Future Air-Space Transportation Technology

Department of Mechanical and Aerospace Engineering

Faculty of Engineering

University of Strathclyde

High Dimensional Uncertainty Propagation for Hypersonic Flows and Entry Propagation

Martin Kubicek

Submitted in fulfilment of the requirements for the degree
of

Doctor of Philosophy

2017

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination which has led to the award of a degree.

Martin Kubicek
Glasgow, Scotland, June 2017

Acknowledgements

I wish to thank my advisor Dr. Edmondo Minisci for his continuing support and help during my Ph.D. time. Edmondo was always helpful and constant ally in times of troubles. He was always there to help me. However, my biggest thanks belong to him due to the trust and freedom he gave me to pursue my ideas. Our brainstorming seasons were extremely helpful and many nice ideas take their basis during them. Moreover, he was always there to ensure top quality of my results and that my ideas were clearly communicated.

Also, I would like to my colleague Dr. Piyush M. Mehta who was willing to test my code on various real life problems. From our cooperation, many interesting papers were written and wonderful ideas for future work were established. But my biggest thanks belong to him for his practical advices and help with paper publications. Also, from our cooperation, I learned many practical things about making my code really useful for common users as programmer does not usually see "all the small stuff around".

My large thanks belong to my colleagues and fellow Ph.D. student Alessandro Falchi, who made the DSMC simulation running and working flawlessly. Without his contribution and help, the interesting GOCE test case would not be possible. Moreover, he is wonderful friend, who was willing to help in desperate times and sacrifice his personal time to make things working.

My thanks belong also to my colleagues from department who always found time for a coffee or simple chat. It was wonderful to have you all around during nights spent at the department.

Without all of you, these times would not be such an interesting
adventure and wonderful experience.

This work is dedicated to my father Ivan Kubicek, who was one of the pioneers of hot air ballooning in the Czechoslovakia (currently the Czech Republic). He was my paragon during childhood and inspiration in my life, specially during my Ph.D. time. Unfortunately, he is no longer with us as he passed away many years ago.

You will be always remembered as a great man and wonderful father

Abstract

To solve complex design problems, engineers cannot avoid to take into account the involved uncertainties. This is important for the analysis and design of hypersonic objects and vehicles, which have to operate in extreme conditions.

In this work, two approaches for a high dimensional uncertainty quantification (UQ) are developed. The first approach performs a single-fidelity non-intrusive forward UQ, while the second one performs a multi fidelity UQ, as an extension of the first approach. Both methods are focused on real engineering problems and, therefore, appropriate heuristics are included to achieve an optimal trade-off between accuracy and computational costs.

In the first approach, the stochastic domain is decomposed into domains of lower dimensionality, and, then, each domain is handled separately. This is possible due to the application of the HDMR, which is here derived in a new way. This new derivation allowed to deduce important conclusions about the high dimensional modelling, which are used in the prediction scheme. This novel approach for the selection of the higher order interaction effects drastically reduce the required number of samples. In order to have optimally distributed samples for the problem of interest, the adaptive sampling scheme is introduced. Moreover, the multi-surrogate approach is introduced in order to improve the robustness of the method. The single-fidelity approach is tested on a debris re-entry case and the method is validated with respect to the MC simulation method.

In the second approach, the multi fidelity approach has been developed. In order to have the optimal combination of the low fidelity models, the power ratio approach is introduced. To correct the low fidelity model, the classical additive correction, adapted to work within the HDMR approach, is used. The multi-fidelity approach has been tested on the GOCE re-entry case, where the performed tests demonstrate the potentialities of the method.

Contents

List of Symbols and Acronyms

$f(\mathbf{x})$	function of interest
f_0	Mean value of $f(\mathbf{x})$ for ANOVA decomposition
$f_i(x_i)$	Contribution of variable x_i to function $f(\mathbf{x})$ for ANOVA decomposition
$f_{i,j}(x_i, x_j)$	Correlated contribution of variable x_i and x_j to function $f(\mathbf{x})$ for ANOVA decomposition
$^{\mathbf{c}}\mathbf{x}$	Central point for the cut-HDMR model
$^{c}x_i$	Central point of the random variable x_i for the cut-HDMR model
$p_i(x_i)$	Probability Density Function for variable x_i
$p_{i...j}(x_i, ..., x_j)$	Joint probability Density Function
μ	Total mean value of the function of interest
μ_t	Partial mean value of increment function t
σ^2	Total variance of the function of interest
σ_t^2	Partial variance of the increment function t
S_t	Sensitivity indices of the increment function t
n	Number of random variables for the function of interest
$\mathbf{C}$	Semi positive covariance matrix for the normal distribution
μ_{PDF}	Mean value of given distributions for the normal distribution
$dF_t/dF_{i...j}$	Increment function t / $i...j$
$dF_t^{LF}/dF_{i...j}^{LF}$	Low fidelity increment function t / $i...j$
$dF_t^{HF}/dF_{i...j}^{HF}$	High fidelity increment function t / $i...j$
z	Number of samples of MC simulation applied on the surrogate model of the increment function, $dF_k(x)$
s_D	Selected number of increment functions

$SdF_t^k(x)/SdF_{i...j}^k(x)$ Surrogate model of the increment function $dF_t(x)/dF_{i...j}(x)$ in iteration k

$F_{pos}(\tilde{x})$ Position function

$F_{ND-pos}(\tilde{x}_i, ..., \tilde{x}_j)$ Multi dimensional Position function

d Maximum selected derivative order

$\tilde{x}$ Normalized stochastic random variable

k_{shift} Shift coefficient

$\tilde{\mathbf{x}}_{\mathbf{k}}$ Vector of normalized samples used to create surrogate model $SdF(x)$

$\epsilon^k(x_i)$ Error Comparison function in iteration k

$\epsilon_{ND}^k(x_i, ..., x_j)$ Multi Dimensional Error Comparison function in iteration k

$\tilde{\epsilon}^k(x)$ 1-D Probability Error Comparison function in iteration k

$E_{ND}^k(x_i, ..., x_j)$ N-D Probability Error Comparison function in iteration k

$\boldsymbol{\epsilon}_{ND}^k$ Vector of MDEC evaluations in iteration k

$\mathbf{E}_{ND}^k$ Vector of MD-PEC evaluations in iteration k

$\mathbf{x}_{new}^k$ New sample proposed with the adaptive algorithm in iteration k

$\mathbf{rx}_i$ Vector of random numbers distributed accordingly to the input distribution, p_i

$\mathbf{rx}_{\mathbf{i...j}}^{\mathbf{ND}}$ Vector of joint probability density according to vectors $\mathbf{rx}$

$\mathbf{Y}_i^k$ Vector of surrogate responses for variable i in iteration k using input vector $\mathbf{x}_i$

$\delta_D \mathbf{Y}_i^k$ Vector of surrogate derivatives of the order D for variable i in iteration k using input vector $\mathbf{x}_i$

$\mathbf{F}_{pos}$ Vector of values of the position function obtained for vector $\tilde{\mathbf{x}}$

$\tilde{h}_i^p$ Normalized histogram created from vector $\mathbf{rx}_i$

$\tilde{h}^p$ Normalized histogram created from vector $\mathbf{rx}^{ND}$

n_s Length of vector $\mathbf{rx}_i$

$R_{\mu_t}^k$ Normal residual of the mean in iteration k for increment function t

$R_{\sigma_t}^k$ Normal residual of the variance in iteration k for increment function t

$LR_{\mu_t}^k$ Logistic residual of the mean in iteration k for increment function t

$LR_{\sigma_t}^k$ Logistic residual of the variance in iteration k for increment function t

$R_{set_t}^{\mu}$ Set residual of the mean for the local convergence process of the increment function t

$R^{\sigma}_{set_t}$	Set residual of the variance for the local convergence process of the increment function t
GR^{μ}_{set}	Global residual for the mean
GR^{σ}_{set}	Global residual for the variance
V_i	Vector of samples used for variable i
M_{total}	Set of all increment functions
$M_{selected}$	Set of selected increment functions
$M_{total:j}$	Set of all increment functions, which does not include the increment functions having x_j as a variable
$M_{selected:j}$	Set of selected increment functions, which does not include the increment functions having x_j as a variable
$dF^j(\mathbf{x})$	Increment function without variable x_j
$dF_{Final:total}(\mathbf{x})$	Model composed of all increment functions
$dF_{Final:selected}(\mathbf{x})$	Model composed of selected increment functions
$dF_{neglected}(\mathbf{x})$	Model composed of neglected increment functions
$dF_{Final:selected:j}(\mathbf{x})$	Model composed of selected increment functions without variable x_j
$dF_{neglected:j}(\mathbf{x})$	Model composed of neglected increment functions without variable x_j
$dF_{not-included:j}(\mathbf{x})$	Model composed of neglected increment functions with variable x_j
$\epsilon_{Residual:total}(\mathbf{x})$	Function of residuals including all neglected increment functions
ϵ_{R-T}	Deduction residual of the higher order increment functions for $\epsilon_{Residual:total}(\mathbf{x})$
$k_{\epsilon_{R-T}}$	Correction factor for the deduction residual
$s_{D:j}$	Number of increment function in set $M_{selected:j}$
$\epsilon_{Residual:total:j}(\mathbf{x})$	Function of residuals including all neglected increment functions without x_j as a functional variable
ϵ_{R-T-j}	Deduction residual of the higher order increment functions for $\epsilon_{Residual:total:j}(\mathbf{x})$
$\epsilon_{Residual-not-included:j}(\mathbf{x})$	Function of residuals including all neglected increment functions with x_j as a functional variable
$\epsilon_{Residual-not-included:j}$	Deduction residual of the higher order increment functions for $\epsilon_{Residual-not-included:j}(\mathbf{x})$
pS_{dF_t}	Sensitivity criteria for increment function dF_t used in the prediction algorithm
$\epsilon_{prediction}$	Prediction residual for the higher order increment functions
$\epsilon_{neglection}$	Neglection residual for the higher order increment functions

${}^{min}p_{x_i}$	Lower bound of the probability distribution p_{x_i}
${}^{max}p_{x_i}$	Upper bound of the probability distribution p_{x_i}
${}^{1}p_{x_i}$	Lower bound of the allowed probability for variable x_i
${}^{2}p_{x_i}$	Upper bound of the allowed probability for variable x_i
$F_{ResidualModel}(x_1, ..., x_n)$	Linear model of residuals
cl_i	Coefficient for the linear model of residuals
$\boldsymbol{\sigma}_i$	Vector of the partial standard deviations of all involved increment functions with variable i as a functional variable
CL	Coefficient for establishing the linear model of residuals in a case of insufficient number of samples
$\sigma_{neglected}$	Statistical properties of the linear model of residuals
$\sigma_{neglected:2}$	Statistical properties of the linear model of residuals without variable x_j
$sf(\mathbf{x})$	Surrogate model
$sf_i(\mathbf{x})$	Surrogate model i
$\mu(\mathbf{x})$	Weighted mean model
wm_i	Normalized weight coefficient for surrogate model i
N_m	Number of active surrogate models
$w_{f_i}(\mathbf{x})$	Correction function for the i-th surrogate model
K	Number of samples in given iteration
$wmns_j^i$	Normalized weight of sample j in current iteration for selected model i
wms_j^i	Non-normalized weight of sample j in current iteration for selected model i
y_j^i	Difference between the function value and surrogate model i
σ_{MS}	Scaling coefficient
σ_{MS-p1}	Standard scaling coefficient
σ_{MS-p2}	Modified scaling coefficient
$Twf(x)$	Time weight function for the multi surrogate approach
$\epsilon_{Threshold}$	Neglection threshold for the multi surrogate approach
ϵ_T	Empirical coefficient for the multi surrogate approach
$\mathbf{yl}_j$	Vector of surrogate responses for sample j
$\mathbf{Rl}_j$	Vector of surrogate accuracies for sample j

$\mathbf{Rl}_j^L$	Vector of surrogate accuracies for sample j considering only negative accuracies
$\mathbf{Rl}_j^R$	Vector of surrogate accuracies for sample j considering only positive accuracies
w_μ^j	Local weight for the mean model at sample j
$w_{f_i}^j$	Local weight for surrogate model i at sample j
MRL_j	Accuracy of the closest model from set $\mathbf{Rl}_j^L$ at sample j
MRR_j	Accuracy of the closest model from set $\mathbf{Rl}_j^L$ at sample j
$f_{s:1}$	Surrogate model selected with MRL_j at sample j
$f_{s:2}$	Surrogate model selected with MRR_j at sample j
$w_{f_{s:1}}^j$	Local weight for the surrogate model selected with MRL_j at sample j
$w_{f_{s:2}}^j$	Local weight for the surrogate model selected with MRR_j at sample j
$\mathbf{w_f}^j$	Vector of local weight coefficients for the multi surrogate approach
$\theta_i(\mathbf{x}, \mathbf{x_c})$	Correlation function for surrogate model i with $\mathbf{x_c}$ determining center of the correlation function
$\mathbf{w_f}^j$	Vector of local weight coefficients for the multi surrogate approach
$\mathbf{J}$	Vector of samples, where each sample consist of a set of local weights $\mathbf{w_f}^j$
K_T	Number of samples in set $\mathbf{J}$
$r_i(\mathbf{x})$	Correlation vector for surrogate model i
$sf_{IPI}(\mathbf{x})$	Independent polynomial interpolant
T_c	Number of samples in given stochastic domain for IPI
$\overline{w_r(\mathbf{x})}$	Weight function for sample r
$pf_r(\mathbf{x})$	Independent surrogate model for sample r
$L2$	L2 Norm
N_{MC}	Number of samples for MC simulation in L2 Norm
P^{MF}	Power ratio
A^{MF}	Accuracy of the low fidelity model against the high fidelity model
T^{MF}	Time ratio of the low fidelity model and the high fidelity model
t^L	Computational time of the low fidelity model
t^H	Computational time of the high fidelity model
$f^{HF}(\mathbf{X})$	High fidelity function of interest
$f^{LF}(\mathbf{X})$	Low fidelity function of interest

$\epsilon(\mathbf{X})/A(\mathbf{X})$	Additive correction
$B(\mathbf{X})$	Multiplicative correction
μ_{t_ϵ}	Partial expected value of FEIF
$\sigma^2_{t_\epsilon}$	Partial variance of FEIF
$M^{i^{MF}}_{Low}$	Set of selected increment functions from the low fidelity model i^{MF}
M_{high}	Set of selected increment functions from the high fidelity model
$M^{i^{MF}}_{Low:Consider}$	Set of selected increment functions used for multi fidelity uncertainty propagation
σ_p^{HF}	Deduced statistical properties of the high fidelity model
$\sigma_p^{LF^i}$	Deduced statistical properties of the i-th low fidelity model
$R^{MF^{i^{MF}}_L}_{\mu_t}$	Residual of the mean value for the i-th low fidelity model
$R^{MF^{i^{MF}}_L}_{(\sigma_t)^2}$	Residual of the variance for the i-th low fidelity model
$\mathbf{f^{HF}}$	Vector of responses from the high fidelity model
$\mathbf{f^{LF}}$	Vector of responses from the low fidelity model
$\boldsymbol{\epsilon}$	Vector of corrections
${}^p\mathbf{x}$	Sample with known correction
$Sd\epsilon_{i...j}(\mathbf{x})$	Surrogate model for FEIF
$Sd\epsilon_{i...j}(\mathbf{x})$	Surrogate model for FEIF
${}^pdFE_t(\mathbf{x})$	Increment function t of the prediction error function at sample ${}^p\mathbf{x}$
${}^pFE_t(\mathbf{x})$	Prediction error function at sample ${}^p\mathbf{x}$ for increment function t
$S^{MF}_{T:t}$	Total volume of the predicted error for increment function t
$S^{MF}_{S:t}$	Influenced volume of the predicted error for increment function t
h	Altitude of debris
V_∞	Speed of the object
γ	Speed of the object
D	Drag force
g	Gravitational acceleration
g_0	Gravitational constant, $g_0 = 9.81$
ω_E	Earth's rotational speed
R_E	Radius of the Earth

γ	Path direction angle
χ	Latitude position
λ	Longitude position
C_p	Local pressure coefficient
θ	Maximum pressure point coefficient
C_τ	Shear coefficient
σ_N	Normal momentum accommodation coefficient
σ_T	Tangential momentum accommodation coefficient
T_w	Surface or body wall temperature
$erf()$	Error function
s	Speed ratio
R_U	Universal gas constant
sig_{10}	Sigmoid function
T_∞	Free stream temperature

CAD	Computer Aided Design
CV	Cross Validation
CoD	Curse of Dimensionality
DE	Derivative Equation
DSMC	Direct Simulation Monte Carlo
EGG	Electrostatic Gravity Gradiometer
ESA	European Space Agency
FEIF	Fidelity Error Increment Function
FOSTRAD	Free Open Source Tool for Re-entry of Asteroids and Space Debris
FMF	Free Molecular Flow
GPS	Global Positioning System
HDMR	High Dimensionality Model Representation
i-gPC	Iterative generalized Polynomial Chaos

IPI	Independent Polynomial Interpolation
JPDF	Joint Probability Distribution Function
MC	Monte Carlo
MDEC	Multi Dimensional Error Comparison
MD-PEC	Multi Dimensional Probability Error Comparison
ME-gPC	Multi-Element generalized Polynomial Chaos
ODE	Ordinary Differential Equations
PDE	Partial Differential Equations
PC/gPC	Generalized Polynomial Chaos
PDF	Probability Distribution Function
SBO	Surrogate Based Optimization
SSTI	Satellite-to-Satellite Tracking Instrument
UQ	Uncertainty Propagation
UQ-HDMR	Uncertainty Quantification - High Dimensionality Model Representation

List of Figures

List of Tables

Chapter 1

Introduction

”As we know there are known knowns. There are things we know we know. We also know there are known unknowns. That is to say, we know there are some things we do not know. But there are also unknown unknowns, The ones we don’t know we don’t know.”

D. Rumsfeld, Feb. 12, 2002, Department of Defense news briefing

Modern engineering problems require new ways of solution to introduce higher quality computing. It is common problem that performed simulations do not have required accuracy, yet the physical or mathematical model is valid. Moreover, the problem is even more complicated as one is never sure that a given model is physically fully correct. These errors are usually created with model’s assumptions, which are not fulfilled or with inaccurately measured inputs. One of the possible ways to bring computer simulations closer to reality is the application of a statistical approach.

Mathematical models are usually created and calibrated in laboratory environment. However, a real industry equipment differs from the laboratory one and such that the mathematical models have to tackle this difference. Moreover, many engineering models are simplified in order to get the model applicable to real problems. This represents another source of possible errors and the statistical approach can highlight possible errors in the computational approach. It takes into account all the small things, which are normally neglected in the process of model creation. This includes things such as production errors, atmospheric change or simple user errors can be considered in mathematical models when the statistical approach is used.

The uncertainty quantification or uncertainty propagation is extremely important for the industry. As mentioned above, mathematical models differ from the reality and one needs to estimate the influence of these uncertainties on the accuracy of the model. From the industrial point of view, uncertainty quantification ensures that the designed product is going to have the same properties as the produced part. Also, it helps to make the final product resistant to deviations from the designed solution. However, to have a method which is applicable in real industrial problems, it is necessary to perform simulations in a fast way.

1.1 Motivation

To bring computational simulations closer to reality, it is necessary to take into account uncertainties in the modelling process. However, the problem is complicated as it is necessary to take into account physical errors in the model of interest. Currently, we define a two main kinds of uncertainties:

- Aleatoric: Known as statistical uncertainty, which represents the unknown that differ each time an experiment is performed. These uncertainties are inherent to the system and such that it is hard to reduce these uncertainties.
- Epistemic: Known as systematic uncertainty and this type of uncertainty could be known in principle but it is not in practice. The missing knowledge about the problem can originate from an inaccurate measurement or the model neglects certain effects.

In reality, both uncertainties are present; however, the developed method aims to quantify the aleatoric uncertainty.

The common approach used in the past was to assume a Gaussian distribution around the final result. It was assumed that the Gaussian distribution considers all aspects such as the physical errors in the model or errors in the measurement. However, it was showed in many cases that this assumption is not correct. To overcome this problem, it is necessary to implement statistical methods. An example of necessity of statistical simulations is given in the work of Mehts et al. (2)

One of the most simple statistical approaches is the Monte Carlo (MC) simulation. It represents thousands or millions of runs with randomly modified inputs.

However, this approach is extremely expensive and in many engineering problems not even possible as one run of a complex engineering code can take days. Nevertheless, MC simulation still represents an easy to implement and very robust method, where nice overview of MC methods is given in the work of Veach (3).

To overcome the computational burden, many new engineering approaches are developed. These approaches try to propagate the uncertainty in a more sophisticated way such as using simplified models, surrogate models or combination of multiple fidelity models. A nice example of the multi fidelity modelling is shown in the work of Ng and Eldred (4). However, the process of uncertainty quantification is more complex from the engineering point of view. It is necessary to take into account timing of the performed simulation, complexity of the problem and also required accuracy.

Indeed, in many engineering problems, it is not necessary to perform extremely accurate simulations. The extremely accurate simulations are extremely complex to run and they require a complex programming skills yet the gain value is minimal. Therefore, a trade off between cost and gain has to be made. This can be graphically shown in the following way:

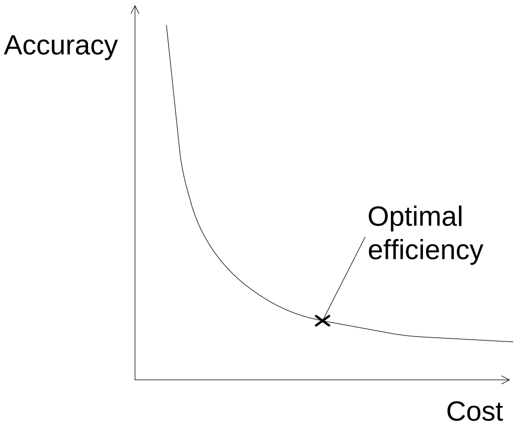


Figure 1.1: Efficiency ratio

One needs to find the optimal point for his/her problem, where the accuracy of the solution is sufficient and the cost of the problem is minimal. However, this is a very delicate task as it depends on many factors. One of the ways how to find such a point is the application of multiple physical models, which differs in fidelity, i.e. complexity and time of the solution. Another way is to use new progressive ways of uncertainty propagation.

1.2 Targets of thesis

The main target of the thesis is to perform the efficient propagation of uncertainty quantification. We try to establish a methodology, which could be widely applied for complex engineering models such as DSMC or CFD simulations. However, we do not want an extremely accurate uncertainty propagation method as obtaining extremely high relative accuracies (e.g. $< 10^{-10}$) represents a large computational burden. In the case of an extremely accurate result, the effective gain in the outcome is extremely small. In other words, the cost of the highly accurate result is not worthy. In this work, we focus on the effective method, which provides a result below a desired accuracy and yet it is still computationally feasible. Therefore, we restrict ourselves to accuracies, which are sufficient yet do not require large computational times.

One of the ways to perform uncertainty propagation and obtain a widely applicable approach is using non-intrusive approach. Non-intrusive techniques consider the code of interest as a black box. In other words, the connection between the uncertainty propagation tool and the code of interest is only through set of inputs and outputs. This step ensures that the developed tool can be widely used and it is not restricted to only one possible application.

The non-intrusive approach requires the definition of a set of inputs and outputs, i.e. the black-box function values. However, these methods suffer from the Curse of Dimensionality (CoD) with respect to the set of inputs, which states that the number of required samples (functional values) grows exponentially with additional parameters introduced into a problem. This leads to a dramatic increase in the computational time and restricts the usage of the non-intrusive approaches to a small number of parameters. One of the ways to decrease the computational time is to use the multi fidelity approach.

The multi fidelity approach considers multiple models, which differs with accuracy and mainly computational time. Therefore, with a proper combination of models and a proper correction approach, one can obtain an accurate result with significantly lower computational time. However, this requires a methodology capable of comparing accuracies and also the computational times of given models.

The multi fidelity approach is not the only possible way to reduce the computational time. With the proper simplification of the stochastic domain, one can gain a significant reduction in the computational time. However, this requires a new and robust approach to current problems. Moreover, a proper combination of multi fidelity modelling and new approach for UQ can bring large computational problems to a point, where accurate results are achievable with commodity hardware.

Another important thing in the uncertainty propagation is the automation of the process. The proposed method must be fully automatic and call the user in outer limits. In other words, the method must start and adaptively adapt to the problem of interest. This is necessary as many engineers are experts in their field but have limited knowledge about the uncertainty propagation or non-intrusive modelling. Therefore, the proposed method must handle all these aspects for the user.

This requires a robust non-intrusive approximation model, which is capable of handling all aspects. However, there is always a trade off between accuracy, efficiency and robustness. Therefore, we focus on an efficient combination of accurate and robust interpolation technique. This represents a real challenge in the uncertainty propagation field.

