
Gravity acceleration	9.81 m/sec^2
Atmospheric pressure	10e5 Pa
Flow drag coefficient	0.25
Flow drag exponent	2
K constant	52110000
k constant	9.95e-6
A constant	0.186
B constant	0.18
Kc constant	1045000000
Charge density	1654 kg/m^3
mass	1kg

The results form ABAQUS solver are presented in figures 5.17 to 5.18 for the fluid and structure, respectively.

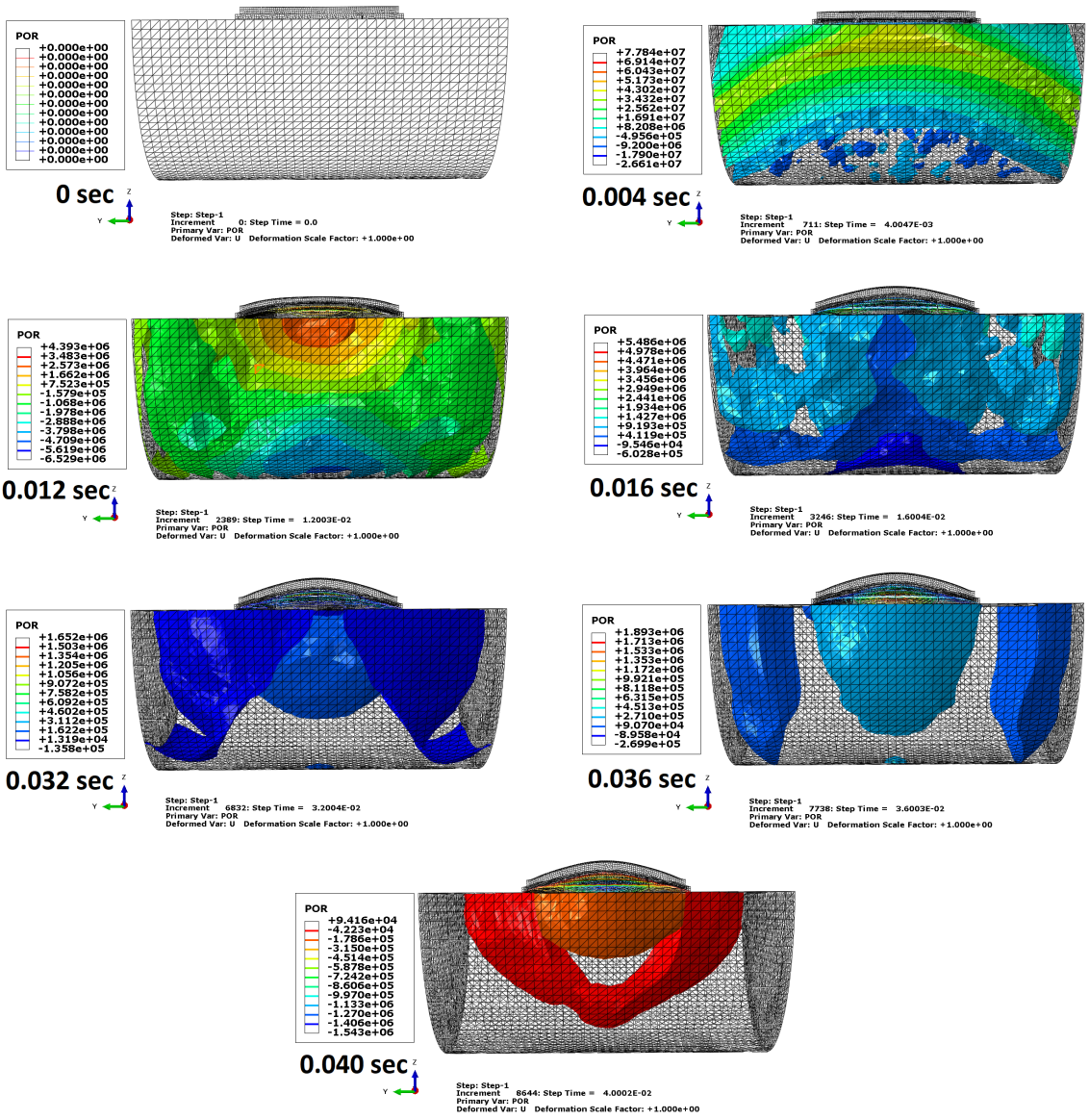


Figure 5.17: Isosurface of pressure field on the fluid acoustic elements

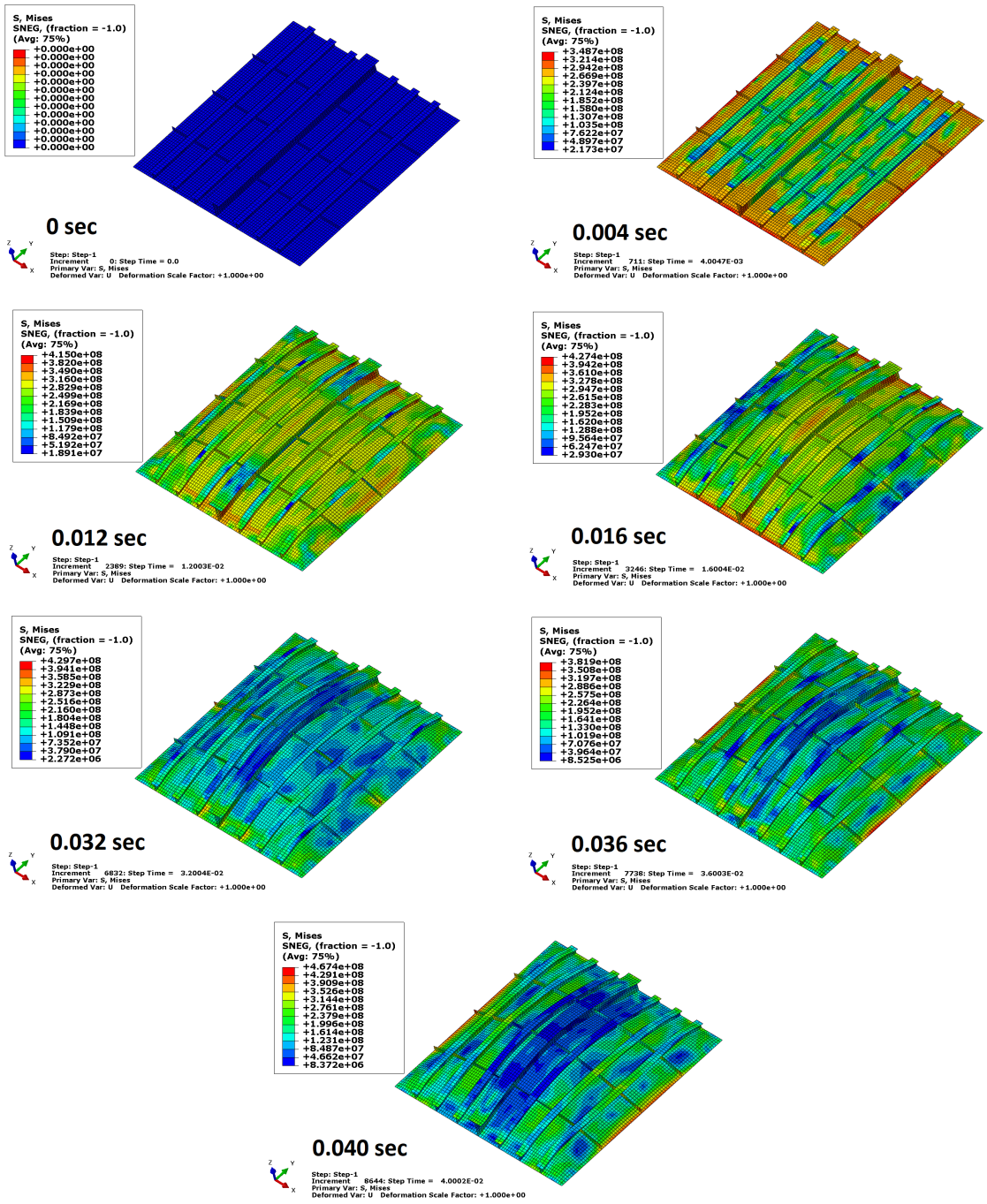


Figure 5.18: Maximum equivalent stress for the stiffened plate

Considering an specific element and node in the mesh of the plate (element 6991 and node 1443) and in the mesh of the fluid (node 127), it is possible to analyze the correspondent kinematic behaviour, which brings the respective time evolution according to figures 5.19 to 5.22.

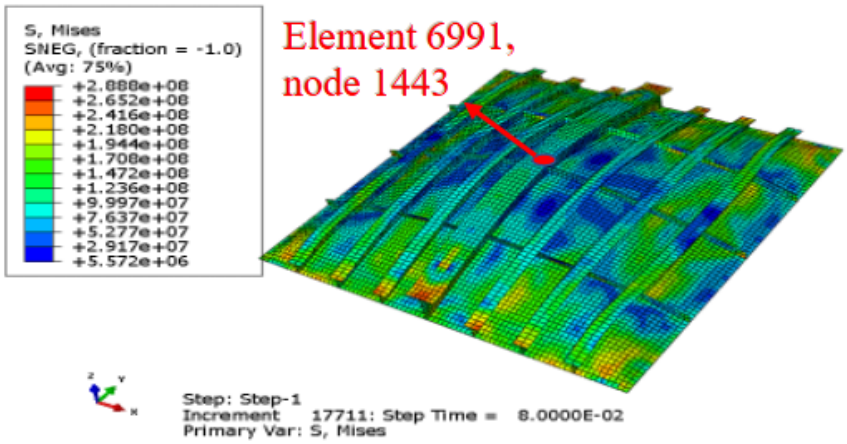


Figure 5.19: Stiffened plate and the element 6991 / node 1443

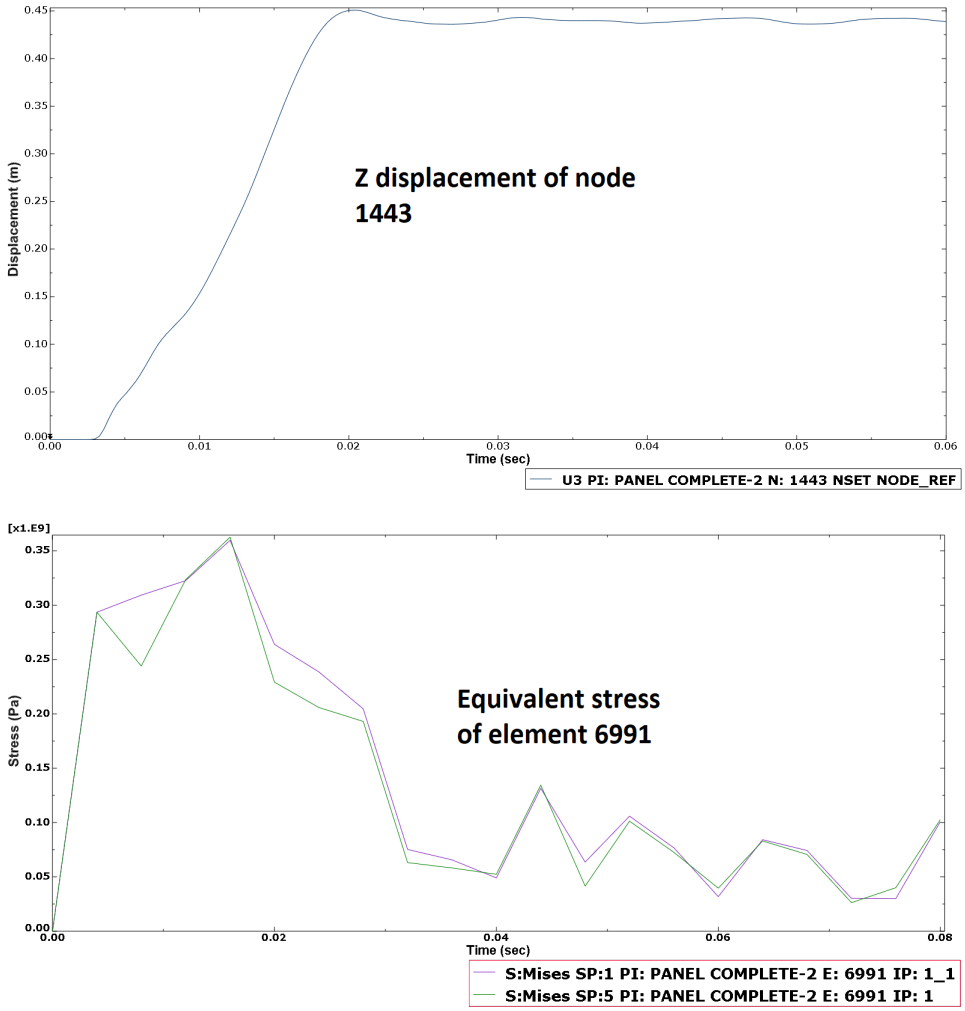


Figure 5.20: Z displacement for node 1443 and equivalent stress for element 6991 of the stiffened plate