1.3 Structure of thesis

The work is structured in the following way:

In the next chapter, after a brief overview on sensitivity and uncertainty quantification approaches, the cut-HDMR approach used in this work is derived in a new way, which allows to derive new statistical conclusions about the nature of the posterior probability density function. The new derivation of the cut-HDMR approach also allows to establish a new interpolation scheme, where the stochastic domain is separated into smaller domains and for each domain, the associated function can be interpolated with an independent interpolation technique. These associated functions, called increments or increment functions, are defined on the basis of levels, where the first level represents all the functions corresponding to 1-D domains, the second level represents all the functions corresponding to 2-D domains, etc.. Moreover, since the proposed interpolation scheme allows to visualize the contribution of each variable (or variable interaction) via partial histograms, also this visualisation approach is introduced in this chapter.

The third chapter is devoted to the description of the 1-D version of the adaptive scheme, developed to have a more efficient sampling of the considered expensive function. The adaptive scheme considers a local adaptive approach and a global one. The local adaptive approach is focused only on some partial functions, while the global approach focuses on the statistical properties of the problem of interest. The combination of these two approaches creates a robust UQ propagation tool for 1-D domains. Moreover, the local adaptive scheme considers a search algorithm, which provides optimal ratio between exploration vs. exploitation for newly proposed samples.

In the fourth chapter, the new scheme for the selection of important domains is introduced. This scheme allows to select important domains with a very small number of samples and therefore, dramatically reduces the necessary number of samples, thus increasing the efficiency of the proposed HDMR based sensitivity and UQ method.

The fifth chapter is focused on the description of the adaptive scheme for higher dimensions. The high dimensional adaptive scheme, the N-N scheme, is again a combination of a local and a global process. In the chapter, the differences existing between the 1-D scheme and N-D scheme, are highlighted and discussed.

Up to chapter five, only Lagrange interpolation is considered and used, but this technique is not robust and cannot handle discontinuities. Therefore, in chapter six a new multi surrogate approach, which tries to select and combine the best interpolation models for the considered problem is described. The multi surrogate approach exploits the characteristics of each surrogate model available and implements a robust combination better suited for the different cases that can be encountered.

In the seventh chapter, a new surrogate technique, which can better handle function discontinuities, is introduced. This newly proposed surrogate technique, resembling the spline technique, is tested on various problems and proves to be a competitive interpolation approach.

The eight chapter is devoted to the description of the multi-fidelity approach. In the beginning, the power ratio approach, which allows comparing different fidelity models in terms of speed and accuracy, is introduced. Then, the application to the cut-HDMR modelling is introduced and the use of the additive correction is showed. Finally, the additive correction is transformed into the multi-fidelity prediction scheme, which predicts the errors between different fidelity models, contributing to keep the multi fidelity scheme efficient.

The ninth and tenth chapters describe two major test cases.

In the ninth chapter, the single fidelity approach is used to treat three cases of space debris re-entry. For the uncertainty propagation, up to 16 random variables are considered, where 10 random variables represent the uncertainty on atmospheric properties and the remaining 6 represent uncertainties on the initial conditions of atmospheric trajectory. Results are validated against MC simulations.

In the tenth chapter, both the single- and the multi-fidelity approaches are used to perform the sensitivity and UQ analyses on the aerodynamic performance of the ESA's Gravity field and steady-state Ocean Circulation Explorer (GOCE) satellite. Two different Direct Simulation Monte Carlo (DSMC) implementations/tuning are considered as low- and high-fidelity models. The results obtained via single- and multi-fidelity approaches are compared and discussed, showing the potentialities of the proposed multi-fidelity approach.

The last chapter gives a final summary of the proposed single- and multi-fidelity approaches. In the discussion, the advantages and disadvantages of the proposed methods are explained and discussed. The final part is given to draw general conclusions and to introduce future developments.

Chapter 2

Derivation of cut-HDMR and new interpolation approach

In this chapter, the focus is given to the development of a new approach for the surrogate modelling and sensitivity analysis. The final surrogate model (approximation model) is constructed as a sum of independent low dimensional surrogate models. This approach proved to be an efficient way for surrogate modelling and also opened new interesting conclusions, which are discussed in this chapter.

The new approach is based on Derivative Equation (DE), which is introduced and derived in this chapter. Also, the statistical properties for DE are developed and new statistical definitions are introduced. DE allows visualizing the influence of each independent variable and combinations of variables. This is showed in Sec. 2.4, where the approach is explained on a toy example.

The chapter is structured in the following way: In the theory section, Derivative Equation (DE) is introduced. In the next section, DE is converted into the numerical approach, which is more suited to a real application and the connection with the well known cut-HDMR approach is showed. Also, the process of the newly proposed surrogate modelling is explained. The last part is given to the discussion of the interpolation process, the newly derived equation and also, the visualization process.

2.1 Literature review

Most of the sensitivity analysis methods handle the complex code as a black box, i.e. they are working with inputs and outputs only. This requires running the complex code many times with various inputs to obtain statistical properties of the given model. The sensitivity analysis can be divided into two main aspects - global and local sensitivity. The global sensitivity tries to estimate the influence over the whole stochastic domain, while the local sensitivity only searches small region of the stochastic space (5). Among the most popular sensitivity analysis methods are the Monte Carlo (MC) based approaches, which represents a branch of global sensitivity analysis. They randomly sample the stochastic domain with a large number of samples and estimate the statistical properties of the given model. Due to its nature, MC methods can cope with non-smooth functions.

The well known Sobol sensitivity method (6, 7) is based on the MC approach. This method decomposes the output variance of the given model into parts attributable to input variables. The outcome of the method is the quantification of the influence of each variable and its interactions in the given model.

Another popular variance based method is the Fourier Amplitude Sensitivity Testing (FAST) (8, 9). The method performs the sensitivity analysis for a smaller number of required samples than the Sobol method. However, FAST is not capable of sensitivity analysis on interaction terms in the given model. Later, the FAST was extended to compute the total effect of a given variable. Despite the fact that FAST performs more efficiently (in terms of required sampling) than the Sobol method, the Sobol method is still the most widely used variance based method due to its easy programming and robustness.

In order to get a better insight into the model and to visualize the influence of a variable, scatter plots (6, 7, 10, 11) can be used. Scatter plots are very good way to estimate the behaviour of a function in a given domain, but they require a lot of experience on sensitivity analysis. In other words, one has to have an experience in the given topic to validate the results obtained with scatter plots.

All the mentioned methods rely on a large number of samples, i.e. function calls. This implies a large computational burden and in many cases, the time required to run sensitivity analysis would be infeasible. For the purpose of reducing the number of samples, the Method of Morris (MoM) (12) can be used. The MoM for global sensitivity analysis is a modification of one-step-at-a-time methods (7). If one is interested in local sensitivity, the partial derivations can be used. Another well used method is the linear regression (6, 7), but this approach requires that the model of interest is linear.

For the visualization of sensitivity, the CobWeb plots (13), also known as web diagrams, or the Variable Interaction Network (VIN) diagrams (14) can be used. The Cobweb plots enable to visualize the combination of the input variables, leading to a specific range of the output variable. The VIN diagrams can be used to track and to visualize additive structures in a function of interest.

A way to overcome the computational time of sensitivity analysis is to build a cheap surrogate model of the expensive function. Among the surrogate approaches, there are methods such as the Stochastic Collocation method (SC) (15, 16), the Polynomial Chaos (PC) (15, 16, 17, 18, 19, 20, 21, 22), the Kriging surrogate model (9, 23, 24) and the Pade-Legendre approximation. These methods are non-intrusive by nature, i.e. they consider the code of interest as a black box. In the work of Sudret (25), the non-intrusive Polynomial Chaos is coupled with the Sobol method and it allows to compute the Sobol Indices directly from the PC expansion. However, this method requires a smooth underlying function.

In the case of other surrogate models, the sensitivity indices are obtained by sampling these cheap models via MC approach. Unfortunately, one of the largest limitations of surrogate models is the so called Curse of Dimensionality (CoD), introduced by Richard Bellman (26). This problem is still an issue and it limits the use of non-intrusive methods to a lower number of stochastic dimensions. Various sampling techniques were proposed to handle the CoD problem. The Latin Hyper-cube sampling (LHS) (9) was successfully used in various problems and some different approaches are available, such as LaPSO (27), Uniform Design (UD) (9) or Hammersley Sampling (19, 28).

In the framework of Uncertainty Quantification (UQ) problems, Smolyak Sparse grid (29, 30, 31) and its various modification became very popular techniques. This sampling strategy combined with Non-intrusive Polynomial Chaos (NIPC) gives very accurate results for a low number of samples. Unfortunately, even this approach is not affordable because of its high cardinality.

The cut-High Dimensional Model Representation (cut-HDMR) (32, 33) was developed to decouple the interaction effects of chemical systems. It was successfully used in other fields such as uncertainty quantification (34, 35), sensitivity analysis (36, 37) and interpolation problems (38, 39, 40) and it proved to be a very efficient tool for high dimensional problems, especially for high dimensional integration.

In this section, all these problems together are addressed. The new way of surrogate modelling is presented and followed by a sensitivity analysis. The approach is based on the cut-HDMR, which is here derived in a different way. A new equation is established, which is derived from ANOVA decomposition. The proposed equation enlighten some important aspects of a high dimensional information propagation and important conclusions are derived here. Following the derivation, the developed equation is transformed to the cut-HDMR model, which was originally derived in the work of Rabitz et al. (32). In the work of Rabitz, the cut-HDMR is directly obtained from the ANOVA decomposition without any further explanation. Like the HDMR approach, the proposed approach allows decomposing the stochastic space into sub-domains, which are then interpolated separately via the selected interpolation technique. Each interpolation technique is built accordingly to the conclusions obtained from a new derivation of the cut-HDMR. The statistical properties of a given function are obtained using the MC simulation on each interpolated sub-domain. The outcome on each interpolated sub-domain can be visualized using histograms. This gives the user a completely new insight into the problem.

2.2 Theory

In order to derive the cut-HDMR, let us first introduce a new equation, the Derivative Equation (DE), which is derived from the ANOVA decomposition (14, 41). Let us consider an integrable function, $f(\mathbf{x})$, which is defined on a n-dimensional unit hypercube - $[0,1]^n$ and $\mathbf{x} \in [0,1]^n$. The ANOVA representation of $f(\mathbf{x})$ can be

$$f(\mathbf{x}) = f_0 + \sum_{s=1}^{n} \sum_{i_1<..<i_s}^{n} f_{i_1 \ldots i_s}(\mathbf{x}) \tag{2.1}$$

where $1 \leq i_1 < i_2 < \ldots < i_s \leq n$ and $1 \leq s \leq n$. The explicit form of Eq. (2.1) is

$$f(\mathbf{x}) = f_0 + \sum_{i=1}^{n} f_i(x_i) + \sum_{1 \leq i < j \leq n} f_{i,j}(x_i, x_j) + \ldots + f_{1,\ldots,n}(x_1, \ldots, x_n) \tag{2.2}$$

where f_0 is the constant term and represents the mean value of $f(\mathbf{x})$, the function $f_i(x_i)$ represents the contribution of variable x_i to function $f(\mathbf{x})$, the function $f_{i,j}(x_i, x_j)$ represents the pair correlated contribution to $f(\mathbf{x})$ by the input variables x_i and x_j, which are defined as $1 \leq i < j \leq n$, etc. The last term $f_{1,...,n}(x_1, ..., x_n)$ contains the correlated contribution of all input variables and the total number of summands for Eq. (2.2) is 2^n.

The ANOVA representation is well known and its properties are well described in various works (36, 37, 40). ANOVA stands for Analysis Of Variances and it represents a particular form of statistical hypothesis testing. Its sums of squares indicate the variance of each component of the decomposition and comparisons of mean squares, along with additional statistical tests, it allows testing of a nested sequence of models. Widely used linear models with coefficient estimates and standard errors are also, closely related to the ANOVA decomposition. Moreover, it is computationally elegant and a variety of experimental designs were adapted for ANOVA.

In order to obtain the separated contributions of the function of interest, it is necessary to obtain the derivative of the function of interest according to its variables. The derivative represents a separate contribution to the function of interest for a given variable. Therefore, consider a function, $f(\mathbf{x})$, which is derivable and integrable. Accordingly, all terms in Eq. (2.2) are integrable and derivable too. Let us derive each term in Eq. (2.2) accordingly to its generic variable x_i and obtain the infinitesimal increment

$$df_i(x_i) = \frac{\partial f(\mathbf{x})}{\partial x_i} dx_i \tag{2.3}$$

which for the two dimensional terms is

$$df_{i,j}(x_i, x_j) = \frac{\partial f(\mathbf{x})}{\partial x_i, x_j} dx_i dx_j \tag{2.4}$$

Higher order terms are derived accordingly. Since the first term on the right side of the Eq. (2.2) is a constant, i.e. $f_0 = C$, hence $df_0 = 0$, then the infinitesimal increment of function $f(\mathbf{x})$ can be written as

$$df(\mathbf{x}) = \sum_{i=1}^{n} \frac{\partial f(\mathbf{x})}{\partial x_i} dx_i + \sum_{1 \leq i < j \leq n} \frac{\partial f(\mathbf{x})}{\partial x_i, x_j} dx_i dx_j + ... + \frac{\partial f(\mathbf{x})}{\partial x_1, ..., x_n} dx_1 ... dx_n \tag{2.5}$$

The Eq. (2.5) is the basic form of DE and it relates the change of the function of interest on the change of input variables. In other words, Eq. (2.5) gives the infinitesimal increment of the function in a particular point. Moreover, the equation describes the relationship between the stochastic spaces. The propagation of information from the lower order stochastic space to the higher order stochastic space depends on given partial derivative of the function of interest. However, the most important aspect is the independence of each derivative. The derivative of a function in a given direction is a separate function and such that can be handled separately. Therefore, multiple surrogate techniques can be used for the problem of interest, i.e. each derivative part in DE can be handled with a different surrogate technique. On the other hand, DE is very hard to use in the practical applications and obtaining derivatives from a function of interest is in many cases a hard task and in some cases practically impossible. Therefore, the integral form is introduced. The integral form is a necessary step to obtain the well known cut-HDMR model. In order to derive the integral form, let us integrate Eq. (2.5) in the same way as derivatives were applied

$$f_i(x_i) = \int \frac{\partial f(\mathbf{x})}{\partial x_i} dx_i \tag{2.6}$$

which for the two dimensional terms is

$$f_{i,j}(x_i, x_j) = \int \int \frac{\partial f(\mathbf{x})}{\partial x_i, x_j} dx_i dx_j \tag{2.7}$$

Higher order terms are derived accordingly. Each term is handled as a separate function and therefore, it is integrated separately.

The left hand side term in Eq. (2.5) is integrated in the following way

$$\int df(\mathbf{x}) = f(\mathbf{x}) + C \tag{2.8}$$

where $\mathbf{x}$ represents a vector of inputs and $C = -f_0$, i.e. it is a constant. However, it is more practical to use definite integrals. The definite integral gives to the equation more physical meaning and represents a finite increment to the quantity of interest. Therefore, let us rewrite Eq. (2.8) in the following form

$$\int_{f(^{\mathbf{c}}\mathbf{x})}^{f(\mathbf{x})} df(\mathbf{x}) = f(\mathbf{x}) - f(^{\mathbf{c}}\mathbf{x}) \tag{2.9}$$

where $f(\mathbf{x})$ represents a function value at point $\mathbf{x}$ and $f(^{\mathbf{c}}\mathbf{x})$ represents a function value at point $^{\mathbf{c}}\mathbf{x}$. Using the Eq. (2.9) and a definite integral for each summand, the integrated form of the Eq. (2.5) reads

$$\begin{gathered} f(\mathbf{x}) - f(^{\mathbf{c}}\mathbf{x}) = \sum_{i=1}^{n} \int_{^{c}x_i}^{x_i} \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_i} d\xi_i + \\ \sum_{1 \leq i < j \leq n} \int_{^{c}x_i}^{x_i} \int_{^{c}x_j}^{x_j} \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_i, \xi_j} d\xi_i d\xi_j + ... + \int_{^{c}x_1}^{x_1} ... \int_{^{c}x_n}^{x_n} \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_1, ..., \xi_n} d\xi_1 ... d\xi_n \end{gathered} \tag{2.10}$$

where ${}^{\mathbf{c}}\mathbf{x}$ represents a central position in the stochastic space, called the central point[1]. In the cut-HDMR model this point is called anchored point. In this work, the central point is considered as the statistical mean value of a given stochastic random variable for all following chapters, i.e. ${}^{c}x_i = mean(x_i)$. If a different position is assumed, it will be explicitly noted in the text.

This equation is similar to the essence of the cut-HDMR approach and to clarify the similarity, the transformation of the derivative equation into the cut-HDMR approach is later introduced in Sec. 2.3. In this work, each integral part of Eq. (2.10) is called increment function. The notation for the increment function is dF_t and the subscript represents the given increment function, which is bounded as follow $1 \leq t \leq 2^n - 1$. The increment function represents a finite increment to function $f(\mathbf{x})$ and its physical meaning is the influence of the given stochastic domain to the function $f(\mathbf{x})$. The number of integrable variables represents the order of the increment function and higher order increment functions, i.e. ≥ 2, represent the influence of the interactions between variables.

In order to obtain the sensitivity analysis used later in many decision criteria, it is necessary to define the mean value and the variance for a given function. Using the integral form of DE, the mean value for Eq. (2.2) can be obtained. Let us remind that the terms in Eq. (2.10) are independent and therefore, each term can be integrated separately. The function (2.10) is integrated into following form

$$\begin{aligned}\mu = \int_{-\infty}^{\infty} f(\mathbf{x})p(\mathbf{x})d\mathbf{x} = f({}^{\mathbf{c}}\mathbf{x}) + \sum_{i=1}^{n} \int_{-\infty}^{\infty} \int_{{}^{c}x_i}^{x_i} \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_i} d\xi_i p_i(x_i) dx_i + \\ \sum_{1 \leq i < j \leq n} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{{}^{c}x_i}^{x_i} \int_{{}^{c}x_j}^{x_j} \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_i, \xi_j} d\xi_i d\xi_j p_{ij}(x_i, x_j) dx_i dx_j + ... + \\ \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} \int_{{}^{c}x_1}^{x_1} ... \int_{{}^{c}x_n}^{x_n} \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_1, ..., \xi_n} d\xi_1 ... d\xi_n p_{1...n}(x_1, ..., x_n) dx_1 ... dx_n\end{aligned} \tag{2.11}$$

where $p_i(x_i)$ is the Probability Density Function (PDF) for the given distribution. The probability of each variable is handled separately too. This is a direct consequence of the independence of terms in Eq. (2.10). From Eq. (2.11), the partial expected value, μ_i, can be defined. The partial expected value for the first order terms is written in the following way

$$\mu_i = \int_{-\infty}^{\infty} \int_{{}^{c}x_i}^{x_i} \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_i} d\xi_i p_i(x_i) dx_i \tag{2.12}$$

[1] In this thesis, an upper-script before a letter represents a specific position in the stochastic space, e.g. ^{c} represents the central point

and for the second order terms the function reads

$$\mu_{ij} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\int_{^{c}x_i}^{x_i}\int_{^{c}x_j}^{x_j} \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_i, \xi_j} d\xi_i d\xi_j p_{ij}(x_i, x_j) dx_i dx_j \tag{2.13}$$

where $p_{ij}(x_i, x_j)$ is the joint PDF. If random variables are independent from each other, then the PDF is given by

$$p_{ij}(x_i, x_j) = p_i(x_i)p_j(x_j) \tag{2.14}$$

Higher order partial expected values are defined accordingly. Unfortunately, the same approach cannot be applied to the higher order moment-generating functions. The explanation is given in Appendix A. On the other hand, the partial variance can still be defined, however, it cannot be summed as the expected value. The partial variance represents a variance of the given increment function and it is a very good estimation of sensitivity for given increment function. The first order partial variance, $\sigma_i^2(x_i)$, reads

$$\sigma_i^2 = \int_{-\infty}^{\infty} \left(\int_{^{c}x_i}^{x_i} \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_i} d\xi_i - \mu_i \right)^2 p_i(x_i) dx_i \tag{2.15}$$

and for the second order terms the function reads

$$\sigma_{ij}^2 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(\int_{^{c}x_i}^{x_i}\int_{^{c}x_j}^{x_j} \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_i, \xi_j} d\xi_i d\xi_j - \mu_{ij} \right)^2 p_{ij}(x_i, x_j) dx_i dx_j \tag{2.16}$$

Higher order partial variances are defined accordingly. The work of Sobol (41) is followed and sensitivity indices are defined in the following way

$$S_t = \frac{\sigma_t^2}{\sigma^2} \tag{2.17}$$

where t represents the selected increment function, i.e. the partial variance function, which is bounded as follow $1 \leq k \leq 2^n - 1$ and σ^2 represents the total variance defined as

$$\sigma^2 = \int_{-\infty}^{\infty} (f(\mathbf{x}) - \mu)^2 p(\mathbf{x}) d\mathbf{x} = \int_{-\infty}^{\infty} ... \int_{-\infty}^{\infty} ((f(^{c}\mathbf{x}) + \sum_{i=1}^{n} \int_{^{c}x_i}^{x_i} \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_i} d\xi_i + \\ \sum_{1 \leq i < j \leq n} \int_{^{c}x_i}^{x_i}\int_{^{c}x_j}^{x_j} \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_i, \xi_j} d\xi_i d\xi_j + ... + \int_{^{c}x_1}^{x_1} ... \int_{^{c}x_n}^{x_n} \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_1, ..., \xi_n} d\xi_1 ... d\xi_n) - \mu)^2 p(\mathbf{x}) dx_1 ... dx_n \tag{2.18}$$

Note that

$$\sigma^2 \neq \sum_{t=1}^{2^n - 1} \sigma_t^2 \tag{2.19}$$

Sensitivity indices are all non-negative and by using them, the functional structure and the rank of variables for given functions can be estimated. In most non-linear functions, the theoretical approach is hard to apply and therefore, the derivative equation is transformed to the numerical approach leading to the well known cut-HDMR model.

Now, let us compare DE to the well known ANOVA decomposition. The basic concept behind HDMR is that many physical systems do not exhibit high-order cooperative behavior as it is proved by statistical evidence (see (42)). On one hand, the ANOVA decomposition was developed in order to quantify the influence of each variable or interaction of variables, but it does not show why the high-dimensional inputs in the various cases have a null influence on the output. In other words, the ANOVA decomposition does not provide an explanation why higher order interactions have a smaller effect than the lower ones (see (41)). On the other hand, DE explains why this phenomenon is happening, i.e. in most of the cases, the high order partial derivatives are smaller than the low order partial derivatives. Moreover, DE shows why samples in the particular space have an influence on the final output, while other samples not. In other words, one can understand the structure of a given function in the stochastic space and influence of the lower stochastic domain on the higher ones. This cannot be deduced from the HDMR approach.

Moreover, if we look at the term $df(\mathbf{x})$ alone, that part can be understood as a change of a function along the stochastic space. One can intuitively think, that $df(\mathbf{x})$ is basically infinitesimal change around the central point of a given function, i.e. the size of uncertainty around given point. Therefore, let $\mathbf{x}$ be a set of random variables with assigned distributions, then $df(\mathbf{x})$ represents an infinitesimal change in the final Probability Density Function (PDF) given by Eq. (2.5). If we focus on the right hand side of the Eq. (2.5), it can be seen that this change is given only by its derivatives. Therefore, we can propose following:

Proposition 2.2.1 *Let $f(\mathbf{x})$ be a function with range $D \in [-\infty\ \infty]^n$ and its domain $\mathbf{x} \in \Re^n$. Let $\mathbf{x}$ be a set of random variables with given PDF, then the shape of resulting PDF is given only by input distributions and derivatives of the given function, $f(\mathbf{x})$.*

The above postulate is proved using MC simulation in Appendix B. However, one can understand from the postulate that the most simple model, which propagates uncertainty is a linear model, i.e. hyper-dimensional surface.

The integral form of DE is a variation of the cut-HDMR approach (introduced later in this chapter - see Sec. 2.3) and represents the finite increment to the function of interest. On the other hand, the DE itself represents the infinitesimal increment to the function of interest and this is the main difference between the integral form of DE and DE itself. Moreover, the standard cut-HDMR representation (see (32) or Sec. 2.3) is written in a set of analytic equations and does not allow to deduce important conclusions discussed later in this section. The integral form of DE shows in a clear way properties of the increment functions such as the zero-th value of the increment function if one of the integral limits is equal to the central point or the connection between derivatives and increment functions.