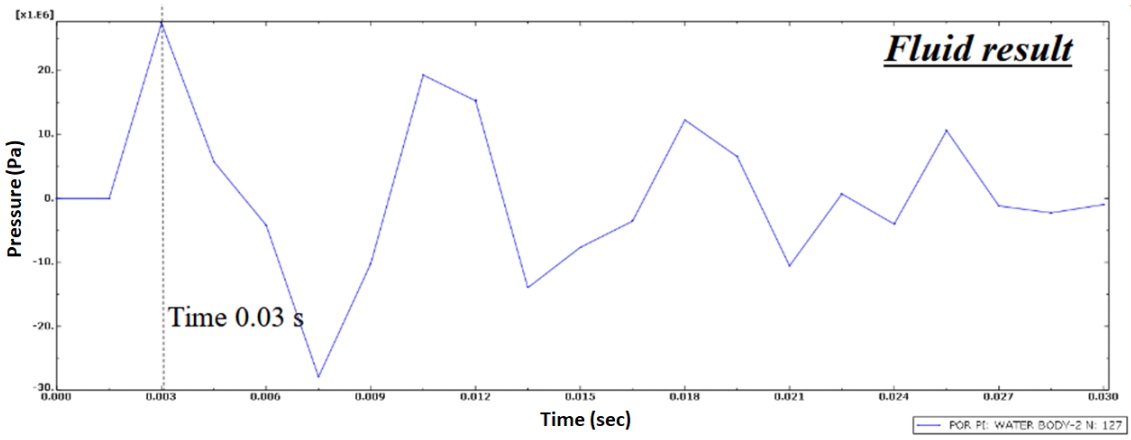


Figure 5.22: Pressure variation for node 127 of the fluid

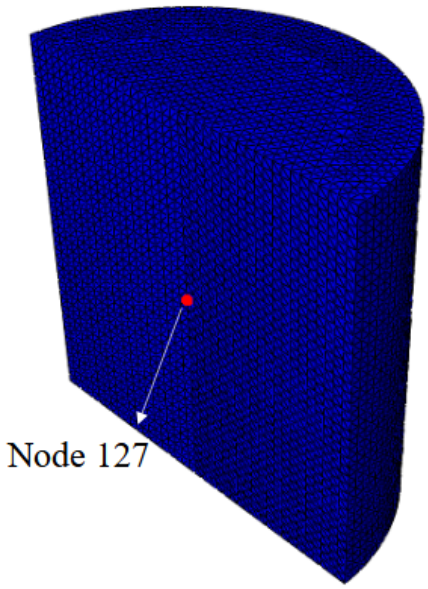
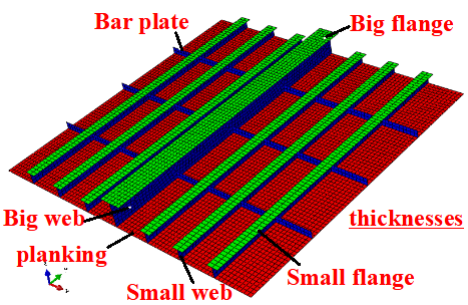


Figure 5.21: Fluid and node 127.

The vertical displacement of node 1443 of the stiffened plate is the output to be monitored. The thicknesses of the stiffeners are the variables, according to the range shown in table 5.3.

Table 5.3: Range of thicknesses adopted for each structural entity of the stiffened plate



Planking - 20mm to 40mm
Small web - 10mm to 25mm
Small flange - 15mm to 30mm
Big web - 10mm to 25mm
Big flange - 15mm to 30mm
Bar plate - 10mm to 25mm

The next subsection presents the results obtained by applying the CK and the HC procedures.

5.2.2 Results

Considering a set of initial sample and test points, the mean error performance can be evaluated by the CK and HC. Figure 5.23 presents the results for a total number of sample points between 29 and 33 points. The initial metamodel is generated by the CK procedure with 28 fixed sample points (Basis Metamodel). The HC procedure generates one new sample point at a time. The results are presented in figure 5.24 considering the 1 to 3 Basis Metamodel loops applied on HC.

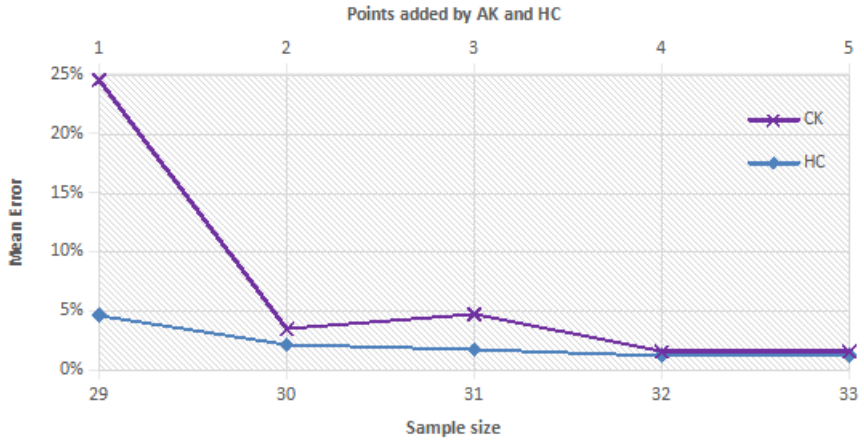


Figure 5.23: Mean error peformance for CK and HC related to UNDEX problem

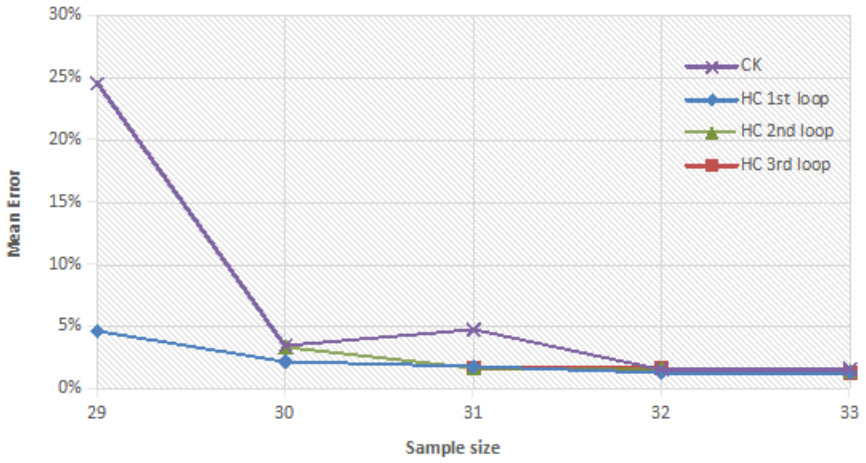


Figure 5.24: Mean error comparison between CK and Basis Metamodel loops for HC method related to UNDEX problem

The mean error obtained by HC in the scenario with 29 sample points has a magnitude 5 times smaller than the one obtained by CK. For more points added, the mean error difference reduces.

About the particular abrupt mean error drop $\bar{\epsilon}$ of CK and its fast agreement with HC starting from 30 points presented in figure 5.23 and 5.24, it can be explained through two main aspects. The first one is related to the Kriging convergence, which shall be compliant with the regression formulation stated in eq. 2.16. Considering the assumed regression model is quadratic, a minimum sample points p for a given quantity of variables n becomes necessary for proper Kriging convergence, as illustrated in Figure 5.25.

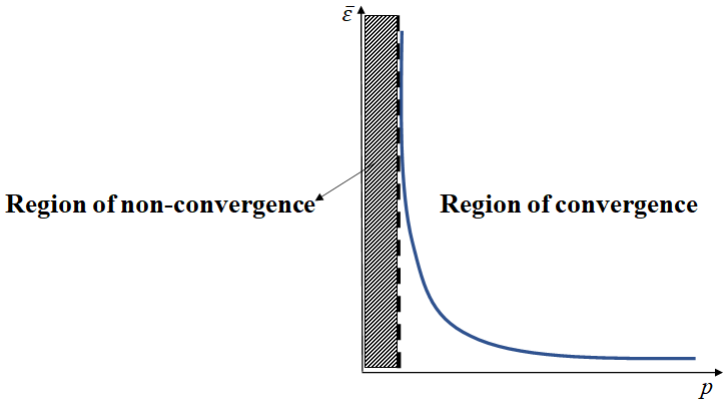


Figure 5.25: Region of convergence for Kriging quadratic regression model

For the current UNDEX problem and considering such formulation, six variables imposes 28 points as minimum necessary sample size, which is applied for the Basis Meta-

model generation. In this scenario and considering one additional point, CK demonstrates relatively higher mean error achieved by a poor predictor. Such behavior is not observed in HC results, which demonstrates lower mean error since the begin. Comparing both approaches, the smaller sample size corresponds to the interested scenario for HC.

The second aspect is related to the visualization of the samples resulted from CK and HC, which is presented in figure 5.26 and 5.27. The green circles are the 28 initial sample considered by the Basis Metamodel. The filled green points are the ones added by CK and the filled red points are the ones related to HC. Two different total sample sizes have been visualized: 29 and 30 points (1 and 2 added points by CK and HC). Considering that are six variables in such problem, the first variable has been fixed and all faces of the domain in relation to the first variable have been compared. What can be observed is the difference between both methods for 29 total sample size considering the majority of such views. The CK added sample is located in the interior of design space, while the HC sample is located at the edges. Considering 30 total sample points, the second CK sample is located closer to the edge approaching to the HC behaviour. This aspect agrees with the first aspect, considering the difficulty of prediction for CK assuming the minimum sample size.

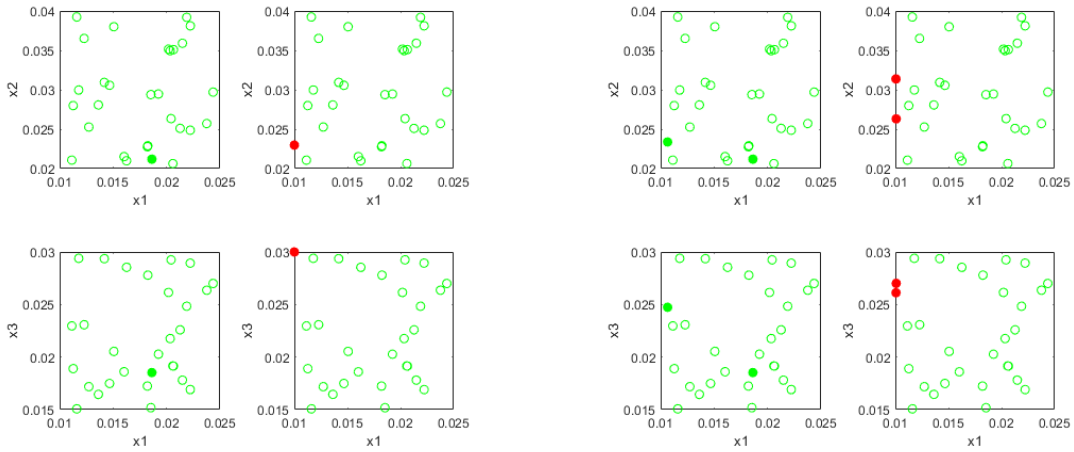


Figure 5.26: Sample comparison results from CK (filled green point) and HC (filled red point)