As discussed earlier, there can be obtained several conclusions from the integral form of DE and its integral variation. The first very important observation can be made from Eq. (2.13) and Eq. (2.16). It can be seen that the magnitude of an integral part, e.g.

$$dF_{i.j}(x_i, x_j) = \int_{c_{x_i}}^{x_i} \int_{c_{x_j}}^{x_j} \frac{\partial f(\boldsymbol{\xi})}{\partial \xi_i, \xi_j} d\xi_i d\xi_j$$

increases with the distance from the central point. At the same time, if the Gaussian distribution is assumed, the probability of occurrence, $p_{ij...k}$, decreases, i.e. the most of the samples are distributed around the central point, where interaction effects are negligible. Therefore, it can be concluded that tails of the output distribution are mainly given by the higher order partial derivatives, i.e. if the output distribution has heavy tails and the input distributions are light tailed (most cases in real life), the interaction terms are very strong in the function of interest (see Chap. 5). This observation can be extended to other probability distributions if one realizes that the higher order increment functions work only if the lower order increment functions works, e.g. in order to be the second order increment function non-zero ($dF_{1.2}(x_1, x_2) \neq 0$), all the first order increment functions included must be non-zero also ($dF_1(x_1) \neq 0$ and $dF_2(x_2) \neq 0$). One can understand that the lower order increment function has a higher probability of being non-zero than the corresponding higher order increment function, i.e.

$$P(dF_i(x_i) \neq 0) > P(dF_{i.j}(x_i, x_j) \neq 0)$$

Taking into account the structure of DE, i.e. its additive form, one can quickly realize that tails of the posterior distribution are given by the higher order interactions as tails of the posterior distribution represents the maximum distance from the expected value. Therefore, we can propose following:

Proposition 2.2.2 *Let $f(\mathbf{x})$ be a function with range $D \in [-\infty\ \infty]^n$ and its domain $\mathbf{x} \in \Re^n$. Let y be the probability density function of an output of function $f(\mathbf{x})$, then the tails of the distribution y are heavier, the stronger interactions are in function $f(\mathbf{x})$.*

In a practical sense, one can expect strong interactions (in the sense of HDMR), if the observed output distribution has heavy tails. This is very important for scientific disciplines such as reliability studies, where the main focus is given on the tails of output distributions. Therefore, one interested into the tails of given distribution must always take into account higher order increment functions. Also, the integral form of DE and proposed definition explain why the high dimensional normal distribution has heavy tails. In other words, the normal distribution is defined in the following way:

$$PDF(x_1, ...x_n) = \frac{1}{\sqrt{(2\pi)^n |\mathbf{C}|}} e^{-\frac{1}{2}(\mathbf{x}-\mu_{PDF})^T \mathbf{C}(\mathbf{x}-\mu_{PDF})} \tag{2.20}$$

where $\mathbf{C}$ is the semi positive covariance matrix and μ_{PDF} represents the mean value of given distributions. Application of DE to Eq. (2.20), one can easily see that all the higher partial derivatives are non-zero. Therefore, one can conclude that Eq. (2.20) have large tails. The proposed definition is confirmed in practical examples given later in this thesis.

The second important conclusion is the stochastic domain dependence. This aspect was already mentioned in Eq. (2.5) and it can be written as follows

$$if$$

$$\frac{\partial f(\mathbf{x})}{\partial x_i} = 0 \qquad \forall \mathbf{x} \in R^n$$

$$then$$

$$\frac{\partial f(\mathbf{x})}{\partial x_j ... \partial x_k \partial x_i} = 0 \qquad \forall \mathbf{x} \in R^n$$

This statement can be extended to the increment functions in the following way

$$if$$

$$dF_i(\mathbf{x}) = 0 \qquad \forall \mathbf{x} \in R^n$$

$$then$$

$$dF_{ij\ldots k}(\mathbf{x}) = 0 \qquad \forall x \in R^n$$

where dF_i represents the increment function. The above assumption comes naturally from a basic integral calculus as an integration of 0 is always 0, i.e. integration of zero-th derivative is always 0. Moreover, the increment function will be either increasing, decreasing or null, it will never be a constant in the given domain. The second conclusion can be extended to the following statement. Each high order partial derivative contains information about the low order partial derivatives and vice versa, i.e. $\partial f(\mathbf{x})/\partial x_1 \partial x_2$ contains information about $\partial f(\mathbf{x})/\partial x_1$ and about $\partial f(\mathbf{x})/\partial x_2$. Therefore, it can be concluded: for the first case

$$if$$

$$\frac{\partial f(\mathbf{x})}{\partial x_i} \geq \frac{\partial f(\mathbf{x})}{\partial x_i \partial x_j} \qquad \forall \mathbf{x} \in R^n$$

$$then$$

$$dF_i(\mathbf{x}) \geq dF_{ij}(\mathbf{x}) \qquad \forall \mathbf{x} \in R^n$$

and for the second case

$$if$$

$$\frac{\partial f(\mathbf{x})}{\partial x_i} < \frac{\partial f(\mathbf{x})}{\partial x_i \partial x_j} \qquad \forall \mathbf{x} \in \left(R^n, \frac{\partial f(\mathbf{x})}{\partial x_i \partial x_j} \neq 0 \right)$$

$$then$$

$$dF_i(\mathbf{x}) < dF_{ij}(\mathbf{x}) \qquad \forall \mathbf{x} \in \left(R^n, \frac{\partial f(\mathbf{x})}{\partial x_i \partial x_j} \neq 0 \right)$$

However, the second case was encountered only for rare problems and the first case is valid for the majority of problems. From the above conditions and for the first case, if the low order increment function converges [1] using a certain number of samples, the higher order increment functions converge under less or same number of samples for a given domain. This will lead to a simplification of the sampling space and dramatic reduction of necessary samples. Moreover, the sensitivity analysis has to hold the same condition. The same conclusion was experimentally confirmed in (36).

[1] Convergent in the sense that the measured residual for the surrogate model used in given increment function is below given threshold.

One very important conclusion can be done with a connection to the Polynomial models and the Smolyak Sparse grid approach. Smolyak noticed that the same accuracy can be achieved using fewer samples if the higher domains are sampled with a lower number of samples, i.e. use a lower order quadrature. He developed the famous Smolyak Sparse grid approach for the numerical integration, where higher order domains are sampled with a lower number of samples. With a combination of the Generalized Polynomial Chaos, this created a powerful tool to propagate uncertainty in high dimensional spaces. However, it was never explained (according to author's knowledge) why this approach works and more importantly where this approach does not work (31). Let us explain why the polynomial models combined with the Smolyak Sparse grid work on an applied example and using DE. Let us assume the following function, $f(x_1, x_2) = x_1^5 + x_2^5 + x_1 x_2$ and its following derivatives:

$$\frac{\partial f(x_1, x_2)}{\partial x_1} = 5x_1^4 + x_2$$
$$\frac{\partial f(x_1, x_2)}{\partial x_2} = 5x_2^4 + x_1$$
$$\frac{\partial f(x_1, x_2)}{\partial x_1 \partial x_1} = 1$$

In this particular example, the lower order derivatives are more complicated than the higher ones, i.e. case 1. Because Eq. (2.5) holds and each term is independent, it can be easily seen that to interpolate the second order derivative is much easier than to interpolate the first order derivative. In other words, to create an interpolation technique for the first order derivatives, one needs more samples than for the second order derivative. Contrary, if the higher order derivatives are more complicated than the lower ones, the polynomial chaos combined with the Smolyak Sparse grid fails. Therefore, observing derivatives gives us insight, when the Sparse Grid approach works and more importantly when it will not work and another approach has to be used. If the higher order derivatives are more complicated than the lower one, naturally one will need more samples in given domain then in the lower domains. For example, Arrhenius equation is used for chemical reactions. The Arrhenius equation is not very well derivable and therefore, it will need more samples in higher domains. This is well documented in work (43). Unfortunately, the chemical flows are more complicated and a presence of discontinuities in the function is a common phenomenon. This issue will be addressed later in this work.

The last conclusion can be deduced from the derivatives in Eq. (2.5). Each derivative represents a separable problem and such that allows using independent surrogate techniques. Therefore, one can intuitively think that each of these surrogate techniques has a local maximum accuracy, which can be acquired by given surrogate technique. Therefore, if only a certain order of DE is selected, i.e. not all increment functions are included in the final model, the maximum achievable accuracy is given by the influence of the neglected increment functions. In other words, adding higher orders of increment function will allow higher accuracy. This is well shown in work of Zhang et al. (40).

2.3 Numerical approach - cut-HDMR model derivation

The integral form of DE can be applied to real problems, however, the derivatives and integrals are not practical to compute. It is more convenient to transform the integral form into a set of equations, which are easy to solve, i.e. transform the integral form to the analytic equation. The numerical approach is based on Eq. (2.10) and the well know First Fundamental Theorem of Calculus (44). The theorem can be written in the following way

$$f(b) - f(a) = \int_a^b B(x)dx \tag{2.21}$$

where $B(x)$ is a continuous function on the closed interval $[a, b]$. Using the Second Fundamental theorem of Calculus (44) and defining $B(x)$ in the following way

$$B(x) = \frac{\partial f(x)}{\partial x}$$

Eq. (2.21) can be rewritten such that

$$f(b) - f(a) = \int_a^b \frac{\partial f(x)}{\partial x} dx \tag{2.22}$$

This simple formula can be applied to the integral form of DE (Eq. (2.10)) for all the first order derivatives. This step allows to cast off the derivative and the integral and replace them with a simple equation.

To closely explain the numerical application, let us consider a continuous function $f(x_1, x_2, x_3)$. The first term in the integral form of DE using Eq. (2.22) reads

$$dF_1(x_1) = \int_{^c x_1}^{x_1} \frac{\partial f(\xi_1,^c x_2,^c x_3)}{\partial \xi_1} d\xi_1 = f(x_1,^c x_2,^c x_3) - f(^c x_1,^c x_2,^c x_3) \tag{2.23}$$

where $^c x_i$ represents the position of the central point of the random variable x_i. The increment function, $dF_1(x)$, contains only one random variable and all other random variables are held constant at their central value. Increment functions for other variables (x_2, x_3) are created in a similar way.

Second order increment functions are a bit more complex to handle. Let us consider the previous function and assume a point on a plane, e.g. $x_3 = {}^c x_3$. If we assume that all integration parts of the integral DE, which are involving x_3, are zero and this assumption allows to rewrite the integral form of DE as follows

$$\begin{aligned} f(x_1, x_2,^c x_3) - f(^c x_1,^c x_2,^c x_3) = \\ \int_{^c x_1}^{x_1} \int_{^c x_2}^{x_2} \frac{\partial f(\xi_1, x_2,^c x_3)}{\partial \xi_1 \partial \xi_2} d\xi_1 d\xi_2 \\ + \int_{^c x_1}^{x_1} \frac{\partial f(\xi_1,^c x_2,^c x_3)}{\partial \xi_1} d\xi_1 + \int_{^c x_2}^{x_2} \frac{\partial f(^c x_1, \xi_2,^c x_3)}{\partial \xi_2} d\xi_2 \end{aligned} \tag{2.24}$$

Clearly the one dimensional integrals of Eq. (2.24) can be replaced by Eq. (2.23) and a similar equation for x_2. This leads to the following simplification

$$\begin{aligned} f(x_1, x_2, {}^c x_3) - f({}^c x_1, {}^c x_2, {}^c x_3) = \int_{^c x_1}^{x_1} \int_{^c x_2}^{x_2} \frac{\partial f(\xi_1, \xi_2, {}^c x_3)}{\partial \xi_1 \partial \xi_2} d\xi_1 d\xi_2 \\ + f(x_1, {}^c x_2, {}^c x_3) - f({}^c x_1, {}^c x_2, {}^c x_3) + f({}^c x_1, x_2, {}^c x_3) - f({}^c x_1, {}^c x_2, {}^c x_3) \end{aligned} \tag{2.25}$$

which can be rewritten into the following form[1]

$$\begin{aligned} dF_{12}(x_1, x_2) = \int_{^c x_1}^{x_1} \int_{^c x_2}^{x_2} \frac{\partial f(\xi_1, \xi_2, {}^c x_3)}{\partial \xi_1 \partial \xi_2} d\xi_1 d\xi_2 = \\ f(x_1, x_2, {}^c x_3) - f({}^c x_1, {}^c x_2, {}^c x_3) - f(x_1, {}^c x_2, {}^c x_3) \\ + f({}^c x_1, {}^c x_2, {}^c x_3) - f({}^c x_1, x_2, {}^c x_3) + f({}^c x_1, {}^c x_2, {}^c x_3) \end{aligned} \tag{2.26}$$

where the increment function, $dF_{12}(x_1, x_2)$, is treated as a function with only two variables and the rest is held constant. These steps allow to cast off the double integral and replace it with a simple equation. The same approach can be

[1] The equation can be simplified; however, in order to show the process, the equation is kept in its original form.

applied to remaining parts of the integral form of DE. Eq. (2.23) and Eq. (2.26) are the same as in the cut-HDMR approach, see (32, 35, 45). From Eq. (2.23) and Eq. (2.26) the following conclusion can be made:

$$\frac{\partial dF_{i...k}(\mathbf{x})}{\partial x_i ... \partial x_k} = \frac{\partial f(\mathbf{x})}{\partial x_i ... \partial x_k} \tag{2.27}$$

The final step is to put all increment functions into the basic shape of the integral form of DE, which reads

$$\begin{aligned} f(x_1, x_2, x_3) - f({}^c x_1, {}^c x_2, {}^c x_3) = dF_1(x_1) + dF_2(x_2) + dF_3(x_3) + dF_{12}(x_1, x_2) + \\ dF_{13}(x_1, x_3) + dF_{23}(x_2, x_3) + dF_{123}(x_1, x_2, x_3) \end{aligned} \tag{2.28}$$

For each increment function, an independent surrogate technique is used and Eq. (2.28) represents a sum of independent interpolation techniques. More importantly, the increment function can be sampled independently, which is direct consequence of orthogonality of increment functions. Various techniques as a surrogate model for each increment function can be used. For example a Kriging surrogate model or Stochastic Collocation approach. Note that higher increment functions in Eq. (2.28) can be zero and such that they can be easily neglected.

To compute the partial expected value and the partial variance, the established theoretical partial mean and variance are not suitable. Therefore, a numerical estimation of partial mean and variance follows

$$\mu_t = \frac{1}{z} \sum_{j=1}^{z} dF_t(\mathbf{x}_j) \tag{2.29}$$

$$\sigma_t^2 = \frac{1}{z-1} \sum_{j=1}^{z} (dF_t(\mathbf{x}_j) - \mu_t)^2 \tag{2.30}$$

where t represents a selected increment function and z represents the number of samples of MC simulation applied on the surrogate model of the increment function, $dF_t(x)$. The expected value is established as a sum of partial expected values and reads

$$\mu = f(^{\mathbf{c}}\mathbf{x}) + \sum_{t=1}^{2^n - 1} \mu_t \tag{2.31}$$

where n represents a number of random variables. The total variance is defined as follows

$$\sigma^2 = \frac{1}{z-1}\sum_{j=1}^{z}\left(\sum_{t=1}^{2^n-1} dF_t(\mathbf{x}_j) - \mu\right)^2 \tag{2.32}$$

where, again, n represents the number of random variables. One fundamental problem arises from the estimation of the total variance. The total variance requires all increment functions to be included and the number of samples required grows exponentially with increasing number of random variables. Practically, higher order increment functions are neglected if they have zero or very low influence on the final result and the functions (2.31,2.32) become

$$\mu \approx f(^c x) + \sum_{t=1}^{s_D} \mu_t \tag{2.33}$$

$$\sigma^2 \approx \frac{1}{z-1}\sum_{j=1}^{z}\left(\sum_{t=1}^{s_D} dF_t(\mathbf{x}_j) - \mu\right)^2 \tag{2.34}$$

where s_D represents a selected number of increment functions, which is limited to $s_D \leq 2^n - 1$. The sensitivity indices are obtained via Eq. (2.17).

2.3.1 The interpolation process

In this work, the increment functions are considered a separable problem. Therefore, the surrogate model building process is done in a new and more efficient way, where each increment function is handled as a separate problem and the final model comes from the sum of various interpolation techniques. This allows combining various interpolation techniques to overcome any interpolation problem. The selected interpolation technique is applied in the following way. Let us consider the general increment function $dF_t(x_i, ..., x_j)$, the interpolation model of the increment function is created using only samples from the given stochastic domain and the rest is held constant, i.e. only samples $[^c x_1, ..., ^c x_{i-1}, x_i, ..., x_j, ^c x_{j+1}, ..., ^c x_n]$ are considered. Other samples are completely neglected in the process of interpolation, i.e. they have a null influence on the process. The interpolation model is created and stored. The statistical properties of the given increment function are obtained using MC sampling applied directly to this surrogate model. The final model is created as a sum of these models (Eq. (2.28)) and overall statistical properties are obtained directly from the final model. In Fig. 2.2 is showed a diagram how the interpolation process is handled.

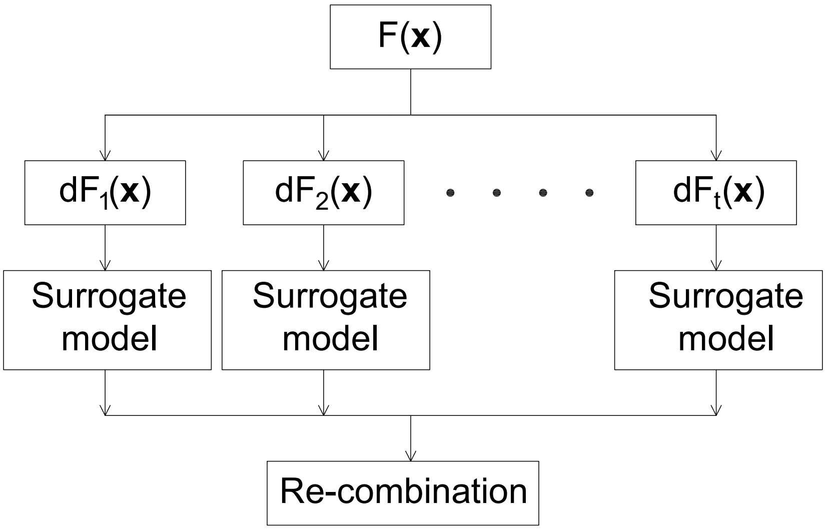


Figure 2.1: Flow chart of the main approach

2.4 Visualization of higher order sensitivity and sensitivity estimation

The process of sensitivity visualization is easy and straightforward. At first, let us recall the first important property of DE, the independence of the increment function. This allows to visualize each increment function separately and observe its influence on the model. For visualization purposes, histograms are used.

The first order increment function represents the sensitivity of one particular variable as it can be seen in Eq. (2.23). Using MC sampling, the statistical properties of this particular variable can be observed and its influence can be estimated. From the obtained samples, the partial mean value and the partial variance can be estimated. The partial mean value represents the influence of the input distribution on the final mean value of the output distribution and the partial variance represents the variability for the given distribution.

Let us now assume a higher order increment function. In this case, if one of the variables is equal to the central point, the increment function is zero, e.g. the increment function dF_{12} have zero-th value at abscissas x_1 and x_2. Therefore, this function represents a pure interaction effect, i.e. how a combination of variables influences the final model. In the engineering world, this represents, for example, chemical reactions in hypersonic flows.

In order to explain the above description on an example, let us consider the following function

$$F(x_1, x_2, x_3) = x_1^3 + 0.1x_2^2 - e^{x_1 x_3} \tag{2.35}$$

where x_i is a random variable with an uniform distribution and defined on interval [0, 2]. On this function, the properties of DE can be easily demonstrated and the visualization of high order increment functions can be showed. Histogram of each increment function is showed in Fig. 2.3 and the histogram of the function F (Eq. 2.35) is showed in Fig. 2.2. From these histograms, it can be observed the influence of each random variable on the output. It can be seen that the increment function, dF_1, is moving the mean value to higher values and slightly influencing the variance. The second increment function, dF_2, is influencing the mean value and the variance in a negligible way. However, it can be easily understood from the histogram that the function is monotonic in a given range. From the histogram of the increment function, dF_3, it can be observed that the function is monotonic and with slight influence on the final PDF. However, if variable x_3 is coupled with variable x_1, the increment function, dF_{13}, plays the major role

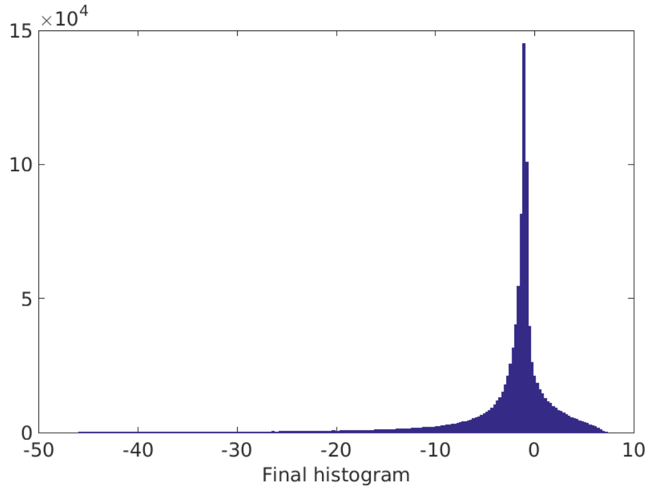


Figure 2.2: Visualization of the final histogram

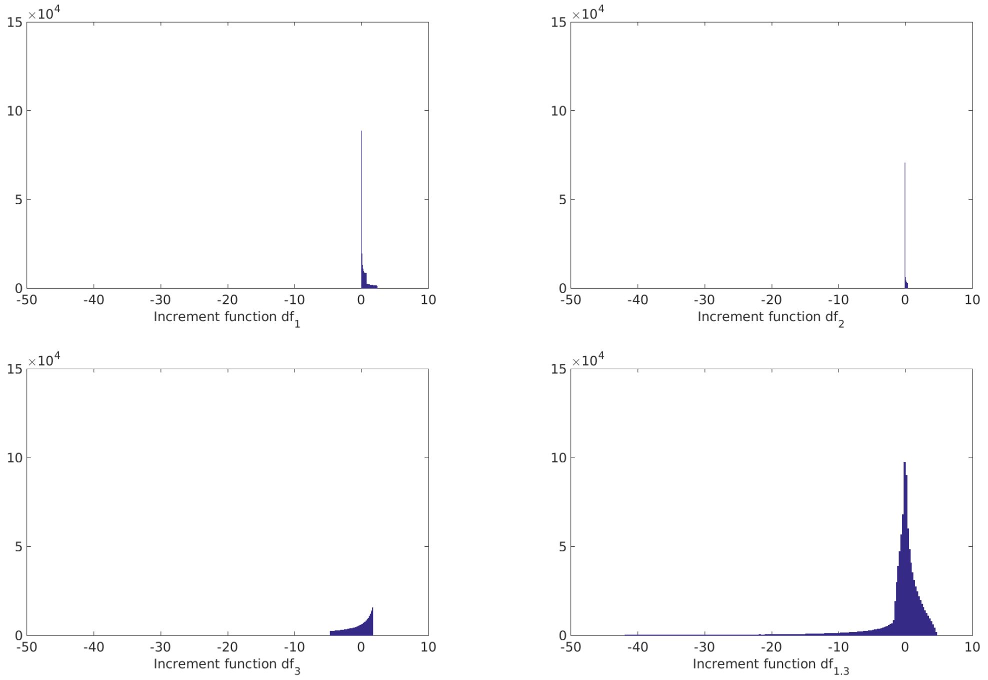


Figure 2.3: Visualization of the partial histograms for each increment function

on the shape of the final PDF and it is responsible for tails of the final distribution. In physical meaning, the increment function, dF_{13} represents an interaction between inputs (x_1, x_3) such as, previously mentioned, chemical reactions. Note that all the other increment functions in this example are zero and therefore, they are not showed. Practical approach how to reconstruct a shape of an increment function from the partial histogram is showed in Appendix C.

The integral form of DE equation allows visualizing the behaviour of a function in the stochastic domain. Note that using DE is different than just simple sampling. The derivative process separates variables of interest from the non-interesting variables and following integration reshapes the function of interest, i.e. creates a function of increment.

2.5 Discussion about Sensitivity analysis, interpolation process, and Derivative Equation

The possibility to combine various surrogate models brings a new way to interpolate high dimensional problems. Followed by the sensitivity analysis, the approach brings a completely new insight into high dimensional uncertainty.

The independence of the increment functions allows establishing the statistical properties of each increment function separately. This is very useful to estimate the PDF of interaction terms, which is necessary for the prediction and neglection of the higher order increment functions used later in this thesis. Moreover, the influence of input probability can be established by application of MC sampling with different input distributions to the created increment function model. However, the probabilistic approach to be valid, the central point has to be selected as a mean of given random variables.