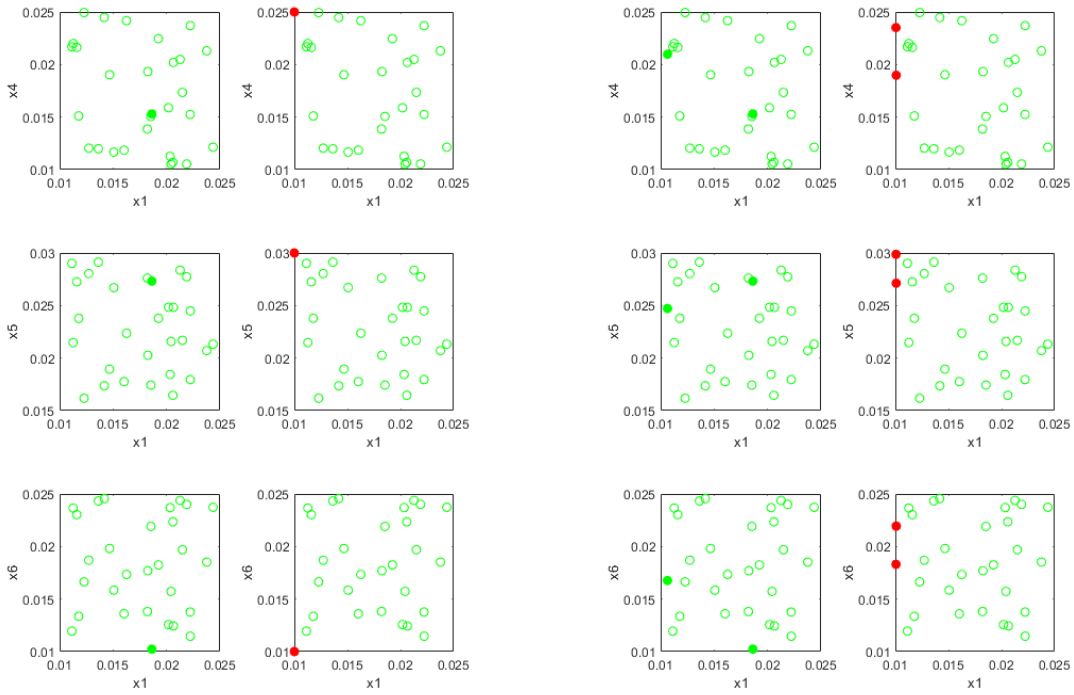


Figure 5.27: Sample comparison results from CK (filled green point) and HC (filled red point) - continuation

Regarding the computational time, HC also demonstrates to be more efficient than CK, obtaining a smaller mean error in less time for the single Basis Metamodel loop (as shown by the first three marks of the first loop on figure 5.28). As an example, for a 2% defined level of mean error, CK demands around 6950 sec, while HC reaches the same performance 350 sec before (figure 5.29). For two and three Basis Metamodel loops, the performance of HC presents the same order of magnitude for the performance of the single loop HC.

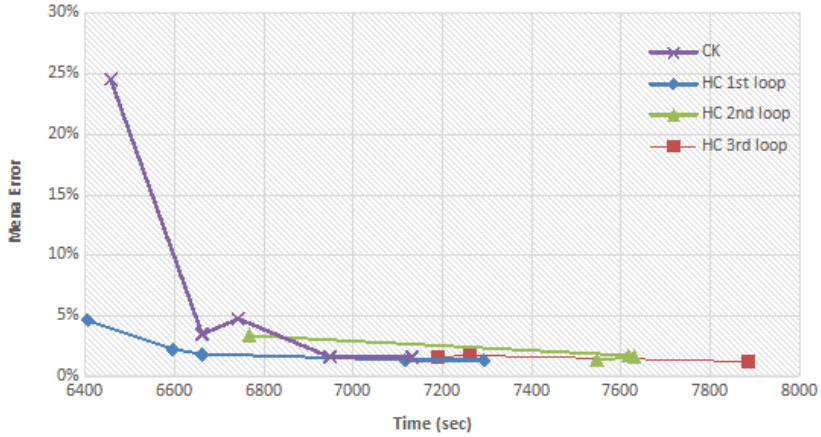


Figure 5.28: Computation time comparison between CK and HC for 1, 2 and 3 Basis Metamodel loops related to UNDEX problem

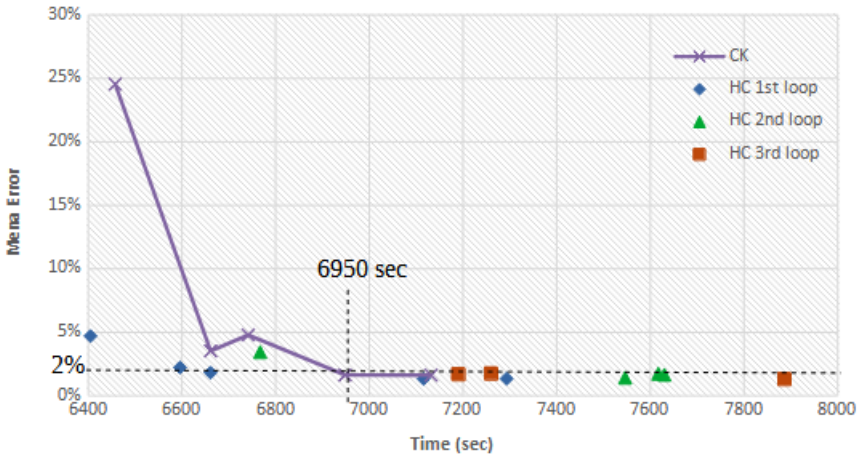


Figure 5.29: Computation time comparison between CK and HC with 1, 2 and 3 Basis Metamodel loops, for a 2% mean error, related to UNDEX problem

5.2.3 General comments

The mean error and the computational time of the HC procedure show a better performance than the ones from the CK procedure, according to the results obtained from the cantilever beam and UNDEX problems. Additionally, for the cantilever beam problem, increasing the number of variables improves even more the computational time performance of the HC for all Basis Metamodel loops compared to CK, as indicated in figures 5.8 and 5.13. For UNDEX problem, although the quantity of variables is less than the cantilever beam, HC still performs better than CK for single Basis Metamodel loop in terms of computation time reaching on the same conclusion obtained from previous problems. For mean error, all scenarios analyzed for HC reached a better behavior than CK.

A paticular discussion has been presented for the minimum sample size necessary to allow the Kriging convergence depending on the quantity of involved variables and the best HC performance for smaller samples when compared with CK, as observed in figures 5.23, 5.26 and 5.27.

Another relevant aspect is related to the complexity of the analyzed problems and the type of output considered. If the vetical displacement results presented in the figure 5.4 and 5.23 are compared, similar mean error drop from lower to larger sample size can be observed. This scenario is not quite the same when the stress result for cantilever beam is analyzed, which needs more samples to make the CK results approaching the HC performance. It can be explained through the typical stress behaviour that is more complex than

the displacement output. Although the particular UNDEX problem analyzed focuses only in the displacement result, the stress prediction achieved in the cantilever problem indicates the HC reaching better performance for more complex problems.

In respect of Basis Metamodel loops, as demonstrated in chapter 4, there is no significant improvement in using 2 and 3 loops over the single loop. In this context, as the computation time is the priority parameter, the obtained results support the use of the single loop HC as the most efficient procedure.

CHAPTER 6

Conclusion and Future Research

In this new proposed method, elements from the general metamodel obtained by the CK and its adaptive approach are combined with the evolutionary theory used in the NSGAII into a single algorithm. This new approach of smart sampling aims to use minimum sample data in order to generate an efficient metamodel with a better performance. Pursuing this main goal, the proposed algorithm brings a sort of achievements that is summarized in the next topic.

6.1 Achievements Overview of the New Approach

1. **Combining GA with AK algorithm in a single hybrid method**
   It has been the main contribution based on a singular combination between GA and AK to be applied on multivariable problems. Both procedures work in a collaborative way providing their best intrinsic aspects in a single algorithm aiming to achieve an optimum sample with relative low computation time and able to generate an efficient metamodel.

2. **Multivariable non-cartesian mesh based on zones of design space**
   For multivariable problems, a cartesian mesh is not time efficient due to the exponential increasing of points that are needed to cover all design space. In order to overcome this computation time difficulty, a random mesh separated in zones of design space through *kmeans* formulation is used. This original solution imposes a more efficient exploitation of a given domain searching for best location to add new sample points;

3. **Selection criteria for Adaptive Kriging**
   MSE parameters have been considered in the selection criteria for AK algorithm.

This aspect works as a filter to create the first population of points selected for each zone in the GA.

4. **Specific combination of objective functions implemented in NSGAII genetic algorithm**
   Due to the GA participation in the overall algorithm and the main goal to provide an efficient sample based on relative small sample size, a specific combination of objective functions based on MSE and dMSE has been implemented. In this sense, a gene selection has been implemented using the maximum MSE and low dMSE in order to elect one single point from each zone of design space;

5. **GA convergence**
   In order to mitigate the computation time and also taking into account the sampling objective, a new GA convergence tracking has been implemented considering the centroid of the population cloud obtained for each generation. This new approach reaches the convergence in a relative short time after few generations;

These five achievements match with the main contribution of this research and provide an overview about all aspects described in this dissertation. The next topic shows potential aspects that can be deeper investigated considering the data obtained from this current research.

6.2 Future Researches

From the contributions of this work, the following sugestions of future researches are presented:

1. **General learning machine improvement**
   The hybrid sampling described in this dissertation illustrates a learning machine process dedicated to provide a suitable sample able to build an efficient surrogate model. However, this process can be generalized on an algorithm able to predict any continuous physical behavior in classical mechanics and fields of physics. In this context, more complex physical models could be considered. The continous behavior is a constraint for this analysis as the Kriging method is based on a continuous response surface. If a discrete domain is object of analysis, another solver could be coupled with GA, such as SVM, RBF or Neural Network (NNW). Nevertheless, the global approach proposed in this work is still applicable for such scenario.

2. **Reducing of samples**

   One of the main goal pursued by HC is to provide minimum data able to generate an efficient surrogate model. Although the results from the examples of chapters 4 and 5 show a considerable reduction in the sample size, the proposed quantity of control points applied in problems that involve more variables could be not sufficient to reach a relative small sample. In this context, a more robust criteria to fill the cp mesh for larger quantity of variables than the problems explored in this dissertation could be investigated in future work. This process can contribute to the robusteness of cp mesh to more efficiently cover the design space and reduce the final sample size as a consequence.

3. **Order reducing**

   Another approach that could be added on this proposed method is the capability to reduce the order of a given problem considering only the most important variables. One technique available in the literature is the Principal Component Analysis (PCA). The main basis of this approach is to identify the strongest variables and to perform the truncation of the problem computing only the more relevant variables for the solver. The straighforward consequence is the reduction of the computational time. The disadvantage is a relative degradation of the final accuracy due to the truncation. In this context, the use of HC with an improved criteria for cp mesh selection mentioned in previous item could increase the global performance, covering only the important variables. This development can become a new approach to deal with higher complex multivariable problems that are usually constrained by large computational time.

BIBLIOGRAPHY

Abu-Lebdeh, G. and R. Benekohal (Dec. 2002). "Convergence Variability and Population Sizing in Micro-Genetic Algorithms". In: *Computer-Aided Civil and Infrastructure Engineering* 14, pp. 321–334. DOI: 10.1111/0885-9507.00151.

Arthur, D. and S. Vassilvitskii (2007). "Kmeans++: The Advantages of Careful Seeding". In: *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms* 07, pp. 1027–1035.

Baco, S. B., P. C. Oprime, L. Campanini, and G. M. D. Ganga (Aug. 2019). "Design of experiments used in computer trials: a supportive method for product development". In: *Brazilian Operations Research Society* 39, pp. 295–316. URL: http://www.scielo.br/scielo.php%5C?script=sci_arttext%5C&pid=S0101-74382019000200295%5C&nrm=iso.

Baquela, E. G. and A. C. Olivera (2019). "A novel hybrid multi-objective metamodel-based evolutionary optimization algorithm". In: *Operations Research Perspectives* 6.C. DOI: 10.1016/j.orp.2019.100098. URL: https://ideas.repec.org/a/eee/oprepe/v6y2019ics221471601830068x.html.

Bossong, C. R., M. R. Karlinger, B. M. Troutman, and A. V. Vecchia (Oct. 1999). "Overview and technical and practical aspects for use of geostatistics in hazardous-, toxic-, and radioactive-waste-site investigations". In: *Water-resources investigation - U.S. Army Corps of Engineers* 98.145.

Bouhlel, M. A. and J. R. R. A. Martins (2017). "Gradient-enhanced kriging for high-dimensional problems". In: *Computer Research Repository* abs/1708.02663. eprint: 1708.02663. URL: http://arxiv.org/abs/1708.02663.

Bouhlel, M. A., N. Bartoli, A. Otsmane, and J. Morlier (2016). "Improving kriging surrogates of high-dimensional design models by Partial Least Squares dimension reduction". In: *Structural and Multidisciplinary Optimization* 53.5, pp. 935–952.

Cressie, N. A. C. (1993). *Statistics for Spatial Data*. Ed. by I. John Wiley & Sons. Revised. Iowa State University. Wiley-Interscience publication.

Deb, K. and R. B. Agrawal (1995). “Simulated Binary Crossover for Continuous Search Space”. In: *Complex Systems* 9, pp. 115–148.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan (2002). “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *IEEE Transactions on Evolutionary Computation* 6.2, pp. 182–197.

Dellino, G., J. P. Kleijnen, and C. Meloni (2009). *Robust Optimization in Simulation : Taguchi and Krige Combined*. Discussion Paper 2009-82. Tilburg University, Center for Economic Research. URL: https://ideas.repec.org/p/tiu/tiucen/d919b893-db2b-4d97-a392-4a55e2ad2494.html.

Eby, W. D., C. R. Averill, F. W. Punch, and D. E. Goodman (1998). “Evaluation of Injection Island GA Performance on Flywheel Design Optimization”. In: *Adaptive Computing in Design and Manufacture*. DOI: https://doi.org/10.1007/978-1-4471-1589-2_10.

Elements Reference - ANSYS Manual (2007). ANSYS Inc.