In this work, the interpolation process is slightly more complex to understand. Each increment function is handled separately, i.e. for each increment function is used a different surrogate model (see Fig. 2.2). This is later used in this thesis and proved to be an efficient way to make the method robust and efficient. Moreover, it is easier and more accurate to interpolate each problem separately rather than trying to interpolate altogether. However, at the same time, each interpolation technique represents a new challenge and errors in one interpolation technique will propagate to the whole model. How to handle propagation of errors from increment functions to the whole model is later explained in this thesis.

2.6 Conclusion

In this chapter, the cut-HDMR model is derived in a different way via the derivative equation, which is also defined here. In the derivation process of the cut-HDMR model, important conclusions are made. The derivative equation shows how the information in the stochastic space propagates from the lower order stochastic space to the higher order stochastic space in an understandable way.

The interesting conclusion is made about the tails of the posterior PDF. It is showed that the tails of the output distribution are mainly given by the high order interaction terms. This is confirmed in the applied examples showed in later chapters.

The new derivation of the cut-HDMR approach allowed different approach for the approximation / interpolation process. The stochastic space is decomposed into smaller spaces and these smaller domains are then interpolated with an independent surrogate model. The final model is constructed as a sum of these partial surrogate models. Each of these partial surrogate models is constructed with a different number of samples, which makes the overall process very effective, especially in high dimensional spaces.

The cut-HDMR approach allowed a new type of sensitivity analysis and result presentation. The sensitivity analysis is based on the Sobol approach and it is extended visualizing the influence via the partial histograms. The visualization is shown on a simple example and helps the user to get a deeper insight into the influence of the stochastic variable.

Chapter 3

Adaptive algorithm - The first order increment functions

In this chapter, an adaptive scheme for the sampling procedure is introduced. However, the focus is only on the first order increment functions. Extension to the higher order increment functions is not straightforward and therefore, it is introduced in the following chapters.

First, the theory is established for the adaptive sampling. The adaptive sampling takes into account shape of the underlying problem and also the input probability distribution. This leads to an optimal distribution of samples along given abscissa. In the next section, the numerical application is introduced. The convergence process is discussed in Sec. 3.4, where the local and global convergence process is described. The adaptive scheme is tested on examples, where the first example, is a simple 1-D problem. The second example is a 9-D real life problem, where the global process is showed and results are compared with the Monte Carlo (MC) simulations. Last sections are given to discussion about the adaptive process and to the conclusion.

3.1 Literature review

In the most cases, the non-intrusive approach is relying on an approximation of the expensive code. The approximation of interest can be a quantity of interest, such as the mean and the standard deviation, or the behavior of an expensive code in a domain of interest. In this respect, two main ways are currently discussed in the literature: the first way relies on the numerical integration theory, while the second approach builds a cheap emulator of the expensive code. The numerical integration approach is easy to implement and it is a natural way if the standard probability space (Lebesgue space) is used. The integration techniques such as Non-Intrusive Polynomial Chaos (NIPC) (15, 16, 17, 18, 19, 20, 21, 22) or stochastic collocation (SC) (15, 16) received a large interest in last years and these techniques proved to be reliable over wide range of engineering problems such as flow over airfoil (46), single phase rectifier (18) or acoustics of horn (4). In the last example, the NIPC is performed as a multi fidelity uncertainty quantification. However, the integration methods rely on the accurate estimation of integration quantity, i.e. mean or standard deviation, while other characteristics of the underlying function do not have to be produced in a satisfactory quality, i.e. tails of the output distribution. Nevertheless, even this approach can be considered as an approximation technique.

The second approach builds a cheap emulator of an expensive code, which behaves similarly to the expensive code. The statistical properties are obtained by the Monte Carlo sampling directly on the emulator while taking advantage of a low computational time of the emulator. This approach is useful for applications, where the whole PDF is needed and it is mainly represented by methods such as Polynomial models (47), Kriging Model (9, 23, 24), the Pade-Legendre approximation (48) and Radial Basis functions (9). However, the surrogate models have certain drawbacks. One of the problems is the application of two methods, i.e. the application of a surrogate model followed by the application of MC sampling. Each method represents a source of error, which needs to be taken into account. Currently, the main focus is given only to minimization of the error of the interpolation, i.e. surrogate model, while MC error is assumed to be negligibly small due to a large number of samples.

Both approaches require a sampling strategy, where different sampling strategy can significantly change the accuracy of the interpolation. All these aspects make an interpolation process, using the surrogate model, very challenging and many interesting papers (49, 50, 51) and books (9, 52) have been written about this topic.

The common problem for both methods, either for an integration theory or a surrogate modelling, is the Curse of Dimensionality, introduced by Richard Bellman (26). This aspect limits the use of interpolation techniques to a low number of stochastic dimensions (¡5). In many previous works, sampling techniques were proposed to handle the CoD problem. The Latin Hypercube sampling (LHS) (9) was successfully used in various problems and some different approaches are available, such as LaPSO (27), Uniform Design (UD) (9) or Hammersley Sampling (19, 28). These sampling strategies coupled with a surrogate model proved to be an efficient tool for surrogate modeling. In the framework of Uncertainty Quantification (UQ) problems, Smolyak Sparse grid (30, 31) and its various modifications became very popular. The Smolyak sparse grid sampling strategy combined with the Non-intrusive Polynomial Chaos (NIPC) gives very accurate results for a low number of samples.

Another aspect is the adaptive process of a numerical integration. The adaptive Smolyak sparse grid was mainly used to handle non-smooth functions in various applications (35, 53, 54) and it proved to be a reliable uncertainty quantification tool. In the integration theory, the focus is also given to the adaptive selection of a proper basis for an integrating polynomial (55, 56). This leads to an adaptive sampling process and it helps to reduce the number of samples. On the other hand, the adaptive techniques for a surrogate approach focus mainly on a sampling process over the stochastic domain. The basic sequential sampling is based on a random sampling, which maximize coverage of the stochastic space, while the convergence of the surrogate model is observed (9, 51, 57, 58). Other methods (59, 60, 61) take into account progress of the surrogate model and take into account the behaviour of the underlying function. Another interesting approach is Lola-Voronoi (62), which tries to use gradients in sequential sampling. Very popular sequential sampling is connected with a Kriging surrogate model (63, 64), where connection of exploration and exploitation is given. All these methods obtained a large interest in the recent development of powerful computers.

3.2 Basic theory of the 1-D adaptive process

For polynomial models used in the uncertainty quantification, the adaptive mechanism is based on the nested property of the Clenshaw-Curtis nodes (65) or Adaptive Newton-Cotes sampling based on a tree methodology (35). The Clenshaw-Curtis nodes are positioned according to Clenshaw-Curtis equation and new samples, in each iteration, are added according to this equation. Adaptive Newton-Cotes approach position new sample in the neighborhood of samples with the highest error (66), where the error is obtained from a previous iteration. Other works, which focus on the adaptive sampling for other surrogate techniques, such as Kriging or radial basis function, can be based on various aspects. Sequential sampling (60) position the new sample in an area of the highest error provided by the surrogate model. Space filling design (67) position the new sample with an aim to maximize the coverage area. Another method (59) focus on the use of Laplacian in search of a position for new samples. There are multiple approaches for surrogate models, however, it is not in a scope of this work to describe them all.

In this work, the approach is handled as a surrogate modelling problem and the algorithm directly finds position of a new sample. The propagation of uncertainty is then done using the Monte Carlo sampling on a given surrogate model. In order to introduce the basics of the algorithm, consider an integrable function, $f(x)$, which is defined on a n-dimensional unit hypercube - $[0,1]^n$ and $x \in [0,1]^n$. Decompose the function, $f(x)$, using the cut-HDMR approach. The focus is aimed on the first order increment function, dF_i. The equation for the first order increment function reads

$$dF_i(x_i) = \int_{{}^c x_i}^{x_i} \frac{\partial f(\xi_i, {}^{\mathbf{c}}\mathbf{x})}{\partial \xi_i} d\xi_i = f(x_i, {}^{\mathbf{c}}\mathbf{x}) - f({}^{\mathbf{c}}\mathbf{x}) \tag{3.1}$$

where x_i represents the stochastic variable of interest, ${}^c x_i$ represents i-th component of the central point and $f(x_i, {}^{\mathbf{c}}\mathbf{x})$ represents a function of interest with all variables hold constant except x_i. Note that the leading superscript represents a position in the stochastic space. The central point is considered the statistical mean value of a given stochastic random variable, i.e. ${}^c x_i = mean(x_i)$. Let $SdF_i^{k-1}(x_i)$ be the surrogate model of the increment function, $dF_i(x_i)$, using n_{SdF} samples in iteration $k-1$. Then, let $SdF_i^k(x_i)$ be the surrogate model of the increment function, dF_i, using $n_{SdF} + k_{SdF}$ samples in iteration k, where k_{SdF} represents added samples. The comparison of each surrogate technique is done via the Error Comparison (EC) function, which reads

$$\begin{gathered}\epsilon^k(x_i) = \left(SdF_i^k(x_i) - SdF_i^{k-1}(x_i)\right)^2 F_{pos}(\tilde{x_i}) + \\ \sum_{D=1}^{d} \frac{1}{(D+1)!} \left(\left(\frac{\partial^D SdF_i^k(x_i)}{\partial x_i^D} - \frac{\partial^D SdF_i^{k-1}(x_i)}{\partial x_i^D} \right)^2 \right)^{\frac{1}{D+1}} F_{pos}(\tilde{x_i})\end{gathered} \tag{3.2}$$

where D represents the order of derivative, d represents the maximum selected derivative order, x_i represents the stochastic random variable, $\tilde{x_i}$ represents a normalized stochastic random variable and $F_{pos}(\tilde{x}_i)$ represents a position function defined in the following. In the ideal case, the order of derivative, d, can go to infinity, while for engineering purposes the order d should be truncated to a finite number, e.g. $d = 2$ or $d = 3$. The normalized variable $\tilde{x}_i$ is defined as follows

$$\tilde{x}_i = \frac{x_i - \frac{(max(x_i)+min(x_i))}{2}}{\frac{(max(x_i)+min(x_i))}{2}} \tag{3.3}$$

where $min(x_i)$ is the minimum for the given distribution of x_i and $max(x_i)$ represents the maximum for the given distribution of x_i. Note that the infinite distributions are truncated to a finite number. The normalization assures that the value of $\tilde{x}_i$ is bounded between -1 and 1. The position function, $F_{pos}(\tilde{x}_i)$, represents a measurement of distance between samples and it reads

$$F_{pos}(\tilde{x}_i) = min\left(\sqrt{(\tilde{x}_i - K(\tilde{x}_i)(\tilde{x}_i - \tilde{\mathbf{x}}_{\mathbf{k_i}})^2) - \tilde{\mathbf{x}}_{\mathbf{k_i}})^2}\right) \tag{3.4}$$

where $\tilde{\mathbf{x}}_{\mathbf{k_i}}$ represents a vector of normalized samples used to create given surrogate model, i.e. $\tilde{x}_{k_i} = \left(\tilde{x}_{i_1}, ..., \tilde{x}_{i_{n_{SdF}+k_{SdF}}}\right)$ and function $K(\tilde{x}_i)$ is defined in the following way

$$K(\tilde{x}_i) = k_{shift}\frac{\tilde{x}_i}{\mid \tilde{x}_i \mid} \tag{3.5}$$

where k_{shift} represents a shift coefficient. The shift coefficient can be set to a constant for all engineering problems due to the process of normalization. However, the shift coefficient should be set between 0 and 0.5. The position function represents an exploration searching process over the stochastic domain, i.e. the stochastic domain is more thoroughly searched and the samples are not piled in small region of the domain. An example of the position function is showed in Fig. 3.1.

The EC function has several significant attributes. The first part of the EC function represents the difference of the interpolation process between each iteration. If the selected maximum order of derivative is 0 ($d = 0$), then the EC function is similar to the other techniques such as Generalization error (56) or Leave-one-out error (56). The derivative part of the EC function compares the shape of given surrogate models. Third part represents the position function. This part assures that the EC function is zero at the position of a sampled point and a new sample point is not positioned at already sampled positions. Moreover, the part

$$K(\tilde{x}_i)(\tilde{x}_i - \tilde{\mathbf{x}}_{\mathbf{k_i}})^2$$

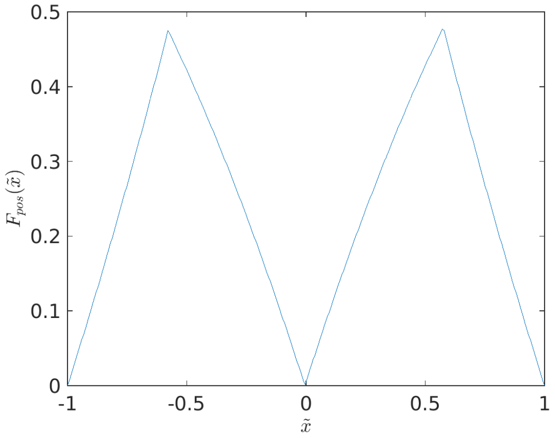


Figure 3.1: The position function created with 3 samples

of Eq. (3.4) slightly curves the position function and therefore, the spacing is not equidistant. For simple functions, this prevents the Runge phenomena (68) for polynomial interpolation and improves the search over the stochastic domain.

When the EC function is defined, the next step is to take into account the input probability distribution. Let $p_i(x_i)$ be the Probability Distribution Function assigned to a random variable x_i. The Probability Error Comparison (PEC) is then defined as follows

$$\tilde{\epsilon}^k(x_i) = \epsilon^k(x_i)p_i(x_i) \tag{3.6}$$

The final step is to obtain the position of a new sample. This is done in the following way

$$x_{new_i} = \arg\max_{x_i} \quad \tilde{\epsilon}^k(x_i) \tag{3.7}$$

The proposed approach takes into account all necessary aspects of the probabilistic surrogate modelling.

It is easy to see that in the case of a highly non-linear function, the main contribution to PEC function is given by its derivative part, while in the case of a flat plateau, the samples are spread more uniformly due to the position function. Moreover, if the surrogate model oscillates, the derivative part of PEC function takes a leading part and propose a new sample in the area of oscillations.

3.3 Numerical application of the 1-D adaptive algorithm

The analytic approach explains the theory behind the proposed algorithm, however, it is impractical for common use. Therefore, the numerical approach is proposed. Let $SdF_i^k(x_i)$ be the surrogate model of the increment function, $dF_i(x_i)$, using $n_{SdF} + k_{SdF}$ samples in iteration k. Assume a vector $\mathbf{x}_i$ of real numbers, $\mathbf{x}_i = (x_{i_1}, x_{i_2}, ..., x_{i_z})$, which are equidistantly distributed on the abscissa i and assume a vector $\mathbf{rx}_i$ of random numbers distributed accordingly to the input distribution, p_i. Then, the first and the last part of vector $\mathbf{x_i}$ are defined in the following way

$$\begin{aligned} x_{i_1} &= inf(\mathbf{rx}_i) \\ x_{i_z} &= sup(\mathbf{rx}_i) \end{aligned} \tag{3.8}$$

where inf stands for infimum and sup stands for supremum. The vector of surrogate responses can be defined in the following way

$$\mathbf{Y}_i^k = SdF_i^k(\mathbf{x}_i) \tag{3.9}$$

Derivatives used in Eq. (3.2) can be obtained directly from a surrogate technique or using the central finite difference method. The equation for the central finite differences reads

$$\delta_1 \mathbf{Y}_i^k = \frac{SdF_i^k(\mathbf{x}_i + \mathit{\Delta}_i) - SdF_i^k(\mathbf{x}_i - \mathit{\Delta}_i)}{2\mathit{\Delta}_i} \tag{3.10}$$

where $\mathit{\Delta}_i$ represents a difference step in direction i. Higher order derivatives are obtained in a similar way. For current chapter, the Lagrange interpolation is used and the derivatives are directly obtained from the interpolation model. However, later in the work, Lagrange interpolation is replaced with a multi surrogate approach and the central differencing scheme is used. The normalization of vector $\mathbf{x}_i$ (Eq. (3.3)) is done in the following way,

$$\tilde{\mathbf{x}}_i = \frac{\mathbf{x}_i - \frac{(max(\mathbf{x}_i)+min(\mathbf{x}_i))}{2}}{\frac{(max(\mathbf{x}_i)+min(\mathbf{x}_i))}{2}} \tag{3.11}$$

Using samples from vector $\tilde{\mathbf{x}}_i$, obtained in Eq. (3.11), and Eq. (3.4), values of the position function along the stochastic space can be obtained. The last step is to take Eq. (3.2) and obtain values of the EC function. The equation reads

$$\boldsymbol{\epsilon}_i^k = \left(\mathbf{Y}_i^k - \mathbf{Y}_i^{k-1}\right)^2 \mathbf{F}_{pos} + \sum_{D=1}^{d} \frac{1}{(D+1)!} \left(\left(\delta_D \mathbf{Y}_i^k - \delta_D \mathbf{Y}_i^{k-1}\right)^2\right)^{\frac{1}{D+1}} \mathbf{F}_{pos} \tag{3.12}$$

where D represents the order of derivative, d represents the selected maximum derivative order, k represents iteration of the adaptive process and $\mathbf{F}_{pos}$ represents the values of the position function obtained for vector $\tilde{\mathbf{x}}_i$. Note that $\mathbf{Y}_i$ is obtained for vector $\mathbf{x}_i$ and therefore, the given stochastic domain is thoroughly searched.

In this work, the PDF represents a weight factor. This factor helps to emphasize regions, where the PDF have a high probability and neglects the areas of a low probability. Therefore, the implementation of the probability into the proposed adaptive code is done in the following way: Let $\mathbf{rx_i}$ be the vector of random numbers defined earlier. The histogram, h_i^p, is created from the vector $\mathbf{rx_i}$, where the number of bins is equal to the number of samples in vector $\mathbf{x}_i$. The histogram is normalized to obtain the probability distribution, $\tilde{h}_i^p$. The PEC function is then calculated as follows

$$\tilde{\boldsymbol{\epsilon}}_i^k = \boldsymbol{\epsilon}_i^k \tilde{h}_i^p \tag{3.13}$$

The final step is to find the position of a new sample, which is done in the following way

$$x_{new_i} = \arg \max_{x_i} \quad \tilde{\boldsymbol{\epsilon}}_i^k \tag{3.14}$$

The search is done over samples from vector $\mathbf{x}_i$, i.e. the sample from vector $\mathbf{x}_i$ with the large value of $\tilde{\boldsymbol{\epsilon}}_i^k$ is selected as a position for new expensive function sample. Only one sample is considered in each iteration. When a new sample is found, the values of the function at a given point are obtained and stopping criteria are tested (discussed in the following section). If the algorithm is not stopped, the process is repeated from the beginning. The stopping criteria are discussed in the following parts of this chapter.

3.4 Convergence - The stopping criteria

The convergence process of the adaptive approach for the proposed method is slightly different from the other techniques. It is divided into two levels, where the first level represents the convergence of each surrogate technique separately and the second level represents the convergence of the whole model. First, the local convergence is explained.

3.4.1 Convergence - The local convergence

The basic idea of the adaptive process lies in the comparison of two surrogate models. In other words, how added new sample influences the whole interpolation process. The convergence criteria could be built directly on the PEC function, however, it is not a good convergence criterion, although it converges in the L_2 sense. Therefore, the convergence process is based on the observation of the mean and the standard deviation of the surrogate model representation. Moreover, this approach allows to compare the increment functions between themselves and therefore, reduce the number of necessary samples.

The local convergence process is applied on each surrogate model independently, i.e. each increment function is subject to an individual local convergence. Let us assume a set of increment functions ST, where each increment function is approximated with an independent surrogate model. When the model is converged, the stochastic space under which the model belongs, is not sampled anymore.

Let us now recall a useful statistical property of each increment function, the partial mean and the partial standard deviation. The partial mean and the partial variance are Eq. (2.12) and Eq. (2.15), respectively. The equations reads

$$\mu_i^k = \int_{-\infty}^{\infty} SdF_i(x_i)^k p_i(x_i) dx_i \tag{3.15}$$

$$(\sigma_i^k)^2 = \int_{-\infty}^{\infty} (SdF_i(x_i)^k - \mu_i)^2 p_i(x_i) dx_i \tag{3.16}$$

The increment function $dF_i(x_i)$ is replaced with $SdF_i(x_i)$ to emphasize that the surrogate model is used to estimate the statistical properties. The numerical approach is followed and both equations (Eq. (3.15) and Eq. (3.16)) are replaced with the following equations

$$\mu_i^k \approx \frac{1}{n_s} \sum_{t=1}^{n_s} SdF_i(\mathbf{rx}_i)^k \tag{3.17}$$

$$(\sigma_i^k)^2 \approx \frac{1}{n_s - 1} \sum_{t=1}^{n_s} (SdF_i(\mathbf{rx}_i)^k - \mu_i^k)^2 \tag{3.18}$$

where n_s represents the length of vector $\mathbf{rx}_i$. Note that PDF is explicitly given in the samples of vector $\mathbf{rx}_i$. The total mean and the total variance of the first order increment functions are defined in the following way

$$\mu^k \approx f(^c x) + \sum_{i=1}^{s_D} \mu_i^k \tag{3.19}$$

$$(\sigma^k)^2 \approx \frac{1}{n_s - 1} \sum_{t=1}^{n_s} \left(\sum_{i=1}^{n} SdF_i(\mathbf{rx}_i)^k - \mu \right)^2 \tag{3.20}$$

The residual of the mean and the variance are estimated in the following way

$$R_{\mu_i}^k = \left| \frac{\mu_i^k - \mu_i^{k-1}}{\mu^k} \right| \tag{3.21}$$

$$R_{\sigma_i}^k = \left| \frac{(\sigma_i^k)^2 - (\sigma_i^{k-1})^2}{(\sigma^k)^2} \right| \tag{3.22}$$

where k represents an iteration of the process. The variance and the mean are observed simultaneously. This is important aspect as both values have to be fully converged to ensure the accuracy of the output distribution. Note that Eq. (3.21) and Eq.(3.22) can diverge if the denominator goes to zero. This is solved by additional convergence criteria discussed in the following part.

Observing only the total residuals (Eq. (3.21) and Eq.(3.22)) is not sufficient in many cases, i.e. the number of samples is not optimal. The total variance (σ^2) and the total mean (μ) are constructed from all the involved increment functions and therefore, all the constructed surrogate models are influencing the total variance (σ^2) and the total mean (μ). In a case that one of the surrogate model is diverging, the whole convergence process is slowed. This is not desirable and therefore, one more convergence condition is applied. The logistic convergence is defined as follows

$$LR_{\mu_i}^k = \left| \frac{\mu_i^k - \mu_i^{k-1}}{\mu_i^k} \right| \tag{3.23}$$

$$LR_{\sigma_i}^k = \left| \frac{(\sigma_i^k)^2 - (\sigma_i^{k-1})^2}{(\sigma_i^k)^2} \right| \tag{3.24}$$

The normal and the logistic convergence have a very important relationship: if all involved surrogate techniques are converged, the normal and the logistic convergence hold the following condition

$$LR_{\mu_i}^k \geq R_{\mu_i}^k$$

$$LR_{\sigma_i}^{k} \geq R_{\sigma_i}^{k}$$

On the other hand, if the condition is not met, one of the surrogate models is diverging and it needs more samples. However, all the other models can be fully converged. In order to decrease the necessary number of samples, the local stopping criterion is implemented in the following way

$$\text{Logistic convergence} \quad \begin{array}{c} if \\ LR_{\mu_i}^{k+1} < R_{set_i}^{\mu} \quad or \quad R_{\mu_i}^{k+1} < R_{set_i}^{\mu} \\ and \\ LR_{\sigma_i}^{k+1} < R_{set_i}^{\sigma} \quad or \quad R_{\sigma_i}^{k+1} < R_{set_i}^{\sigma} \\ then \rightarrow stop \end{array} \quad \text{Normal convergence} \tag{3.25}$$

where R_{set} represents the residual for a local convergence process. How to set the residual is discussed later in this chapter. The proposed approach ensures that each surrogate model is handled according to its influence to the final global model, i.e. if the increment function has a significant contribution to the final global model, the samples are added. On the other hand, the non-significant increment functions are sampled with the minimum samples necessary to satisfy the global accuracy. Moreover, the convergence process is stopped if the local surrogate model is locally converged and further sampling would not bring desired improvement. This ensure that the optimal number of samples is used. The whole process of the local convergence is described in Alg. 1:

Algorithm 1 Algorithm for the local convergence of the first order increment functions - part 1

```
Initialize the process:
1. Obtain $R^{\mu}_{set_i}$ and $R^{\sigma}_{set_i}$ for each increment function
2. Obtain the maximum order of derivative $d$
3. Obtain the order of iteration $k$
4. Obtain $ST$ of the selected increment functions
while set $ST$ contains non-converged increment functions do
    for $i = 1$ to length of set $ST$ do
        1. Construct the surrogate model $SdF_i^k(x_i)$ for an increment function $dF_i$
        2. Calculate all derivatives up to the selected order $d$ for each sample in vector $x_i$
        3. Calculate the position function $\mathbf{F}_{pos}$ (Eq. (3.4)) for each sample in vector $x_i$
        4. Obtain stored values of derivatives for the surrogate model $SdF_i^{k-1}(x_i)$
        5. Calculate EC (Eq. (3.2)) and PEC (Eq. (3.6))
        6. Obtain a position of the new sample (Eq. (3.7))
        7. Store the values of derivatives for the surrogate model, $SdF_i^k(x_i)$
    end for
    1. Calculate the partial mean (Eq. (3.17)) and the partial variance (Eq. (3.18)) for each increment function
    2. Calculate the total mean (Eq. (3.19)) and the total variance (Eq. (3.20))
    if $k \geq 2$ then
        1. Calculate the normal (Eq. (3.21) and Eq. (3.22)) and the logistic residuals (Eq. (3.23) and Eq. (3.24)) for all incr. fun. from set $ST$
        for $i = 1$ to length of set $ST$ do
            if Cond. 3.25 hold then
                1. Consider the approximation of increment function, $SdF_i$, from set $ST$ as converged
            end if
        end for
        2. $k = k + 1$
    else
        1. $k = k + 1$
    end if
end while
```

Algorithm 1 Algorithm for the local convergence of the first order increment functions - part 2

```
**Finalize the process:**
1. Store the residuals of each increment function
2. Store each surrogate model $SdF_i(x_i)$
3. Store the values of the sensitivity analysis
**End**
```

3.4.2 Convergence - The Global convergence

The local convergence ensures the properties of the interpolation technique for each model. However, the accuracy of the final global model is not assured. The sum of local errors, i.e. errors for each surrogate model used, can excess the residual set by the user. This problem becomes even more significant with a large number of random variables, i.e. surrogate model used. This problem can be mathematically written in the following way

$$\sum_{i=1}^{n} R^{\mu}_{local_i} > GR^{\mu}_{set} \tag{3.26}$$

$$\sum_{i=1}^{n} R^{(\sigma)^2}_{local_i} > GR^{(\sigma)^2}_{set} \tag{3.27}$$

where R_{local_i} represents a lower value of LR_i (Eq. (3.23) and Eq. (3.24)) and R_i (Eq. (3.21) and Eq. (3.22)) and GR_{set} represents the global residual set by the user. Therefore, an approach for a global convergence is applied.