Etman, L. (1994). *Design and analysis of computer experiments: the method of Sacks et al.* English. DCT rapporten. WFW 1994.098. Technische Universiteit Eindhoven.

Fernandez, P. M., I. V. Sanchis, V. Yepes, and R. I. Franco (June 2019). “A review of modelling and optimisation methods applied to railways energy consumption”. In: *Journal of Cleaner Production* 222, pp. 153–162. DOI: 10.1016/j.jclepro.2019.03.037.

Forrester, A. I. and A. J. Keane (Jan. 2009). “Recent advances in surrogate-based optimization”. In: *Progress in Aerospace Sciences* 45.1-3, pp. 50–79. DOI: 10.1016/j.paerosci.2008.11.001.

Geers, T. L. and K. S. Hunter (2002). “An Integrated Wave-Effects Model for an Underwater Explosion Bubble”. In: *Journal of the Acoustical Society of America* 111(4), pp. 1584–1601.

Geers, T. L. and C. K. Park (2005). “Optimization of the G & H Bubble Model”. In: *Shock and Vibration* 12(1), pp. 3–8.

Gu, J., W. Li, and Y. Cai (Oct. 2019). “A hybrid meta-model based global optimization method for expensive problems”. In: *Computers & Industrial Engineering* 136, pp. 421–428. DOI: 10.1016/j.cie.2019.07.044.

Hacker, K., J. Eddy, and K. Lewis (Sept. 2001). “Tuning a Hybrid Optimization Algorithm by Determining the Modality of the Design Space”. In: DOI: 10.1115/DETC2001/DAC-21093.

He, J. and G. Lin (2016). “Average Convergence Rate of Evolutionary Algorithms”. In: *IEEE Transactions on Evolutionary Computation* 20.2, pp. 316–321.

Hibbitt, K. and Sorensen (2018a). *Abaqus Analysis User's Guide 14 - 34.4.6 Acoustic and Shock loads*. Simulia.

Hibbitt, K. and Sorensen (2018b). *Abaqus Theory Guide 14 - 6.3.1 Loading due to an incident dilatational wave field*. Simulia.

Jen, C.-Y. and Y.-S. Tai (2010). "Deformation behavior of a stiffined panel subjected to underwater shock loading using the non-linear finite element method". In: *Materials and Design* 31, pp. 325–335.

Jeong, S., M. Murayama, and K. Yamamoto (Sept. 2005). "Efficient Optimization Design Method Using Kriging Model". In: *Journal of Aircraft* 42, pp. 1375–1375. DOI: 10.2514/1.17383.

Joe, J. C. (Mar. 2017). "A human factors meta model for U.S. nuclear power plant control room modernization". In: URL: https://www.osti.gov/biblio/1375221.

Jones, D. R., M. Schonlau, and W. J. Welch (Dec. 1998). "Efficient Global Optimization of Expensive Black-Box Functions". In: *Journal of Global Optimization* 13.4. DOI: 10.1023/A:1008306431147.

Kleijnen, J. P. C. and W. C. M. van Beers (2005). "Robustness of kriging when interpolating in random simulation with heterogeneous variances : some experiments". In: *European Journal of Operational Research* 165.3, pp. 826–834. DOI: 10.1016/j.ejor.2003.09.037.

Kleijnen, J. P. C. and R. Sargent (1997). *A methodology for fitting and validating metamodels in simulation*. Discussion Paper 1997-116. Tilburg University, Center for Economic Research. URL: https://ideas.repec.org/p/tiu/tiucen/b8c2217c-f02c-4786-b704-77587495aec8.html.

Kleijnen, J. P. C. (2017). "Design and Analysis of Simulation Experiments: Tutorial". In: *Simulation Foundations, Methods and Applications*. Cham: Springer International Publishing, pp. 135–158. DOI: 10.1007/978-3-319-64182-9_8. URL: https://doi.org/10.1007/978-3-319-64182-9_8.

Lee, K.-H., G.-I. Jeong, and S.-H. Lee (Sept. 2019). "An approximate optimization strategy using refined hybrid metamodel". In: *Computational Intelligence* 36.1, pp. 35–54. DOI: 10.1111/coin.12237.

Lee, K., G. Jeong, and S. Lee (Sept. 2019). "An approximate optimization strategy using refined hybrid metamodel". In: *Computational Intelligence* 36. DOI: 10.1111/coin.12237.

Liu, H., J. Cai, and Y.-S. Ong (May 2017). "An Adaptive Sampling Aproach for Kriging Metamodelling by Maximizing Expected Prediction Error". In: *Computers and Chem-*

ical Engineering 106.2, pp. 171–182. DOI: 10.1016/j.compchemeng.2017.05.025.

Liu, H., S. Xu, and X. Wang (June 2016). "Sampling Strategies and Metamodeling Techniques for Engineering Design: Comparison and Application". In: DOI: 10.1115/GT2016-57045.

Liu, Y. and M. Collette (Aug. 2014). "Improving surrogate-assisted variable fidelity multi-objectiveoptimization using a clustering algorithm". In: *Applied Soft Computing* 24, pp. 482–493.

Lloyd, S. P. (1982). "Least Squares Quantization in PCM". In: *IEEE Transactions on Information Theory* 28, pp. 129–137.

Lophaven, S. N., H. B. Nielsen, and J. Sondergaard (2002). *DACE, A Matlab Kriging Toolbox*. Tech. rep. IMM-TR-2002-12. Lyngby - Denmark: Technical University of Denmark.

Louis, S. J. and G. J. E. Rawlins (1992). "Predicting Convergence Time for Genetic Algorithms". In: *Foundations of Genetic Algorithms 2*. Morgan Kaufmann, pp. 141–161.

Maki, K., R. Sbragio, and N. Vlahopoulos (2012). "System design of a wind turbine using a multi-level optimization approach". In: *Renewable Energy* 43, pp. 101–110. DOI: https://doi:10.1016/j.renene.2011.11.027. URL: https://www.elsevier.com/locate/renene.

MATLAB Manual - k-means Clustering (2018). Mathworks.

Menz, M., C. Gogu, S. Dubreuil, N. Bartoli, and J. Morio (Dec. 2019). "Adaptive coupling of reduced basis modeling and Kriging based active learning methods for reliability analyses". In: *Reliability Engineering and System Safety*. DOI: 10.1016/j.ress.2019.106771.

Ming, L., Y. Wang, and Y. Cheung (2006). "On Convergence Rate of a Class of Genetic Algorithms". In: *2006 World Automation Congress*, pp. 1–6.