The global convergence approach is based on the convergence of the mean and the variance of the final global model, i.e. the total mean and the total variance. The process to check the global convergence starts when the partial surrogate models are locally fully converged under given threshold. The total mean and the total variance are computed at each iteration and therefore, the convergence rate of the whole process can be established. To compute the global convergence residuals, the following equations are used

$$GR_{\mu} = \left| \frac{(\mu^k) - (\mu^{k-1})}{(\mu^k)} \right| \tag{3.28}$$

$$GR_\sigma = \left| \frac{(\sigma^k)^2 - (\sigma^{k-1})^2}{(\sigma^k)^2} \right| \tag{3.29}$$

where k represents the iteration. The global residuals (GR_μ and GR_σ) represent the error of the final global model in the mean value and the variance and if the global residuals are bellow threshold set by the user, the final global model is fully converged.

Let us now focus on the aspect of not fully converged model. The first part is the selection of surrogate models, which requires more samples. Each increment function in the cut-HDMR approach has a different influence on the final model and it converges for a different number of samples. Therefore, the selection process is based on the convergence of the partial residuals, $R^{\mu}_{local_i}$ and $R^{\sigma}_{local_i}$. These residuals take into account the global influence of a given surrogate model and also, the local accuracy of a given prediction.

In order to introduce the algorithm, let STC be the set of all increment functions in the final model, which contains the values of the residuals. The residuals considered in the set STC are the maximum residuals for each surrogate model, i.e.

$$STC_i = max(R^{\sigma}_{local_i}, R^{\mu}_{local_i}) \tag{3.30}$$

The maximum residual from the whole set is selected and corresponding increment function is selected for the accuracy improve. The local accuracy (used in the Alg. 1) for the selected surrogate model is modified in the following way:

$$R_{set_i} = \frac{R_{set_i}}{2} \tag{3.31}$$

The process of selection ends, when the sum of residuals in the set STC is smaller then the global residual GR_{set}, i.e. Eq. (3.26) and Eq. (3.27) are no longer valid. The procedure is described in Alg. 2.

Algorithm 2 Selection scheme for the improvement of the first order increment functions

```
Initialize the process:
1. Obtain $R^{\mu}_{set_i}$ and $R^{\sigma}_{set_i}$ for each increment function
2. Obtain $R^{\mu}_{local_i}$ and $R^{\sigma}_{local_i}$ for each increment function and create set $STC$
3. Obtain $GR^{\mu}_{set}$ and $GR^{\sigma}_{set}$
4. Create an empty set $ST$
if $GR_{\mu} > GR^{\mu}_{set}$ or $GR_{\sigma} > GR^{\sigma}_{set}$ then
    while Eq. (3.26) and Eq. (3.27) hold do
        1. Select the increment function with the highest residual from set $STC$,
        i.e. $max(STC)$
        2. Modify $R^{\mu}_{set_i}$ and $R^{\sigma}_{set_i}$ according to Eq. (3.31) and select the corre-
        sponding increment function for the accuracy improvement, i.e. increase
        a number of samples in given domain
        3. Erase the increment function from set $STC$ and store this increment
        function in set $ST$
    end while
end if
Finalize the process:
1. Store set $ST$ for Alg. 1
2. Store residuals $R^{\mu}_{set_i}$ and $R^{\sigma}_{set_i}$ for Alg. 1
End
```

3.5 The global process and the starting conditions

The global process starts with the selection of distribution for each random variable, where each distribution is represented as the random vector, $\mathbf{rx}_i$, mentioned in Sec. 3.3. The central point is selected to be the expected value of the random vector, i.e. ${}^{c}x_i = mean(\mathbf{rx}_i)$. The expensive function is decomposed into subdomains using the cut-HDMR approach and the boundaries of given stochastic space are obtained. This is a step from the analytic approach to the numerical one, wherein the analytic approach, the distribution is assumed to be an infinite (Gaussian case) and in the numerical approach, the distribution has finite boundaries. The boundaries of the stochastic domain are easily set as the minimum and the maximum of the random vector $\mathbf{rx}_i$.

Next step is to set the desired accuracy, GR_{set}, and the maximum derivative order, d, for the adaptive algorithm. The desired accuracy is set the same for the expected value and the variance, i.e. $GR^{\mu} = GR^{\sigma}$. This step ensures that all the statistical properties are accurate as desired. The maximum derivative order should be kept low due to the computational burden and therefore, the value of 2 is suggested for engineering problems.

The algorithm starts with a sample positioned in the central point. This step is considered to be the zeroth iteration, i.e. $k = 0$. The first iteration ($k = 1$) is to set the samples on the boundaries of given domain to circumscribe the interpolation domain. In other words, to sample the boundaries of the stochastic domain, e.g. $f(x_{i_1}, {}^{\mathbf{c}}\mathbf{x}^{\mathbf{i}})$ and $f(x_{i_z}, {}^{\mathbf{c}}\mathbf{x}^{\mathbf{i}})$, where x_{i_1} and x_{i_z} are defined in Sec. 3.3. The algorithm can start the iterative process of an improvement for each increment function. The global process is described in Alg. 3.

Algorithm 3 The global process for the first order increment functions

```
**Initialize the process:**
1. Set the probability distribution (the random vector $\mathbf{rx}_i$) for each random variable
2. Set the central point ${}^{c}x_i = mean(\mathbf{rx}_i)$
3. Set the global residual $GR_{set}$
4. Assign $GR^{\mu}_{set} = GR_{set}$ and $GR^{\sigma}_{set} = GR_{set}$
5. Assign for all $R^{\mu}_{set_i} = GR^{\mu}_{set}$ and $R^{\sigma}_{set_i} = GR^{\sigma}_{set}$
6. Compute the central point $f({}^{c}x_i)$ and set the iteration to 0, i.e. $k = 0$
7. Decompose the stochastic domain according the cut-HDMR model and construct the increment functions $dF(x_i, {}^{c}x)$
8. Assign all increment functions to set $ST$
9. Sample the boundaries of the stochastic space and set the iteration to 1, i.e. $k = 1$
while $GR_{\mu} > GR^{\mu}_{set}$ or $GR_{\sigma} > GR^{\sigma}_{set}$ do
    1. Call Alg. 1
    2. Call Alg. 2
end while
**Finalize the process:**
1. Construct the final model accordingly to Eq. (2.28)
2. Sample the final model with the MC sampling and obtain the statistical properties of the problem of interest
End
```

3.6 Applied examples using the 1-D adaptive scheme

To illustrate the proposed method, two examples are selected. The first example is used for the adaptive algorithm, where the properties of the adaptive approach are showed. The second example represents a common engineering problem, where the global process is described. The results of the global process are then discussed.

3.6.1 Applied example using the adaptive algorithm

In order to illustrate the method, let us introduce a simple example. The function of interest is following

$$F_{test}(x) = (6x-2)^2 \sin(12x-4) \tag{3.32}$$

where x represents a random variable with a uniform distribution and boundaries equal to $[-1, 1]$. This function is a non-linear function and it is showed in Fig. 3.2.

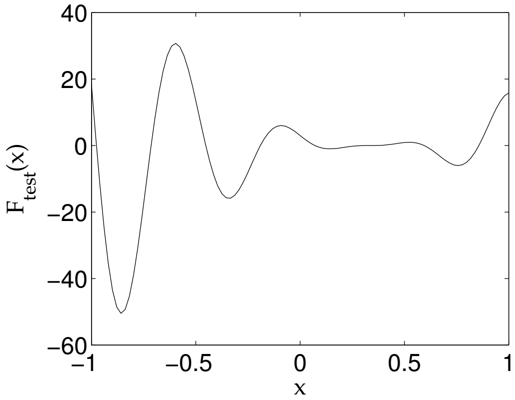
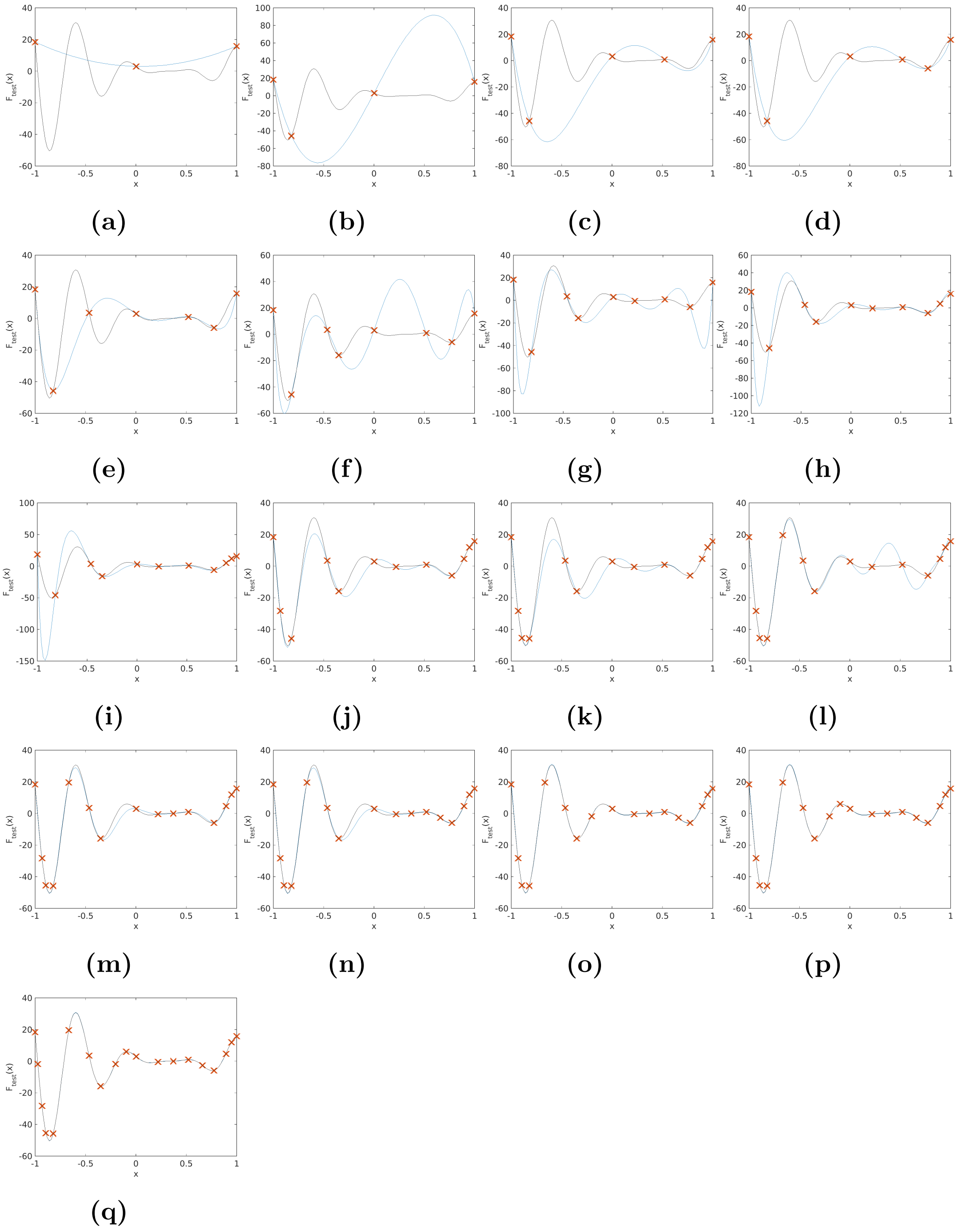


Figure 3.2: Function of interest $F_{test}(x)$

The function F_{test} (Eq. (3.32)) has all necessary problematic attributes, which can be found in engineering problems. It has a highly oscillatory region and a flat area followed by a steep ascent. All these parts represent a problem for interpolation techniques. First, let us focus on the interpolation process of the function of interest. The interpolation process is showed in Fig. 3.3, where each step in the adaptive sampling process is shown.



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q)

Figure 3.3: Process of the adaptive sampling for the function of interest $F_{test}(x)$
Blue - Interpolation
Black - Original function
Red cross - Sample from the expensive function

The function of interest is highly oscillatory and therefore, the algorithm starts to converge after a certain number of samples. An interesting part is the flat plateau, where the interpolation technique starts to oscillate. This is a natural behaviour of the Lagrange interpolation, which is prone to oscillation if the function has a steep gradient followed by a plateau. The proposed method is able to capture this phenomenon and sample the problematic region. The algorithm finds a suitable interpolation for a highly non-linear function, which is desired property for an adaptive approach as Partial Differential Equations are non-linear in general. However, in this particular case, the maximum order of derivative is selected to be $d = 5$. Such a high number for the maximum derivative is selected only for this mathematical problem due to its high non-linearity.

The convergence of the mean and the variance for the function of interest is captured in Fig. 3.4. The algorithm starts to converge after a number of samples and steep convergence can be seen. The algorithm automatically sample the areas of the greatest change, which can be seen in Fig. 3.3 between steps 3.3f and 3.3j. Therefore, it can be said that the adaptive algorithm prevents the oscillations of the interpolation technique. After the interpolation technique stops to oscillate, the function starts to steeply converge.

Another interesting aspect is the influence of the maximum selected order of derivative, d, in Eq. (3.2). The influence on the mean and the standard deviation is showed in Fig. 3.5. It can be seen that the higher order derivative starts to influence the convergence process after the overall shape is caught (Fig. 3.3o - 3.3q). This is due to the fact that the higher order derivatives become important later in the "polishing" process. However, most of the real problems converge under a low number of samples and in order to reduce the computational burden, it is suggested to keep the maximum allowed derivative low, e.g. $d = 2$.

3.6.2 Applied example for the global 1-D approach

The function selected for the application of the global process is the well known steel column problem (22). The steel column function is a 9-D problem and it is defined as follows

$$
\begin{aligned}
F(X) &= F_s - P\left(\frac{1}{2BD_{column}} + \frac{F_0}{BD_{column}H}\frac{E_b}{E_b - P}\right) \\
P &= P_1 + P_2 + P_3 \qquad (3.33) \\
E_b &= \frac{\pi^2 EBD_{column}H^2}{2L^2}
\end{aligned}
$$

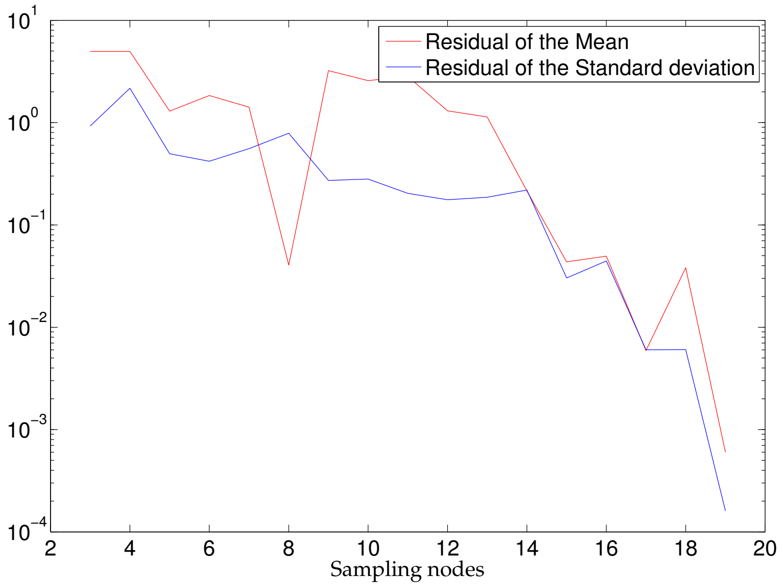


Figure 3.4: Convergence of the mean and the standard deviation for the function of interest - $F_{test}(x)$

where $F(X)$ represents a safety margin function (MPa), F_s is the yield stress (MPa), P_1 is the dead weight load (N), P_2 is the variable load (N), P_3 is the variable load (N), B is the flange breadth (mm), D_{column} is the flange thickness (mm), H is the profile height (mm), F_0 is the initial deflection (mm) and E is the Young's modulus (MPa). Distributions given to each random variable are summarized in Tab. 3.1.

ID	Random Variable	Distribution type	Mean	Standard deviation
x_1	F_s	Log-Normal	400	35
x_2	P_1	Normal	500 000	50 000
x_3	P_2	Gumbel	600 000	90 000
x_4	P_3	Gumbel	600 000	90 000
x_5	B	Log-Normal	300	3
x_6	D	Log-Normal	20	2
x_7	H	Log-Normal	300	5
x_8	F_0	Normal	30	10
x_9	E	Weibull	210 000	4200

Table 3.1: Distributions for the 9-D Steel column model

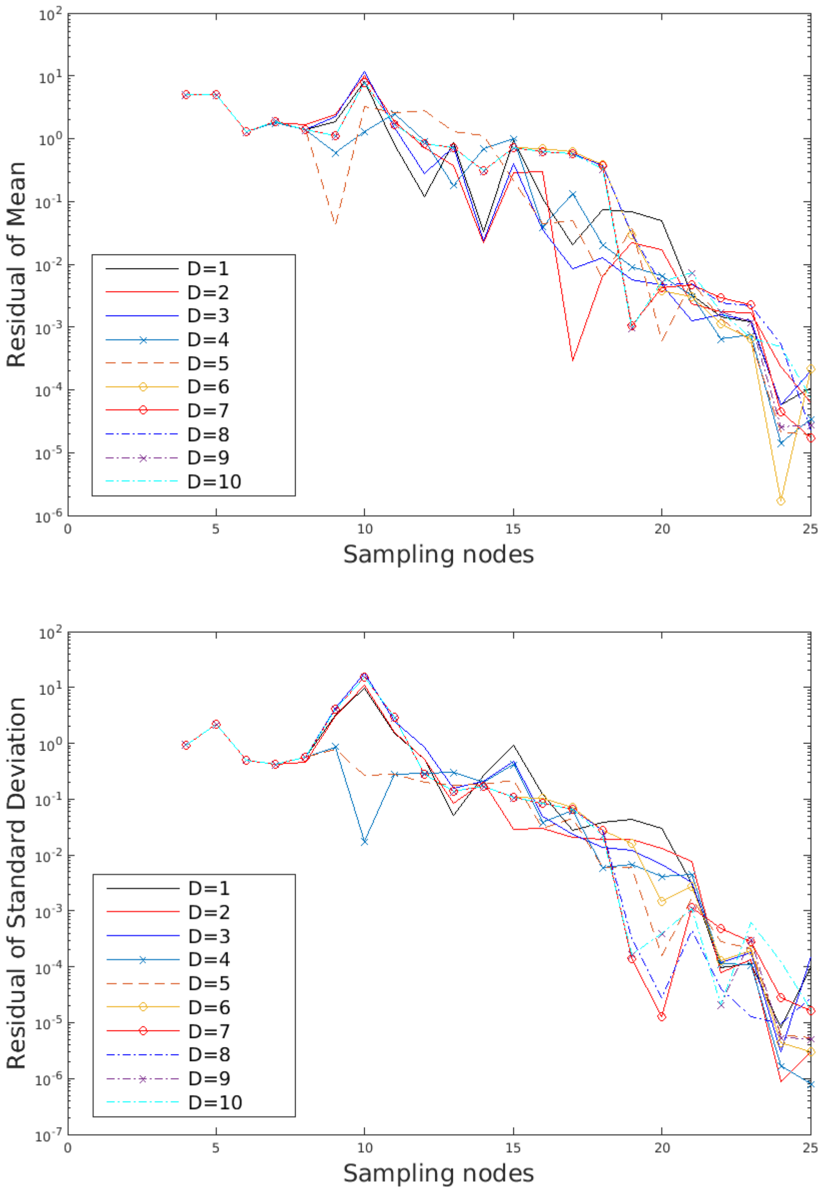


Figure 3.5: Convergence of the mean and the standard deviation for the function $F_{test}(x)$ using various order of the maximum derivative

The algorithm is tested using various thresholds and the mean value and the standard deviation are observed. The comparison is made with MC sampling with distributions from Tab. 3.1 sampled directly on the model of interest. The results of MC simulation are summarized in Tab. 3.2. The convergence of the standard deviation and the mean value for various thresholds are summarized in Tab. 3.3.

Function calls	Mean	Standard deviation
100000	2.3336e+02	44.8231

Table 3.2: MC simulation for the 9-D Steel column model

Case	Desired accuracy	Function calls	Mean	Standard Deviation	Relative error of Mean	Relative error of S. D.
1	0.1	28	2.329e+02	44.3011	1.703e-03	1.164e-2
2	0.01	30	2.333e+02	44.5635	8.818e-06	5.792e-03
3	0.001	30	2.333e+02	44.5635	8.818e-06	5.792e-03
4	0.0001	32	2.333e+02	44.5636	7.524e-06	5.789e-03

Table 3.3: The results of the Adaptive UQ-HDMR approach

The histogram obtained via MC sampling is showed in Fig. 3.6. The histograms obtained by the adaptive method for various thresholds are summarized in Fig. 3.7. The comparison graphs of the MC method and the adaptive method are showed in Fig. 3.8.

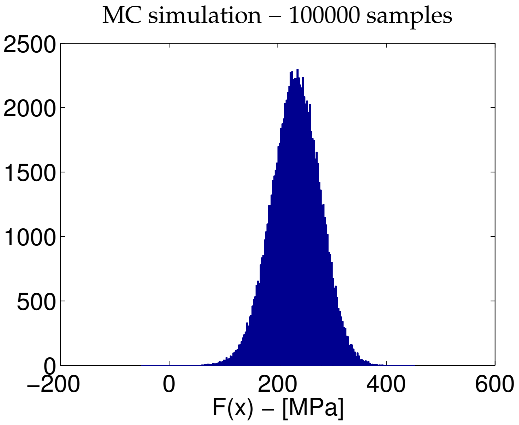


Figure 3.6: PDF obtained by MC sampling for the 9-D Steel column problem

3.6.3 Discussion about the applied example

The uncertainty quantification is performed on the steel column problem, which represents a standard engineering problem with various non-standard input distributions. Moreover, the problem is high dimensional (> 7 dimensions), which makes it challenging for the most UQ techniques. Both of these aspects represent

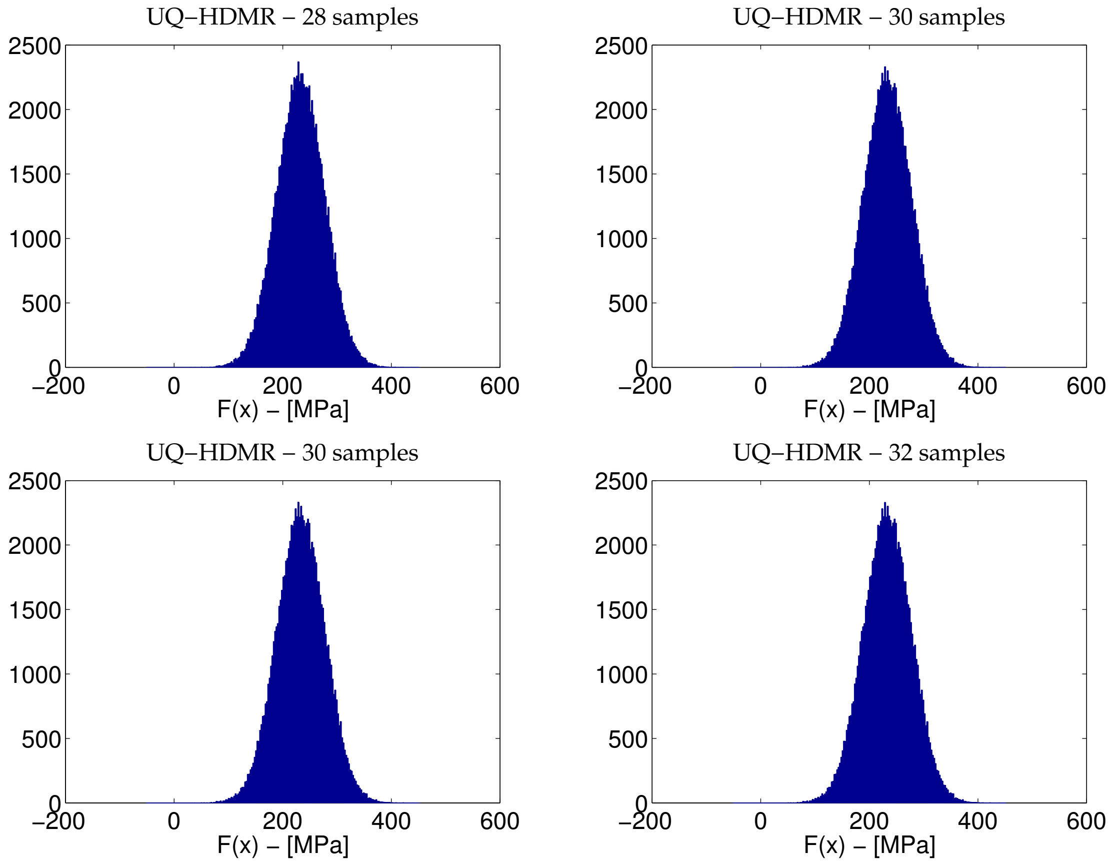


Figure 3.7: PDF obtained by the adaptive method
Left upper - case 1, Right upper - case 2, Left lower - case 3, Right lower - case 4

a significant challenge and the proposed algorithm was able to provide accurate results using only the first order increment functions. It is impossible for the algorithm to obtain higher accuracy because the interaction effects are not involved in the final model, i.e. higher accuracy would require the involvement of the higher order interaction terms in the final model. However, the method was able to successfully recognize that adding samples to the stochastic domain would not bring an increase of the accuracy. The method automatically recognizes when the limits of the first order increment functions are met and it stops further sampling. Therefore, the number of required samples is always kept optimal for given desired accuracy.

Results of the algorithm using various accuracy levels are summarized in Tab. 3.3. For the case 1, where the accuracy is set to 0.1, the algorithm uses the minimum number of samples, e.g. 3 for each abscissa plus the central point. For the cases

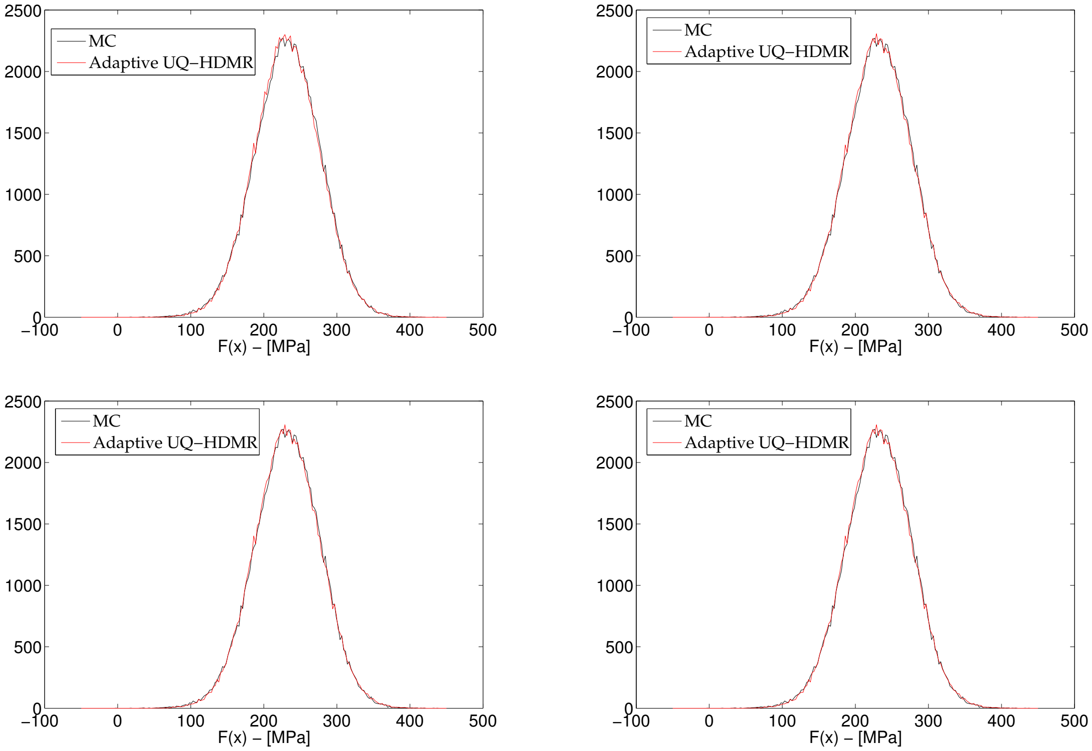


Figure 3.8: Comparison of PDF obtained by the MC sampling and the adaptive method
Left upper - case 1, Right upper - case 2, Left lower - case 3, Right lower - case 4

2 and 3, the random variable x_6 is sampled twice, i.e. the interpolation process is done with 6 samples for this variable. Other variables converged after the minimum allowed samples, e.g. 3 samples per abscissa plus the central point. The case 4 is slightly more complicated. In the first iteration, the algorithm sampled the variables x_3, x_4 and x_6. The variables x_3 and x_4 are converged after the second iteration and only the variable x_6 is further sampled. The variable x_6 is converged after 6 samples, e.g. 5 samples plus the central point. Therefore, the variable x_6 is the most problematic aspect of the whole problem.

The steel column problem is reported in (22) as a problematic due to the non-standard input probability distributions. The proposed method naturally overcome this problem as each problem is handled as a separate interpolation problem and each distribution is handled numerically. Moreover, due to an anisotropic sampling, the method uses an optimal number of samples. This is highly desired for time expensive problems such as CFD simulations.

3.7 Discussion about adaptive scheme for the first order increment functions

In this chapter, only the first order increment functions are used. This leads to a low number of samples, however, if the underlying function has strong interaction effects, one needs to implement the adaptive algorithm for the higher order increment functions. This algorithm is introduced later in the thesis.

The cut-HDMR approach allows decomposing the stochastic space, which dramatically simplifies the interpolation process. However, the overall accuracy of the interpolation process is dependent on the selection of the central point. The selection process is discussed in various works such as Gao and Hesthaven (69) and Zhang et al. (40) and the optimal selection of the central point can improve the accuracy of a solution by the order of 10. On the other hand, the optimal position of the central point is not known a priory and to select the optimal position of the central point, a random sampling has to be done. This step requires a large (more than hundred) number of samples, which are not later used and this is not acceptable for the approach taken in this thesis. Moreover, the accuracy of the final surrogate model can be improved with additional higher order increment functions introduced in the final model. Therefore, the central point is selected as the mean value of the given distributions. Also, if the central point is selected as the mean value of the given distributions, it has a physical meaning.

The adaptive algorithm is based on the idea of the comparison between surrogate models in different iterations. In other words, the question is: how new samples influence the surrogate model instead of how accurate the surrogate model is. The accuracy of the surrogate model is automatically caught in the process. In other words, if new samples do not change the surrogate model, it must mean that the surrogate model reached its maximum accuracy, i.e. it cannot be improved with additional samples. However, there is an important difference between our approach and testing the surrogate process against new samples. If only samples are used to measure accuracy in the process, the accuracy in those samples is tested, wherein in our approach, we measure accuracy also around those samples. This ensures that the final model does not have oscillatory regions and it is less prone to over-fitting.

On the other hand, a disadvantage of our approach is that it is assumed the surrogate model can approximate function according to the desired accuracy. For example, consider a linear model for a highly non-linear problem. In our approach, the model soon or later will converge, but the overall accuracy does not have to be satisfactory. However, to solve this problem, one has to manually change the interpolation technique and such that it is not suited for an automatic method. Therefore, surrogate models which always react on new samples (Lagrange interpolation, Spline etc...) are used in this work.

The proposed method searches over the whole stochastic domain and proposes a new sample to the position of need. The adaptive process takes into account multiple problematic aspects of the underlying function such as non-linearity, oscillatory behaviour and the probability of the input variables. Moreover, it is a combination of exploration and exploitation, which is highly desired in real problems. The position function represents an exploration process, where samples are uniformly distributed in a case of a simple function. On the other hand, the EC function assures that the function of interest is sampled on positions, where the interpolation process changed the most. The EC function represents the exploitation approach and in combination with the position function, it creates an automatic exploration/exploitation process. This assures that the stochastic space is optimally searched and the surrogate model is correctly converged as it can be observed on a given example in Fig. 3.3.

The EC function combined with a given probability distribution allows emphasizing the region of interest. A similar approach is used in the Polynomial chaos, see (16, 18, 19). On the other hand, the probability distribution only emphasizes the region of interest and does not influence the interpolation process. This allows to combine the proposed method with any interpolation or approximation technique and obtain a reliable uncertainty quantification method. This is done later in this thesis, where a multi surrogate approach is introduced.

The method uses all samples to interpolate the domain of interest and this can lead to an over-fitting phenomenon. However, the proposed PEC function sample mainly those parts, where the interpolation process showed fast change, which is a common behaviour of an over-fitting phenomenon. Therefore, the algorithm naturally suppresses this problem.

The convergence process is built on two aspects, where the first aspect is the convergence of the increment function and the second aspect is the convergence of the whole problem. The convergence of the increment function is considering the mean and the variance as a convergence criterion. Considering both statistical values assures that the final distribution has correct final statistical properties. The second aspect is the global convergence, which ensures convergence of the final model. It prevents unacceptable errors in a case of a large number of random variables.

3.8 Conclusion

In this chapter, the adaptive sampling process for the first order increment functions is introduced. The adaptive sampling process compares the surrogate models from different iterations and based on their differences propose a new sample. The comparison is done via the EC function, which takes into account the position of previous samples, complexity of the underlying model and the input probability distribution. Altogether, this ensures the optimal sample position for the given increment function.

In this chapter is also introduced the convergence process for the first order increment functions. The convergence process is separated into two steps. The first step represents the local convergence, where the convergence of the first order increment function is checked. The second step represent the global convergence, where the properties of the final model composed of the first order increment functions are tested. The global convergence ensure that the sum of the partial residuals is lower than the requested threshold.

The adaptive sampling scheme is tested on the two examples. In the first example, an analytic function is considered and the adaptive sampling properties are showed. In the second example, a real engineering problem is considered. In this example, the overall process is tested and properties of the whole method are showed. The proposed method shows a very good agreement with the MC simulation. Moreover, the method was able to recognize that further sampling would not bring higher accuracy and finished.

The developed algorithm in this chapter considers only the first order increment functions. Before we can focus on the higher order increment functions, it is necessary to select the influential higher order increment functions. The selection process is ensured with the selection scheme, which is topic of next chapter.

Chapter 4

Selection scheme for the high order increment functions

In this chapter, the focus is given to the Curse of Dimensionality (CoD) and its mitigation. The selection scheme selects the important increment functions and therefore, reduces the expensive samples required to interpolate the whole problem. It is based on conclusions obtained in Chap. 2, deduction logic and empirical knowledge. Contrary to commonly used interpolation techniques, the selection scheme also decides about the convergence of the global model.

This chapter is structured in the following way: First the literature review for the high dimensional interpolation is done. The literature review section is followed by the prediction theory section, which is separated into three subsections. The first subsection is the deduction approach, where is explained the inverse selection logic. The second subsection is the prediction approach, where is explained the prediction of influence for each increment function. The last subsection is the neglection approach, which neglects all the null increment functions. Once the theory is set, the selection of residuals necessary for the decision process is explained. In Sec. 4.4, the numerical application of the prediction theory and the final model convergence process is explained. The last section is given to the discussion about the prediction theory and its advantages and disadvantages.

4.1 Literature review

The Curse of Dimensionality is one of the largest issues for the approximation/interpolation methods. This aspect limits the usage of approximation / interpolation techniques to a low number of random stochastic variables. The stochastic random variables alone do not represent a significant computational burden, however, their interactions represent an exponential growth of a resource demand. CoD represents a major problem for approximation/interpolation techniques.

Currently, there are several ways how to handle the problem. One way to solve CoD is to simply restrict the problem to a lower number of the random stochastic domains, i.e. consider non-important variables as constants. However, this approach has two significant disadvantages. The first disadvantage is the restriction of the stochastic domain and this naturally leads to an inaccurate result. Moreover, this can lead to large errors in the process, especially when one is interested into tails of the output distribution. Moreover, the effect of errors is emphasized in a case, where only linear effects are established, e.g. partial derivatives or linear regression. This is usually the case of expensive problems, where one cannot afford large MC simulations.

To estimate the importance of given random stochastic variables, one needs to run sensitivity analysis. This represents a second disadvantage as the samples used for the sensitivity analysis are simply discarded. This leads to an additional computer burden, which in a case of time expensive problems such as DSMC (70), one cannot afford. On the other hand, the approach of holding certain variables constant was successfully used in various works and proved to be sufficient in many problems.

Another approach called Principal Component Analysis (PCA), developed by Pearson (71), was successfully used to mitigate the effect of the CoD problem. Series of extensions to PCA such as Probabilistic PCA (72) or Principal curves (73) were developed. Basically, PCA tries to transform the data to a new coordinate system. Each stochastic variable is sorted according to its variance and made independent. In other words, PCA tries to fit an N-dimensional ellipsoid to the data set, where each axis of the ellipsoid represents a principal component.

The PCA were extended to Proper Orthogonal Decomposition (74) (POD) method, also, known as the Karhunen-Loeve decomposition. The POD tries to approximate a function of interest over the stochastic domain as a finite sum in the variables-separated form. When the sum goes to infinity, the approximation becomes exact. The method also, modifies the coordinate system in a way, that the abscissas are aligned with the principal radii of an N-dimensional ellipsoid. The POD and PCA were successfully used for many dimension reduction problems. However, both methods rely on relatively large data set, i.e. the method is based on statistical approach. Therefore, this method cannot be used in a case of a limited data availability or an expensive code.

Interesting research into the CoD problem is done in (75, 76, 77, 78, 79). In these works, the focus is given to the exploration of the high-dimensional space and it is closely related to k-Nearest Neighbor search (80, 81). They provide interesting observations such as "hubbiness", regarding the high dimensional spaces. However, these works are not closely related to the topic of uncertainty quantification. Nevertheless, they can provide insight into problems of high dimensional modelling. These works also consider a reduction of the high dimensional space, before an approximate model is done.

In this chapter, the approach to a high dimensional space ($6 <$) is taken in a different way. The exponential growth of samples required for a high dimensional space is given by the higher order increment functions ($2 <$). Therefore, restricting the number of higher order increment functions leads to a dramatic reduction in necessary samples. However, the fundamental question is how to select the important increment functions.

In all cases, it is necessary to do a preliminary analysis of the problem. Usually, this incorporates a certain number of samples around given stochastic domain. However, if one sample the stochastic space in a smart way, the samples do not have to be neglected and they can be used further.

4.2 General prediction theory for the higher order increment functions

The prediction algorithm is based on the cut-HDMR approach and conclusions obtained in Chap 2. First, let us remind the integral form of the DE, which reads:

$$\begin{gathered} f(x_1,...,x_n) - f(^c x_1,...,^c x_n) = \sum_{i=1}^{n} \int_{^c x_i}^{x_i} \frac{\partial f(\text{¸})}{\partial \xi_i} d\xi_i + \\ \sum_{1 \leq i < j \leq n} \int_{^c x_i}^{x_i} \int_{^c x_j}^{x_j} \frac{\partial f(\text{¸})}{\partial \xi_i, \xi_j} d\xi_i d\xi_j + ... + \int_{^c x_1}^{x_1} ... \int_{^c x_n}^{x_n} \frac{\partial f(\text{¸})}{\partial \xi_1, ..., \xi_n} d\xi_1 ... d\xi_n \end{gathered} \tag{4.1}$$

where $f(.)$ is the function of interest, n is the total number of random variables, x_i is the random stochastic variable and $^c x_i$ is the central point for the given stochastic variable. The central point is selected to be the mean value of given input distributions, i.e. $^c x_i = mean(x_i)$. Also, let us remind a notation of the increment function which reads

$$dF_{i...j}(x_i,...,x_j) = \int_{^c x_i}^{x_i} ... \int_{^c x_j}^{x_j} \frac{\partial f(\text{¸})}{\partial \xi_i, ..., \xi_j} d\xi_i ... d\xi_j \tag{4.2}$$

Following this, let us remind important properties of the integral form of DE. The first important attribute is the integration part, where if the central point is equal to the second point, i.e., $^c x_i = x_i$, the whole integral part becomes zeroth and it can be neglected. Second important aspect is the derivative inside an integral part. It is clear that if the derivative is zero, the integral part is zero and the higher order integral parts, i.e. the increment functions, are also zeroth. Moreover, the increment function is driven by the derivative part and therefore, it can be used as a sensitivity measurement. In the last part, the final model is composed of a sum of these increment functions, i.e.

$$f(x_1,...,x_n) - f(^c x_1,...,^c x_n) = \sum_{i=1}^{n} dF_i(x_i) + \sum_{1 \leq i < j \leq n} dF_{ij}(x_i, x_j) + ... + dF_{i...n}(x_i,...,x_n) \tag{4.3}$$

The prediction of the important interaction effects can be divided into three steps. The first step represents the deduction approach, where the influence of interaction effect is deduced. The second step represents the prediction approach, where the influence of interaction effects are predicted and the last step is the neglection approach, where domains are neglect based on few samples in given domain.

4.2.1 Deduction algorithm

To introduce the deduction algorithm, let us assume a following arbitrary function $f(\mathbf{x})$ and point in the stochastic domain ${}^2\mathbf{x}$, where

$$ {}^2x_i \neq {}^cx_i \qquad i = 1 \cdots n \tag{4.4}$$

where n represents the number of variables and ${}^cx_i$ represents the central point and the superscript ^{2} represents the position in the stochastic domain, which is defined by user. Let us assume set M_{total}, which consists of all increment functions, i.e. the model contains all integral parts of Eq. (4.1). Next, let us assume set $M_{selected}$, which consist of selected increment functions, i.e. some integral parts of Eq. (4.1) are missing. Note that set $M_{selected}$ is a subset of set M_{total}. Two models can be composed of the proposed sets, i.e.

$$dF_{Final:total}(\mathbf{x}) = \sum_{t_t \in M_{total}} dF_{t_t}(\mathbf{x}) \tag{4.5}$$

$$dF_{Final:selected}(\mathbf{x}) = \sum_{t_t \in M_{selected}} dF_{t_t}(\mathbf{x}) \tag{4.6}$$

Eq. (4.5) is composed of set M_{total} and Eq. (4.6) is composed of set $M_{selected}$. Subtracting Eq. (4.5) from Eq. (4.6) leads to the following equation

$$dF_{Final:total}(\mathbf{x}) - dF_{Final:selected}(\mathbf{x}) = dF_{neglected}(\mathbf{x}) \tag{4.7}$$

where $dF_{neglected}(\mathbf{x})$ represents a sum of neglected increment functions, i.e. increment functions not selected. Assuming previously defined point ${}^2\mathbf{x}$, one can establish a value of the neglected increment functions at given sample, i.e.

$$dF_{Final:total}({}^2\mathbf{x}) - dF_{Final:selected}({}^2\mathbf{x}) = dF_{neglected}({}^2\mathbf{x}) = \epsilon_{Residual:total}({}^2\mathbf{x}) \tag{4.8}$$

where $\epsilon_{Residual:total}(\mathbf{x})$ represents a function of residuals, i.e. function of neglected increment functions. However, in reality, the model $dF_{Final:total}$ is usually not know. On the other hand, one can realize that the following condition must hold, i.e.

$$dF_{Final:total}({}^2\mathbf{x}) = f({}^2\mathbf{x}) - f({}^c\mathbf{x}) \tag{4.9}$$

In other words, obtaining function values at position ${}^2\mathbf{x}$ and subtracting them from the central value, gives the influence of all increment functions in the point

${}^2\mathbf{x}$. Now, if the values of function $\epsilon_{Residual:total}(\mathbf{x})$ around the stochastic domain are below given threshold, i.e.

$$\epsilon_{Residual:total}({}^2\mathbf{x}) \leq k_{\epsilon_{R-T}} \epsilon_{R-T} \tag{4.10}$$

then all the increment functions in $dF_{neglected}(\mathbf{x})$ have negligible effect on the model and they can be excluded from the model, i.e. they do not need to be sampled. The correction factor $k_{\epsilon_{R-T}}$ is used to modify the convergence criteria and ϵ_{R-T} represents the deduction residual. The correction factor and deduction residual are discussed in Sec. 4.5.

The given approach can be extended to a point, where one can select increment functions with a negligible effect. Let us assume point ${}^3\mathbf{x}$, where

$$\begin{array}{rcl} {}^3x_i & \neq & {}^cx_i \\ {}^3x_j & = & {}^cx_j \end{array} \qquad i = 1 \cdots j-1, j+1 \cdots n \tag{4.11}$$

For this particular point, some integral parts of Eq. (4.1) are zero and therefore, these integral parts can be neglected from the final model. Let us assume set $M_{total:j}$ of increment functions, which does not include the increment functions having x_j as a variable, i.e.

$$dF_{Final:total:j}(\mathbf{x}) = \sum_{t_t \in M_{total:j}} dF^j_{t_t}(\mathbf{x}) \tag{4.12}$$

where $dF^j(\mathbf{x})$ represents the increment function without variable x_j. Next, assume set $M_{selected:j}$, which consists of selected increment functions and it is a subset of set $M_{total:j}$. Note that set $M_{total:j}$ is a sub set of M_{total} and $M_{selected:j}$ is a subset of $M_{selected}$. One can construct the following model

$$dF_{Final:selected:j}(\mathbf{x}) = \sum_{t_t \in M_{selected:j}} dF^j_{t_t}(\mathbf{x}) \tag{4.13}$$

where $dF_{Final:selected:j}(\mathbf{x})$ is constructed using only selected increment functions, i.e. it is constructed from set $M_{selected:j}$. Subtracting Eq. (4.12) from Eq. (4.13) leads to the following equation

$$dF_{Final:total:j}(\mathbf{x}) - dF_{Final:selected:j}(\mathbf{x}) = dF_{neglected:j}(\mathbf{x}) \tag{4.14}$$

where $dF_{neglected:j}(\mathbf{x})$ represents all neglected increment functions, which considers variable x_j at its nominal value. Assuming previously defined point ${}^3\mathbf{x}$, one can establish a value of the neglected increment functions at given point, i.e.

$$dF_{Final:total:j}({}^3\mathbf{x}) - dF_{Final:selected:j}({}^3\mathbf{x}) = dF_{neglected:j}({}^3\mathbf{x}) = \epsilon_{Residual:total:j}({}^3\mathbf{x}) \tag{4.15}$$

where $\epsilon_{Residual:total:j}(\mathbf{x})$ represents a function of residuals for increment functions without x_j as a functional variable, i.e. a function of neglected increment functions. Usually, one does not know the model $dF_{Final:total:j}$ a priory. However, the following condition can be established

$$dF_{Final:total:j}(^3\mathbf{x}) = f(^3\mathbf{x},^c x_j) - f(^c\mathbf{x}) \tag{4.16}$$

where $f(^3\mathbf{x},^c x_j)$ represents the function value, where Cond. 4.11 holds and $f(^c\mathbf{x})$ represents the function value at the central point. In other words, obtaining function values at position $^3\mathbf{x}$ and subtracting them from the central value, gives the influence of all the increment functions, which have variable j as a functional variable. Now, if the values of function $\epsilon_{Residual:total:j}(\mathbf{x})$ around the stochastic domain are below given threshold, i.e.

$$\epsilon_{Residual:total:j}(^3\mathbf{x}) \leq \epsilon_{R-T-j} \tag{4.17}$$

then all increment in $dF_{neglected:j}(\mathbf{x})$ have negligible effect on the model and can be excluded from the final model, i.e. they can be neglected without being sampled.