Pan, G., P. Ye, and P. Wang (2014a). "A Novel Latin Hypercube Algorithm via Translational Propagation". In: *The Scientific World Journal* 2014, pp. 1–15. DOI: 10.1155/2014/163949.

Pan, G., P. Ye, P. Wang, and Z. Yang (2014b). "A Sequential Optimization Sampling Method for Metamodels with Radial Basis Functions". In: *The Scientific World Journal* 2014, pp. 1–17. DOI: 10.1155/2014/192862.

Queipo, N. V., R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker (Jan. 2005). "Surrogate-based analysis and optimization". In: *Progress in Aerospace Sciences* 41.1, pp. 1–28. DOI: 10.1016/j.paerosci.2005.02.001.

Ran, R., W. Li, and Y. Li (Apr. 2018). “A two-level global optimization method based on hybrid metamodel for expensive problems”. In: *Advances in Mechanical Engineering* 10.4, p. 168781401876954. DOI: 10.1177/1687814018769542.

Ryu, J.-S., K.-J. Cha, and T. H. Lee (May 2002). “Kriging interpolation methods in geostatistics and DACE model”. In: *KSME International Journal* 16.5.

Sacks, J., W. J. Welch, T. J. Mitchell, and H. P. Wynn (1989). “Design and Analysis of Computer Experiments”. In: *Statistical Science* 4.4, pp. 409–435.

Sahin, C. S., S. Gundry, E. Urrea, M. Ü. Uyar, M. Conner, G. Bertoli, and C. Pizzo (2010). “Convergence analysis of genetic algorithms for topology control in MANETs”. In: *2010 IEEE Sarnoff Symposium*, pp. 1–5.

Simpson, T., L. Dennis, and W. Chen (2002). “Sampling Strategies for Computer Experiments: Design and Analysis”. In: *International Journal of Reliability and Application* 2, pp. 209–240.

Simpson, T. W., A. J. Booker, D. Ghosh, A. A. Giunta, P. N. Koch, and R.-J. Yang (2004). “Approximation methods in multidisciplinary analysis and optimization: a panel discussion”. In: *Structural and Multidisciplinary Optimization* 27.5, pp. 302–313.

Stark, D. R. and J. C. Spall (2002). “Computable rate of convergence in evolutionary computation”. In: *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002. (IEEE Cat.No.02EX5997)*. Vol. 1, pp. 88–93.

Streltsov, S. and P. Vakili (Sept. 1995). “Global Optimization: Partitioned Random Search and Optimal Index Policies”. In: *Manufacturing Engineering Department*.

Surjanovic, S. and D. Bingham (Sept. 2020). *Virtual Library of Simulation Experiments: Test Functions and Datasets*. Retrieved September 7, 2020, from http://www.sfu.ca/~ssurjano. URL: http://www.sfu.ca/~ssurjano.

Tarek, E.-m., A. Hopgood, N. Lars, and A. Battersby (Aug. 2006). “Hybrid Genetic Algorithms: A Review”. In: *Engineering Letters* 3(2).

Thurman, C. S. and J. R. Somero (2020). “Comparison of meta-modeling methodologies through the statistical-empirical prediction modeling of hydrodynamic bodies”. In: *Ocean Engineering* 210, p. 107566. DOI: https://doi.org/10.1016/j.oceaneng.2020.107566. URL: https://www.sciencedirect.com/science/article/pii/S0029801820305746.

Vlahopoulos, N. and D. Singer (2007). *Advanced System of Systems Design Capability*. Tech. rep. 1007-3. Fnal Report. Ann Arbor, MI - US: Department of Naval Architecture and Marine Engineering, University of Michigan. URL: https://apps.dtic.mil/dtic/tr/fulltext/u2/a464016.pdf.

Wang, G. G. and S. Shan (May 2006). "Review of Metamodeling Techniques in Support of Engineering Design Optimization". In: *Journal of Mechanical Design* 129.4, pp. 370–380. DOI: 10.1115/1.2429697.

Wang, L., X. Wang, K. Liu, and Z. Sheng (May 2019). "Multi-Objective Hybrid Optimization Algorithm Using a Comprehensive Learning Strategy for Automatic Train Operation". In: *Energies* 12.10, p. 1882. DOI: 10.3390/en12101882.

Woyak, D. B. (2002). "Modeling Submerged Structures Loaded by Underwater Explosions with ABAQUS/Explicit". In: *ABAQUS User's Conference*.

Yao, W., X. Chen, Y. Huang, and M. van Tooren (Apr. 2013). "A surrogate-based optimization method with RBF neural network enhanced by linear interpolation and hybrid infill strategy". In: *Optimization Methods and Software* 29.2, pp. 406–429. DOI: 10.1080/10556788.2013.777722.

Ye, P. (Nov. 2019). "A Review on Surrogate-Based Global Optimization Methodsfor Computationally Expensive Functions". In: *Software Engineering* 7.4, pp. 68–84. DOI: 10.11648/j.se.20190704.11.

Yu, X., J. Cao, H. Shan, L. Zhu, and J. Guo (2014). "An Adaptive Hybrid Algorithm Based on Particle Swarm Optimization and Differential Evolution for Global Optimization". In: *The Scientific World Journal* 2014, pp. 1–16. DOI: 10.1155/2014/215472.

Yusoff, Y., M. S. Ngadiman, and A. M. Zain (2011). "Overview of NSGA-II for Optimization Machining Process Parameters". In: *Advanced in Control Engineering and Information Science* 15, pp. 3978–3983.

Zhang, J., S. Chowdhury, and A. Messac (Jan. 2012). "An adaptive hybrid surrogate model". In: *Structural and Multidisciplinary Optimization* 46.2, pp. 223–238. DOI: 10.1007/s00158-012-0764-x.

Zhang, J., S. Chowdhury, J. Zhang, A. Messac, and L. Castillo (Mar. 2013). "Adaptive Hybrid Surrogate Modeling for Complex Systems". In: *AIAA Journal* 51.3, pp. 643–656. DOI: 10.2514/1.J052008.

Zhou, G., L. Duan, W. Zhao, C. Wang, Z. Ma, and J. Gu (July 2016). "An enhanced hybrid and adaptive meta-model based global optimization algorithm for engineering optimization problems". In: *Science China Technological Sciences* 59.8, pp. 1147–1155. DOI: 10.1007/s11431-016-6068-4.

Zhu, J., Y.-J. Wang, and M. Collette (May 2013). "A multi-objective variable-fidelity optimization method for genetic algorithms". In: *Engineering Optimization* 46.4, pp. 521–542. DOI: 10.1080/0305215x.2013.786063.