To extend the deduction process further, one can invert the approach. In other words, if the difference between the full model (Eq. (4.8)) and the model without certain increment functions (Eq. (4.15)) is small, this means that the increment functions not included must have negligible effect. Therefore, subtracting Eq. (4.6) from Eq. (4.15) leads to the following equation:

$$\begin{aligned} dF_{Final:total:j}(^3\mathbf{x}) - dF_{Final:selected:j}(^3\mathbf{x}) - (dF_{Final:total}(^2\mathbf{x}) - dF_{Final:selected}(^2\mathbf{x})) \\ = dF_{neglected:j}(^3\mathbf{x}) - dF_{neglected}(^2\mathbf{x}) = dF_{not-included:j}(\mathbf{x}) = \epsilon_{Residual-not-included:j}(\mathbf{x}) \end{aligned} \tag{4.18}$$

However, in this case, samples $^3\mathbf{x}$ and $^2\mathbf{x}$ have to hold the following condition:

$$\begin{array}{rcl} ^3x_i & = & ^2x_i \\ ^3x_j & \neq & ^2x_j \end{array} \qquad i = 1\cdots j-1, j+1\cdots n \tag{4.19}$$

One can quickly realize, that this condition can be easily fulfilled for the previous two conditions (see Cond. 4.4 and Cond. 4.11) and therefore, smart sampling reduce the number of samples drastically. Finally, one can set the following neglect condition:

$$\epsilon_{Residual-not-included:j}(\mathbf{x}) \leq \epsilon_{Residual-not-included:j} \qquad \mathbf{x} \in \begin{cases} x_j = {^2x_j} \\ x_{i\neq j} = {^3x_{i\neq j}} \end{cases} \tag{4.20}$$

where $\epsilon_{Residual-not-included:j}$ represents the deduction residual defined later in this work. However, in this case, one neglects the increment functions which do have variable j as a functional variable.

Very important part is to estimate the residuals (ϵ_{R-T}, ϵ_{R-T-j} and $\epsilon_{Residual-not-included:j}$). These residuals are not independent and such that their estimation is based on a mathematical background, which is later described in this chapter.

The deduction approach is based on an inverse logic, i.e. we deduce which increment functions have a negligible effect and the remaining increment functions must be in the final model. The process is independent of the surrogate model, i.e. the deduction is done using only samples in the stochastic domain. It is very efficient in reducing the higher order interaction terms. Especially for a large number of the stochastic variables. However, to reduce the samples even more and to make the selection scheme more efficient for most functions, another selective approach is introduced.

4.2.2 Prediction algorithm

The prediction algorithm is based on an empirical knowledge that the higher order increment functions do not have a high influence on the final model. Many of the higher order increment functions have an extremely small influence on the final model yet these functions still pass through the deduction process. Therefore, it is worthy to predict the influence of the increment function and neglect the increment function if the influence is below given threshold.

The prediction algorithm is based on results obtained in Sec. 2.2. First, let us recall some basic aspects of the integral form of DE (Eq. (4.1)). The shape of the increment functions is given by the derivative part and information about the higher stochastic domains propagates via derivatives as it is concluded in Sec. 2.2. In other words, one can estimate the influence of the higher domains, using information from the lower domains.

The prediction algorithm stands on the idea that the higher order increment functions are less influential than the lower order increment functions (see Sec. 2.2). In other words, sensitivity of the higher domain is lower than sensitivity in the lower domain, e.g.

$$min(S_{dF_1}, S_{dF_2}) \geq S_{dF_{12}}$$

where S_{dF_t} represents a sensitivity of given increment function. However, it was empirically found that using the above condition is very strict and therefore, the condition for the sensitivity estimation is defined as follows:

$$pS_{dF_t} = mean(S_{dF_i}, ..., S_{dF_j}) \tag{4.21}$$

where pS_{dF_t} represents the selected sensitivity criteria, e.g. the partial mean or the partial variance. In other words, the mean value of estimated sensitivities is taken. This ensures that the predicted value is always assumed higher than it really is. The sensitivities are closely related to the variance of given increment functions and in uncertainty propagation, one does not want to check only sensitivities, i.e. variances, but also, the mean values. Therefore, the following predictions are defined

$$\begin{aligned} \mu_{i...j} &= mean(\mu_i, ..., \mu_j) \\ (\sigma_{i...j})^2 &= mean((\sigma_i)^2, ..., (\sigma_j)^2) \end{aligned} \tag{4.22}$$

where μ_t represents the partial mean value of the increment function dF_t and $(\sigma_t)^2$ represents the variance of the increment function dF_t. The increment function is neglected if the following conditions hold:

$$\begin{gathered} \mu_{i...j} \leq \epsilon^{\mu}_{prediction} \\ \& \\ \sigma_{i...j} \leq \epsilon^{\sigma}_{prediction} \end{gathered} \tag{4.23}$$

where $\epsilon_{prediction}$ is the residual set by the user to neglect the increment function. In this work, the prediction residual is set to the global residual set by the user, i.e. $\epsilon^{\mu}_{prediction} = GR^{\mu}_{set}$ and $\epsilon^{\sigma}_{prediction} = GR^{\sigma}_{set}$.

The prediction algorithm has several attributes, which are necessary to understand. The first problem is a function, where the higher order increment functions have a larger influence than the lower ones (see Sec. 2.2). In that case, the prediction algorithm neglects an important interaction. Moreover, it can happen that the lower domains are composed of badly derivable functions, where the prediction algorithm fails. This problem was observed in the work of Mehta et al. (82), where the prediction algorithm failed. However, the deduction algorithm can detect this problem and therefore, it has a higher priority for the decision process, i.e. one should consider the increment function fully neglected only if it is neglected with the deduction algorithm. To illustrate the approach, consider the following function:

$$f(\mathbf{x}) = e^{x_1 x_2 x_3}$$

and the central point is selected to be 0 for all three input variables. One can easily see that the first and second order increment functions are 0 along the whole stochastic space and such that these increment functions would be neglected. Therefore, in this case, the first and second order increment functions are sampled and considered as an important increment functions.

4.2.3 Neglection algorithm

The deduction and prediction algorithm detects the non-influential increment functions. However, it can happen that a zeroth increment function passes the selection process. The reason for this is that the deduction algorithm selects the important increment functions in a group and inside the group, where are important increment functions are also, the non-important ones. Moreover, the lower order increment functions are important and the non-important increment function is assumed to be important. It would be a waste of samples and computational time to process these increment functions. Therefore, it is necessary to use few samples to detect if the derivative in given domain is zero.

To introduce the approach, let us assume arbitrary second order increment function $dF_{ij}(x_i, x_j)$ and also, the following point:

$$\begin{array}{ll} {}^4x_i \neq {}^cx_i & \\ {}^4x_j \neq {}^cx_j & \begin{array}{l} k = 1 \cdots n \\ k \neq i \neq j \end{array} \\ x_k = {}^cx_k & \end{array} \tag{4.24}$$

where ${}^cx_i$ represents the central point and n represents the number of stochastic variables. For many functions, the higher order increment functions can be approximated with a linear model and such that their increment is getting larger, the further the we move from the central point. Therefore, the proposed point is sampled on boundaries of the stochastic domain. Using the previously sampled point and Eq. (2.26), one can establish the value of increment function at that point, e.g. for $n = 3$

$$\begin{aligned} dF_{12}({}^4x_1, {}^4x_2) = {} & f({}^4x_1, {}^4x_2, {}^cx_3) - f({}^cx_1, {}^cx_2, {}^cx_3) - f({}^4x_1, {}^cx_2, {}^cx_3) \\ & + f({}^cx_1, {}^cx_2, {}^cx_3) - f({}^cx_1, {}^4x_2, {}^cx_3) + f({}^cx_1, {}^cx_2, {}^cx_3) \end{aligned} \tag{4.25}$$

If value of Eq. (4.25) is zero, one can conclude that the underlying derivative is zero. However, it can happen that the selected point is passing the region where the increment function is zero. Nevertheless, the increment function has an influence around the stochastic domain and such that cannot be neglected. This problem can be solved by application of multiple points around given stochastic domain.

Another issue with the neglection approach is the selection of the central point. If the central point is selected in a way that the whole increment function is zeroth around given stochastic domain, the neglection approach fails. However, this problem is caught with the deduction approach and therefore, until the increment function is neglected with the deduction approach, one can put the neglected function aside. The process of application is described later in this chapter. The last problem represents the numerical errors. The cut-HDMR model is prone to the numerical errors and therefore, the following condition is given

$$\begin{aligned} if\ dF_{12}({}^4x_1, {}^4x_2) < \epsilon_{neglection} \\ then \implies neglect \end{aligned} \tag{4.26}$$

where $\epsilon_{neglection}$ represents the user defined residual and in this work, it is set to $\epsilon_{neglection} = 1^{-12}$. In this section, an applied example for the second order increment function is used. However, the same logic applies to the higher order increment functions and therefore, the neglection approach can be used for any increment function.

4.3 Selection residuals

The very large problem in the prediction theory is the deduction residual, i.e. how to set a residual to neglect given increment function. This topic is very important because it decides the overall accuracy of the final model. If one selects a large residual, the final model will be inaccurate. On the other hand, if one selects a small residual, the number of the selected increment functions will be high with a little added value, i.e. the price for the additional accuracy will be high. In this work, we focus on an efficient approach and therefore, some compromise is necessary.

First, let us have a look at the residual for the deduction approach. This residual is the most important as it has the highest priority. One can simply select the residual from experience of his/her knowledge of the problem. However, this approach is not suggested as one can set the residual inaccurately. Therefore, a more robust approach is considered in this work.

To introduce the problematic, let us start with the following experiment. Let us assume function $f(x_1)$, where only one stochastic input is considered. The function is decomposed using the cut-HDMR approach as follows:

$$f(x_1) - f(^c x_1) = dF_1(x_1) \tag{4.27}$$

We want to interpolate the given function with an approximation technique and desired relative accuracy is 0.01. One immediately see that the increment function $dF_1(x_1)$ have to be converged under a relative accuracy 0.01 as there are no other increment functions in Eq. (4.27).

Let us now assume function $f(x_1, x_2)$, where are considered two stochastic variables. This function can be decomposed using the cut-HDMR approach into increment functions, e.g.

$$f(x_1, x_2) - f(^c x_1, ^c x_2) = dF_1(x_1) + dF_2(x_2) + dF_{12}(x_1, x_2) \tag{4.28}$$

Looking at Eq. (4.28), one can realize that the required relative accuracy of increment function $dF_1(x_1)$ (Eq. (4.28)) is higher than for the increment function, $dF_1(x_1)$, from Eq. (4.27). In other words, for increment function $dF_1(x_1)$ (Eq. (4.28)), one needs less accurate surrogate model than in Eq. (4.27). This is closely related to the sensitivity of a given function, i.e. the less influential the variable or combination of variables is, the less accurate the surrogate model has to be for given increment function. Moreover, the number of samples necessary to achieve given convergence is lower and therefore, the surrogate model is simpler. This can be extended to a case, where the convergence criteria are met for a linear model. Note that the linear model is the most simple model which propagates uncertainty (see Sec. 2.2). Moreover, the convergence criteria will be achieved with a linear model, no matter what the underlying increment function is, i.e. the underlying increment function can have an arbitrary shape.

Now, let us have a look on the linear model. In a case of inputs with the same influence, the output distribution of the linear model must be Gaussian or resemble Gaussian. This is well known Central Limit Theorem (83). However, to extend this approach for our problem and to explain further aspects of the problem, let us consider the following example. Let us assume function $f(x_1, x_2) = x_1 + x_2$ with an uniform distribution for both stochastic variables defined as follows:

Variable	Prob. notation	Min	Max
x_1	p_{x_1}	0	1
x_2	p_{x_2}	0	1

Model will be linear only if it is composed of the first order increment function and therefore, the other terms are not presented. The increment function has to have the following form:

$$dF_i(x_i) = x_i \tag{4.29}$$

Now, we can compute the probability that the function will be between the maximum value (achievable with given input distributions, e.g. $max(f(1,1)) = 2$) and value lower than the maximum value. For explanation purposes, we select $f(x_1, x_2) = 1.9$. Note that this value can be achieved with multiple combination of inputs using the input distributions. Using the defined probabilities in the previous table, one can compute the probability that the output will be between previously defined values. The probability is calculated as follow:

$$P(1.9 < f(x_1, x_2) < 2) = \frac{\int_{^1p_{x_1}=0.9}^{^2p_{x_1}=1} \int_{^1p_{x_2}=0.9}^{-x_2+1} dF_1(x_1)p_{x_1} + dF_2(x_2)p_{x_2} dx_2 dx_1}{\int_{^{min}p_{x_1}=0}^{^{max}p_{x_1}=1} \int_{^{min}p_{x_2}=0}^{^{max}p_{x_2}=1} dF_1(x_1)p_{x_1} + dF_2(x_2)p_{x_2} dx_2 dx_1} \tag{4.30}$$

where $^{min}p_{x_i}$ is the lower bound of the probability distribution, p_{x_i}, $^{max}p_{x_i}$ represents the upper bound of the probability distribution, p_{x_i}, $^1p_{x_i}$ represents the lowest value of probability p_{x_i} for which we can get the required value of $f(x_1, x_2) = 1.9$ and $^2p_{x_i}$ represents the highest value of probability p_{x_i} for which we can get the required value of $f(x_1, x_2) = 1.9$. Eq. (4.30) represents the ratio of possible combination of inputs to obtain required value from the defined interval, e.g. $1.9 < f < 2$. In other words, it represents the probability that the function value will be between the given interval.

Now, let us assume an interval closer to the expected value (e.g. $E(f(x_1, x_2)) = 1$) of the defined linear model composed of increment functions (Eq. 4.29). The upper limit of the interval is selected to be $f(x_1, x_2) = 1.9$ and the lower limit of the interval is selected to be $f(x_1, x_2) = 1.8$. Notice that the difference between the results is the same for both intervals, e.g. $2 - 1.9 = 1.9 - 1.8$. The probability for given interval can be calculated as follows

$$P(1.8 < f(x_1, x_2) < 1.9) = \frac{\int_{^1p_{x_1}=0.8}^{^2p_{x_1}=1} \int_{^1p_{x_2}=0.8}^{-x_2+1} dF_1(x_1)p_{x_1} + dF_2(x_2)p_{x_2} dx_2 dx_1}{\int_{^{min}p_{x_1}=0}^{^{max}p_{x_1}=1} \int_{^{min}p_{x_2}=0}^{^{max}p_{x_2}=1} dF_1(x_1)p_{x_1} + dF_2(x_2)p_{x_2} dx_2 dx_1} \tag{4.31}$$

Notice that the bounds of the finite integral in numerator changed, i.e. the difference between the lower bound (e.g. ${}^1p_{x_1} = 0.8$) and the upper bound (e.g. ${}^2p_{x_1} = 1$) become larger. In other words, the volume represented by the integral in numerator in Eq. (4.30) is larger compared to the volume represented by the integral in numerator in Eq. (4.31). Therefore, for the linear model, the following condition must hold:

$$P(1.9 < f < 2) < P(1.8 < f < 1.9) \tag{4.32}$$

where $P()$ represents the probability of given event. Note that this condition holds only for linear models, but the input distribution can be any from the Askey scheme of input distributions (see (84)) . However, Cond. 4.32 holds only until the integral crosses the expected value, e.g. $E(f(x_1, x_2)) = 1$. After crossing the expected value, the integral in the numerator will become smaller, i.e. the probability of occurrence is going to be smaller. Using this idea, one can conclude that the propagated distribution must be Gaussian in nature. For multiple dimensions, the integrals in the numerator can be easily modified; however, for explanation purposes, only 2-D case is used.

In real cases, the influence of the input variables is not always the same as used in above example. Therefore, let us modify given function in the way that the influence of variable 1 is higher than influence of variable 2, e.g. $f(x_1, x_2) = 1.9x_1 + 0.1x_2$. The integral part of Eq. (4.31) has to be modified in the following way

$$P(2 > f > 1.9) = \frac{\int_{{}^1p_{x_1}=0.95}^{{}^2p_{x_1}=1} \int_{{}^1p_{x_2}=0}^{-0.51x_2+0.51} dF_1(x_1) + dF_2(x_2) dx_2 dx_1}{\int_{{}^{min}p_{x_1}=0}^{{}^{max}p_{x_1}=1} \int_{{}^{min}p_{x_2}=0}^{{}^{max}p_{x_2}=1} dF_1(x_1) + dF_2(x_2) dx_2 dx_1} \tag{4.33}$$

Comparing Eq. (4.30) and Eq. (4.33), one can see that the integration range is shortened for variable 1, i.e. the range of possible inputs is smaller. On the other hand, the integration range of variable 2 is larger. One can understand that smaller the integration range, the larger influence of the final distribution. In certain cases, the input variable has such a large impact on the output distribution that the condition of the output normal distribution is no longer valid. In other words, there must be at least 2 variables with the same influence to have normal output distribution. The result is confirmed in the appendix D.

4.3.1 Linear model of residuals

Application of the conclusions is relatively easy. We try to build a linear surrogate model for the neglected increment functions. In other words, estimate the statistical influence of neglected increment function with a linear model. However, it is necessary to use conclusions obtained in Sec. 4.2.1.

The first step represents a selection of samples for the linear model. These samples have to hold Cond. 4.4, i.e. all samples must differ from the central point. These samples can be positioned all over the stochastic domain, however, it is proposed to sample them on the boundaries of a given stochastic domain. This step emphasizes the influence of given interactions and therefore, the important interactions are selected in the process.

Next step represents the linear model, which reads:

$$F_{ResidualModel}(x_1, ..., x_n) = \sum_{i=1}^{n} cl_i x_i \tag{4.34}$$

where n represents the number of stochastic variables and cl_i represents an unknown coefficient, which needs to be established. In many cases, the number of samples selected are not sufficient to establish the linear model, i.e. to use technique such as the least square fitting (47). Therefore, in this work, a simplified approach is used. Each coefficient cl_i is set according to the following process

$$cl_i = \frac{mean(\boldsymbol{\sigma}_i)}{\sum_{t \in M_{selected}}^{s_D} \sigma_t} CL \tag{4.35}$$

where $\boldsymbol{\sigma}_i$ is the vector of the partial standard deviations of all involved increment functions, i.e. increment functions which have variable i as a functional variable and σ_t is the partial standard deviation of the selected increment function. The sum is done over all selected increment functions, i.e. increment functions in set $M_{selected}$.

Now, let us have a closer look on Eq. (4.35). In the previous section, it was showed that the sensitivity of input variables is extremely important for the posterior distribution. This aspect is reflected in the weighted coefficient, cl_i, where each coefficient is weighted according to its influence. This ensures that the input distributions are proportionally influencing the posterior distribution. In other words, the shape of the posterior distribution is mainly influenced by the input distributions, which have the largest impact. Moreover, this approach ensures that the influential interactions will be more weighted and such that selected for interpolation.

Next step represents the solution of Eq. (4.34). It is necessary to establish coefficient CL in Eq. (4.35), which is done by substituting Eq. (4.35) to the linear model, Eq. (4.34). The coefficient CL is solved using the following linear equation:

$$f_{residual}({}^2\mathbf{x}) = \sum_{i=1}^{n} \frac{mean(\boldsymbol{\sigma}_i)}{\sum_{t\in M_{selected}}^{s_D} \sigma_t} CL({}^2x_i) \tag{4.36}$$

where ${}^2x_i$ represents the selected point (Cond. 4.4), $f_{residual}({}^2\mathbf{x})$ represents the residual function value defined as follows

$$f_{residual}({}^2\mathbf{x}) = (f({}^2\mathbf{x}) - f({}^c\mathbf{x})) - dF_{Final:selected}({}^2\mathbf{x}) \tag{4.37}$$

The difference between function increment at position ${}^2\mathbf{x}$ ($f({}^2\mathbf{x}) - f({}^c\mathbf{x})$) and given model ($dF_{Final:selected}({}^2\mathbf{x})$ defined in Eq. (4.6)) represents the residual value. In other words, we approximate the difference between the selected increment functions and the actual increment at particular point ${}^2\mathbf{x}$. In a case of multiple samples satisfying Eq. (4.37), it is necessary to average their influence, i.e. calculate the average of the residual values and compute coefficient CL.

The proposed approach is used only if there is insufficient number of samples, i.e. the number of samples holding Cond. 4.4 is not enough to use the least square fit. In other words, if there is enough samples to create the linear model using methods such as least square fit, one should use these methods.

Once, the coefficient CL is solved, one can substitute Eq. (4.35) to the linear model (Eq. (4.34)). Now, looking at function $\epsilon_{Residual:total}(\mathbf{x})$ (Eq. (4.8)), which represents the function of neglected increment functions, one can understand that the linear model represents the approximation of neglected increment functions. In other words, the linear model is the approximation for all the neglected increment functions.

The fundamental question is why to use the linear model. We can look at the conclusions from the previous section and realize that our idea is to pose a question: "*Is the influence of the neglected increment functions on the final model so small that they can be approximated with a linear model?*". If this assumption is wrong, it is obvious that the considered increment functions cannot be neglected and must be sampled.

Once the linear model (Eq. (4.34)) is created, one can do MC simulation on this cheap model to obtain the statistical properties, i.e. the mean value and the standard deviation. These statistical properties reflect the properties of the neglected values and if these values are too large, one needs to include more increment functions in the final model. Therefore, the following condition can be establish:

$$\begin{gathered} if \\ \sigma_{neglected} < \epsilon_{deduction} \\ then \implies \textit{The final model is converged} \end{gathered} \tag{4.38}$$

where $\sigma_{neglected}$ represents the statistical properties obtained from the linear model (Eq. (4.34)) and $\epsilon_{deduction}$ represents the deduction residual, which is equal to the deduction residual defined in Sec. 4.2.1, i.e. $\epsilon_{deduction} = \epsilon_{R-T}$. In this work, the deduction residual is set to the global residual defined in Sec. 3.4.2.

The proposed approach can be extended to selection of non-important increment functions. However, one needs to replace Eq. (4.37) with the following equation

$$f_{residual}(^3\mathbf{x},^c x_j) = (f(^3\mathbf{x},^c x_j) - f(^c\mathbf{x},^c x_j)) - dF_{Final:selected:j}(^3\mathbf{x}) \tag{4.39}$$

where $dF_{Final:selected:j}$ is model defined in Sec. 4.2.1. Also, one of the coefficients cl_i in Eq. (4.35) is reduced to zero, i.e. $cl_j = 0$. As in the previous approach, observing function $\epsilon_{Residual:total:j}(\mathbf{x})$, which represents the function of neglected increment functions without increment functions considering x_j as a variable, one can understand that the linear model represents the approximation of these increment functions. In other words, the linear model is the approximation for all the neglected increment functions without x_j as a variable. The same process is applied for the selection process of increment functions, which are not included in the model. However, Eq. (4.37) is replaced with the following equation:

$$\begin{aligned} f_{residual}(^3\mathbf{x},^c x_j) = \\ (f(^3\mathbf{x},^c x_j) - f(^c\mathbf{x},^c x_j)) - dF_{Final:selected:j}(^3\mathbf{x},^c x_j)) \\ -(dF_{Final:total}(^2\mathbf{x}) - dF_{Final:selected}(^2\mathbf{x})) \end{aligned} \tag{4.40}$$

where $dF_{Final:selected}$ is model defined in Sec. 4.2.1. Again, observing function $\epsilon_{Residual-not-included:j}(\mathbf{x})$, which represents the function of increment functions not included in the model, one can understand that the linear model represents the approximation of the increment functions not included in the model.

Different aspect is the convergence criteria defined in Cond. 4.38. This condition has to be modified in the following way

$$\begin{gathered} if \\ \sigma_{neglected:2} < \epsilon_{deduction:2} \\ then \implies Remove\ the\ involved\ increment\ functions \end{gathered} \tag{4.41}$$

where $\sigma_{neglected:2}$ represents the statistical properties obtained from the linear model (either constructed with the left hand side equal to Eq. (4.39) or equal to Eq. (4.40), respectively) and $\epsilon_{deduction:2}$ represents the second deduction residual, which is equal to the deduction residuals defined in Sec. 4.2.1, i.e. $\epsilon_{deduction:2} = \epsilon_{R-T-j}$ using Eq. (4.39) as a left hand side and $\epsilon_{deduction:2} = \epsilon_{Residual-not-included:j}$ using Eq. (4.40) as a left hand side, respectively. In this work, the second deduction residual is set to the global residual defined in Sec. 3.4.2. One can conclude that the model is fully converged, when Cond. 4.38 is fulfilled. Moreover, Cond. 4.41 is fulfilled for all tested variables, when Cond. 4.38 is fulfilled.

4.4 Application of the selection scheme

The practical application of the prediction algorithm is relatively easy. However, one should always keep in mind the conclusions obtained in the previous sections. The prediction algorithm starts with the fully converged first order increment functions as it is done in the previous chapter (see Chap. 3). It should be noted that the prediction approach is done only one level up, e.g. if the first order increment functions are converged then only the second order increment functions are considered etc... For explanatory purposes, an applied example will be considered and it reads:

$$\begin{gathered} f_{Ex-Pr}(x_1, x_2, x_3, x_4) = \\ x_1 + x_2 + x_3 + x_4 + x_1x_2 + x_1x_3 \end{gathered} \tag{4.42}$$

where for each input variable, x_i, is considered a uniform distribution with bounds equal to -1 and 1. The first order increment functions are fully converged and the second order increment functions are being tested. Prediction algorithm can be divided into the following steps:

Step 1. First step represents the selection of the testing points for the deduction algorithm, i.e. selection of points, which hold Cond. 4.4. These points are suggested to be positioned on the boundaries of a given stochastic domain, where is the influence of the higher order increment emphasized. For these points, it is necessary to obtain values of the function of interest, ${}^{2}\mathbf{Y}_F = f({}^{2}\mathbf{X})$. For the proposed example, we consider sample point $x_i = 1$, where Eq. (4.42) is equal to ${}^{2}\mathbf{Y}_F = 6$.

Step 2. It is necessary to obtain values of the model (Eq. (4.6)) at the test points (Cond. 4.4), i.e.

$$dF_{Final:selected}({}^{2}\mathbf{x}) = \sum_{t_t \in M_{selected}} dF_{t_t}({}^{2}\mathbf{x})$$

Using Eq. (4.37) one can establish the values of residuals, i.e.

$$f_{residual}({}^{2}\mathbf{x}) = (f({}^{2}\mathbf{x}) - f({}^{c}\mathbf{x})) - dF_{Final:selected}({}^{2}\mathbf{x})$$

The model (Eq. (4.6)) is constructed from the selected lower order increment functions.

For our example, the model is constructed from the first order increment functions and it reads,

$$dF_{Final:selected}({}^{2}\mathbf{x}) = dF_1({}^{2}x_1) + dF_2({}^{2}x_2) + dF_3({}^{2}x_3) + dF_4({}^{2}x_4) = 4$$

The value of residual is estimated for our example as follows:

$$\begin{aligned} f_{residual}({}^{2}x_1, {}^{2}x_2, {}^{2}x_3, {}^{2}x_4) = \\ \big(f({}^{2}x_1, {}^{2}x_2, {}^{2}x_3, {}^{2}x_4) - f({}^{c}x_1, {}^{c}x_2, {}^{c}x_3, {}^{c}x_4)\big) \\ - \big(dF_1({}^{2}x_1) + dF_2({}^{2}x_2) + dF_3({}^{2}x_3) + dF_4({}^{2}x_4)\big) = \\ 6 - 4 = 2 \end{aligned}$$

Step 3. It is necessary to construct a linear model, which will reflect the influence of the neglected increment functions. In other words, we assume that the neglected functions have such a low influence that they can be approximated with a linear model (see Sec. 4.3.1). Using the residuals obtained in the previous step, one can build a linear model using Eq. (4.34), i.e.

$$F_{ResidualModel}(x_1, ..., x_n) = \sum_{i=1}^{n} cl_i x_i$$

This model is used to propagate the statistical properties, which are obtained by MC simulation. The inputs for MC simulation are defined in Sec. 3.3.

For our example, the linear model has the following shape

$$F_{ResidualModel}(x_1, x_2, x_3, x_4) = 0.5x_1 + 0.5x_2 + 0.5x_3 + 0.5x_4$$

where each coefficient is calculated with the following procedure. First, CL coefficient is established with the following equation:

$$\begin{aligned} 2 &= \frac{\sigma_1}{\sum_{t=4}^{1} \sigma_t} CL + \frac{\sigma_2}{\sum_{t=4}^{1} \sigma_t} CL + \frac{\sigma_3}{\sum_{t=4}^{1} \sigma_t} CL + \frac{\sigma_4}{\sum_{t=4}^{1} \sigma_t} CL \\ &= 0.25CL + 0.25CL + 0.25CL + 0.25CL \Rightarrow CL = 2 \end{aligned}$$

with the CL coefficient established, each coefficient is calculated using Eq. (4.35) as follows:

$$\begin{aligned} cl_1 &= \frac{\sigma_1}{\sum_{t=4}^{1} \sigma_t} CL = 0.25\ 2 = 0.5 \\ cl_2 &= \frac{\sigma_2}{\sum_{t=4}^{1} \sigma_t} CL = 0.25\ 2 = 0.5 \\ cl_3 &= \frac{\sigma_3}{\sum_{t=4}^{1} \sigma_t} CL = 0.25\ 2 = 0.5 \\ cl_4 &= \frac{\sigma_4}{\sum_{t=4}^{1} \sigma_t} CL = 0.25\ 2 = 0.5 \end{aligned}$$

Step 4. Once, the statistical properties are obtained, one can check the convergence criteria of the final model (Cond. 4.38). If the condition is fulfilled, the final model is converged and the whole process of model building is stopped.

For the proposed example, the statistical properties obtained with the linear model are following:

$$\begin{aligned} \mu_{normalized} &= 0 \\ \sigma_{normalized} &= \frac{\sigma_{ResidualModel}}{\sigma_{Final:selected}} = \frac{0.57}{1.15} = 0.5 \end{aligned}$$

where μ represents the expected value obtained from directly the linear model, $\sigma_{ResidualModel}$ represents the standard deviation obtained directly from the linear model and $\sigma_{Final:selected}$ represents the standard deviation obtained from the adaptive scheme for the first order increment functions (see Chap. 3). The standard deviation is normalized in order to provide percentual influence on the posterior distribution. In our case, the model is not considered as a converged and therefore, the process continue.

Step 5. If the condition is not fulfilled, one needs to establish, which increment functions should be added. First, it is necessary to select variable j and samples for testing, i.e. samples holding Cond. 4.11. Using these samples, one needs to obtain values at these samples, e.g. ${}^{3}\mathbf{Y}_F = f({}^{3}\mathbf{X})$.

In the proposed example, the selected variable is 1 and the proposed sample have the following coordinates:

Variable	Notation	Coordinate
x_1	${}^{c}x_1$	0
x_2	${}^{3}x_2$	1
x_3	${}^{3}x_3$	1
x_4	${}^{3}x_4$	1

The obtained value for this sample is ${}^{3}Y_F = 3$.

Step 6. Next step is to obtain value of the model (Eq. (4.13)) at the testing point (Cond. 4.11), i.e.

$$dF_{Final:selected:j}({}^{3}\mathbf{x}) = \sum_{t_t \in M_{selected:j}} dF^{j}_{t_t}({}^{3}\mathbf{x})$$

Note that the same model as in Step 2 is used. However, all the increment functions, which do not include variable j are neglected, i.e. they are zero. The value of residual is established with Eq. (4.39), i.e.

$$f_{residual}({}^{3}\mathbf{x}, {}^{c}x_j) = f({}^{3}\mathbf{x}, {}^{c}x_j) - f({}^{c}\mathbf{x}, {}^{c}x_j) - dF_{Final:selected:j}({}^{3}\mathbf{x})$$

For our example, the model has the following shape:

$$dF_{Final:selected:1}({}^{3}x_2, {}^{3}x_3, {}^{3}x_4) = dF_2({}^{3}x_2) + dF_3({}^{3}x_3) + dF_4({}^{3}x_4)$$

and the value of residual is estimated as follows:

$$\begin{aligned} f_{residual}({}^{c}x_1, {}^{3}x_2, {}^{3}x_3, {}^{3}x_4) &= \\ f({}^{c}x_1, {}^{3}x_2, {}^{3}x_3, {}^{3}x_4) - f({}^{c}x_1, {}^{c}x_2, {}^{c}x_3, {}^{c}x_4) & \\ - \left(dF_2({}^{3}x_2) + dF_3({}^{3}x_3) + dF_4({}^{2}x_4)\right) &= \\ 3 - 3 &= 0 \end{aligned}$$

Step 7. As in Step 3, one needs to build a linear model (Eq. (4.34), i.e.

$$F_{ResidualModel}(x_1, ..., x_n) = \sum_{i=1}^{n} cl_i x_i$$

Once, the linear model is build, one can propagate the statistical properties using MC simulation. The inputs for MC simulation are defined in Sec. 3.3. For our example, the linear model has the following shape

$$F_{ResidualModel}(x_2, x_3, x_4) = 0.0x_2 + 0.0x_3 + 0.0x_4$$

Note that cl_i coefficients are established in the same way as in Step 3; however, the value of residual is 0 and therefore, all cl_i coefficients has to be 0.

Step 8. The convergence criteria is checked with Cond. 4.41. If the condition is fulfilled, one needs to exclude from the final model all the increment functions up to the highest order.

In the proposed example, the statistical properties for the linear model reads:

$$\mu_{normalized:1} = 0$$
$$\sigma_{normalized:1} = 0$$

where $\mu_{normalized:1}$ represents the expected value obtained from directly the linear model, $\sigma_{normalized:1}$ represents the standard deviation obtained directly from the linear model (Step 7) and $\sigma_{Final:selected}$ represents the standard deviation obtained from the adaptive scheme for the first order increment functions (see Chap. 3). This result means that Cond. 4.41 is fulfilled and therefore, the involved increment functions can be neglected without loss of accuracy. The list of neglected increment functions read:

$$dF_{23}(x_2, x_3) \quad dF_{24}(x_2, x_4) \quad dF_{34}(x_3, x_4) \quad dF_{234}(x_2, x_3, x_4) \quad dF_{1234}(x_1, x_2, x_3, x_4)$$

Note that increment function $dF_1(x_1)$ is irelevant for the problem as it is equal to 0, i.e. $dF_1({}^c x_1) = 0$.

Step 9. It is necessary to construct the linear model using the residual obtained with Eq. (4.40), i.e.

$$\begin{aligned} f_{residual}({}^3\mathbf{x}, {}^c x_j) &= \\ f({}^3\mathbf{x}, {}^c x_j) - f({}^c\mathbf{x}, {}^c x_j) &- dF_{Final:selected:j}({}^3\mathbf{x}, {}^c x_j)) \\ &-(dF_{Final:total}({}^2\mathbf{x}) - dF_{Final:selected}({}^2\mathbf{x})) \end{aligned}$$

For our example, the residual is calculated as follows:

$$\begin{aligned}
&f_{residual}({}^cx_1,{}^3x_2,{}^3x_3,{}^3x_4) = \\
&f({}^cx_1,{}^3x_2,{}^3x_3,{}^3x_4) - f({}^cx_1,{}^cx_2,{}^cx_3,{}^cx_4) - dF_{Final:selected:1}({}^3x_2,{}^3x_3,{}^3x_4) \\
&-(dF_{Final:total}({}^2x_1,{}^2x_2,{}^2x_3,{}^2x_4) - dF_{Final:selected}({}^2x_1,{}^2x_2,{}^2x_3,{}^2x_4)) = \\
&3-3-(6-4) = -2
\end{aligned}$$

and the linear model has the following shape:

$$F_{ResidualModel}(x_2,x_3,x_4) = -0.67x_2 - 0.67x_3 - 0.67x_4$$

where each coefficient is calculated with the following procedure. First, CL coefficient is established with the following equation:

$$\begin{aligned}
-2 &= \frac{\sigma_2}{\sum_{t=4}^{2}\sigma_t}CL + \frac{\sigma_3}{\sum_{t=4}^{2}\sigma_t}CL + \frac{\sigma_4}{\sum_{t=4}^{2}\sigma_t}CL \\
&= 0.33CL + 0.33CL + 0.33CL \Rightarrow CL = -2
\end{aligned}$$

with the CL coefficient established, each coefficient is calculated using Eq. (4.35) as follows:

$$\begin{aligned}
cl_2 &= \frac{\sigma_2}{\sum_{t=4}^{2}\sigma_t}CL = 0.33\ 2 = 0.67 \\
cl_3 &= \frac{\sigma_3}{\sum_{t=4}^{2}\sigma_t}CL = 0.33\ 2 = 0.67 \\
cl_4 &= \frac{\sigma_4}{\sum_{t=4}^{2}\sigma_t}CL = 0.33\ 2 = 0.67
\end{aligned}$$

Step 10. The convergence criteria is again checked with Cond. 4.41. If the condition is fulfilled, one needs to exclude from the final model all the increment functions up to the highest order.

In the proposed example, the statistical properties for the linear model reads:

$$\begin{aligned}
\mu_{normalized:1-n} &= 0 \\
\sigma_{normalized:1-n} &= \frac{\sigma_{ResidualModel}}{\sigma_{Final:selected}} = \frac{0.66}{1.15} = 0.57
\end{aligned}$$

where μ represents the expected value obtained from the linear model, $\sigma_{ResidualModel}$ represents the standard deviation obtained from the linear model (Step 9) and $\sigma_{Final:selected}$ represents the standard deviation obtained from the adaptive scheme for the first order increment functions (see Chap. 3).

The standard deviation is normalized in order to provide percentual influence on the posterior distribution. This result means that Cond. 4.41 is not fulfilled and therefore, the involved increment functions can not be neglected without loss of accuracy.

Step 11. Repeat Step 5 - Step 10 for all variables. Once, all the variables are tested, one can proceed to the prediction algorithm.

For our example, the selected test samples have the following coordinates:

Selected variable j	x_1	x_2	x_3	x_4
2	1	0	1	1
3	1	1	0	1
4	1	1	1	0

Obtained residuals for each variable and neglected increment functions:

Selected variable j	Residual using Eq. (4.39)	Residual using Eq. (4.40)	Neglected increment functions
2	1	-1	-
3	1	-1	-
4	2	0	$dF_{23}(x_2, x_3)$, $dF_{13}(x_1, x_3)$, $dF_{12}(x_1, x_2)$, $dF_{123}(x_1, x_2, x_3)$, $dF_{1234}(x_1, x_2, x_3, x_4)$

Note that we provide only the values of residuals in order to show the process. The liner models and corresponding statistical properties are not showed. Also, the list of neglected increment functions is created as a disjunction of all the neglected increment functions from the series of tests.

Step 12. The prediction algorithm takes into account only increment functions, which passes the deduction algorithm and they are one level higher than the current model, e.g. the model includes only the first order increment functions and only the second order increment functions are tested. The first step is to compute the statistical properties of converged increment functions. For the convergence of the first order increment functions see Chap. 3 and for the higher order increment functions see Chap. 5. Once, the statistical properties of involved increment functions are established, one can predict the statistical influence using Eq. (4.22).

For our example, the selected increment functions added to the model are: The predicted sensitivities for the proposed increment functions are:

$$dF_{12}(x_1, x_2) \quad dF_{13}(x_1, x_3)$$

Increment function	Lower order increment functions	μ_t	σ_t	Predicted μ_t	Predicted σ_t
$dF_{12}(x_1, x_2)$	$dF_1(x_1)$	0	0.33	0	0.33
	$dF_2(x_2)$	0	0.33		
$dF_{13}(x_1, x_3)$	$dF_1(x_1)$	0	0.33	0	0.33
	$dF_3(x_3)$	0	0.33		

Step 13. Using Cond. 4.23 and the predicted statistical properties, one can select which increment functions should stay in the final model. However, if all the increment functions from this step are neglected and Step 4 (Cond. 4.38) is not fulfilled, it means that the prediction failed. Therefore, it is necessary to consider all the increment functions as functional, i.e. they need to be sampled and interpolated.

For our example, the predicted statistical properties suggest that all the increment functions are important and such that they need to be passed to the neglection algorithm.

Step 14. The last step in the selection process is the neglection approach (see Sec. 4.2.3). First, sample the stochastic domain with samples holding Cond. 4.24. One does not need a large set of samples and in this work, only one sample is used and it is positioned in the corner of the stochastic domain given by the increment function. With Eq. (4.25) one can establish if the increment function is zero, i.e. it can be neglected.

For the proposed example, we consider one sample for each increment function. The coordinates of the proposed samples are summarized in the following table:

Sample	x_1	x_2	x_3	x_4
1	1	1	0	0
2	1	0	1	0

Obtained values for each increment function are summarized in the following table:

Increment function	Obtained increment
$dF_{12}(x_1, x_2)$	1
$dF_{13}(x_1, x_3)$	1

Step 15. The increment function is neglected if Cond. 4.26 is fulfilled. However, it can happen that the increment function is zeroth in the whole stochastic domain yet the deduction algorithm requires a higher order increment function. For example, consider the following function:

$$f(x_1, x_2, x_3) = e^{x_1 x_2 x_3}$$

where the central point is selected to be 0 for all three input variables. In this case, the second order increment functions are 0; however, the zeroth derivative condition is not holding, i.e.

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}} \neq 0 \qquad \forall \mathbf{x} \in R^n$$

In this case, the neglection algorithm fails to neglect a non-influential increment function, i.e. the second order increment function is necessary for the third order increment function. Therefore, the increment function has to be sampled and included in the final model. Fortunately, this problem is caught with the deduction algorithm.

For our example (Eq. 4.42), both the obtained increment functions have non-zero increment and therefore, they have to be included in the final model. These increment functions are going to be sampled with the N-dimensional adaptive algorithm and these increment functions are:

$$dF_{12}(x_1, x_2) \quad dF_{13}(x_1, x_3)$$

Once, the second order increment functions are sampled and converged (see Chap. 5), the model of selected increment functions (Eq. (4.6)) will be enriched with these increment functions and the full model will have the following increment functions:

$$\begin{array}{llll} dF_1(x_1) & dF_2(x_2) & dF_3(x_3) & dF_4(x_4) \\ dF_{12}(x_1, x_2) & dF_{13}(x_1, x_3) & & \end{array}$$

This new model is then tested for the convergence (Step 4) and for the proposed example, the residual is 0. Therefore, the final model is converged and the whole process is considered as finalized.

The selection process is showed in Fig. 4.1.

4.5 Discussion about the selection scheme and selection residuals

One of the largest problems in the uncertainty propagation field is CoD, which represents a major obstacle in the high dimensional interpolation. The prediction approach is a robust way to estimate the influence of the increment functions and therefore, reduce the burden of high dimensionality. In recent years, it was a custom to stop the cut-HDMR model at the second order increment functions. However, this approach is not accurate enough and it could provide serious errors in the estimation of the final PDF. For example, consider the following function:

$$f(x_1, x_2, x_3) = e^{x_1 x_2 x_3} \tag{4.43}$$

where the central point is selected to be 0 for all three input variables. In this case, the previously mentioned approach would provide a completely wrong solution. Moreover, if one is interested in tails of the final PDF, the high order increment functions ($> 2nd$ order increment functions) play a vital role. The proposed selection scheme has a safety mechanism to avoid false prediction and therefore, it is a robust estimator.

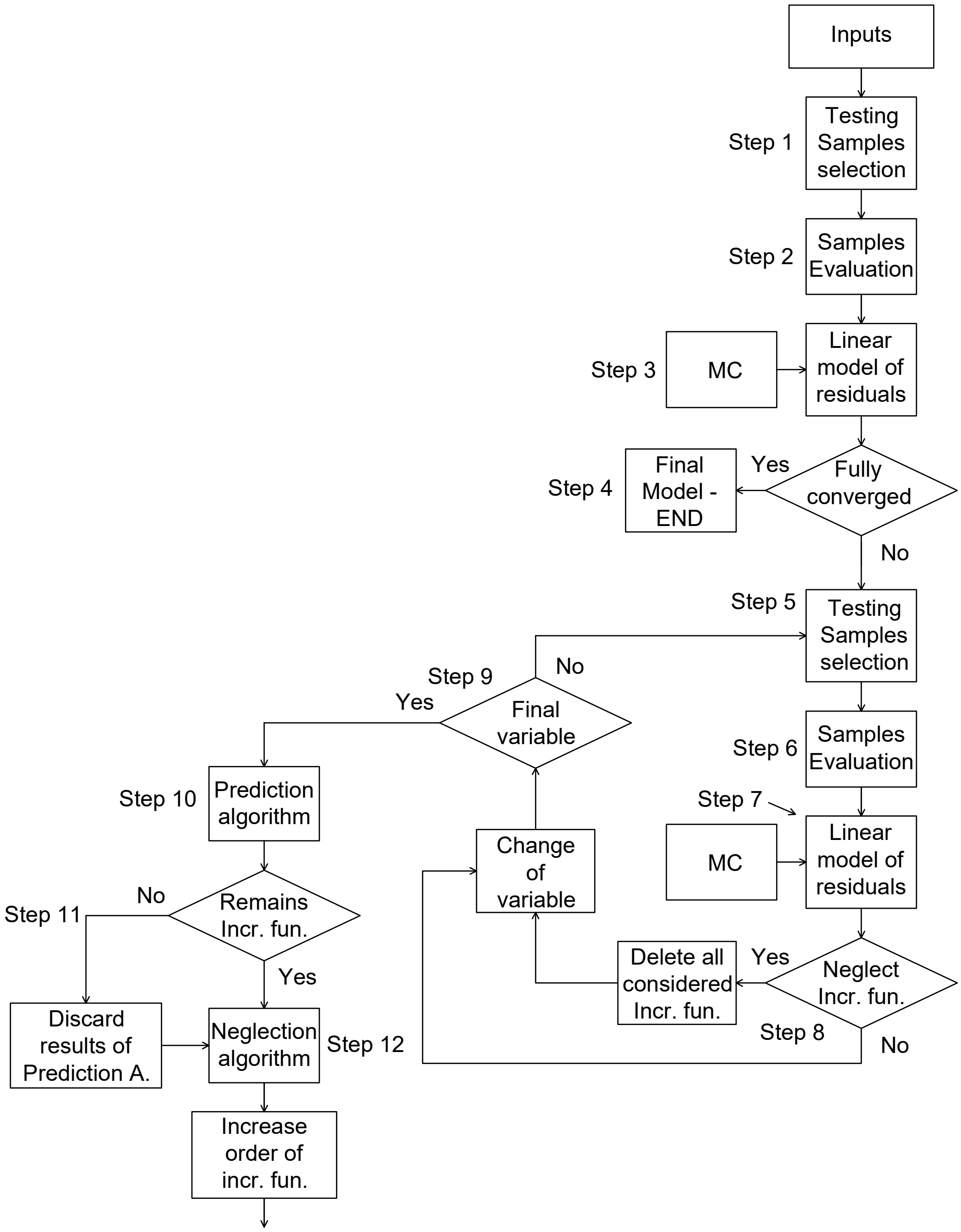


Figure 4.1: Flow chart for the selection scheme

One can quickly realize that the minimum number of samples necessary to perform the deduction approach is $n + 1$ and unfortunately, these samples are not used in any other process. One can say that using one sample for testing the neglected variables is not enough, i.e. creating the linear model of residuals using only one sample. However, it was found that for the selection process, one sample is sufficient. On the other hand, when only one sample is selected, the position of a sample can lead to over-estimating the influence of neglected increment functions. This problem occurs in cases with strong interaction terms such as the wing weight model (24). Moreover, the problem is magnified if the test sample is selected on the edge of the stochastic domain, e.g. ${}^{2}\mathbf{x} = max(\mathbf{rx}_1), ..., max(\mathbf{rx}_n)$. In this case, the input distribution is not taken into account and the influence of non-important domains is emphasized. However, in order to simplify the work, we position all the test points on the boundaries of given stochastic domain. The model is considered fully convergent when the deduction approach neglects all non-important increment functions using Cond. 4.38. This leads to an efficient estimation of the important increment functions.

The deduction approach has one important property. If the test samples are sampled in a smart way, one can completely avoid error in the interpolation technique. In other words, made the deduction approach independent of the interpolation technique. To explain this further, let us consider a three dimensional case, where all the stochastic random variables have a uniform distribution with bounds $[-1\ 1]$. The central point is selected at $[0\ 0\ 0]$. The first order increment functions have to have samples at boundaries and therefore, let us assume the following samples: the first sample is $[1\ 0\ 0]$, the second sample is $[0\ 1\ 0]$ and the third sample is $[0\ 0\ 1]$. The deduction approach requires a sample to hold Cond. 4.4, e.g. test sample ${}^{2}\mathbf{x}$ at position $[1\ 1\ 1]$. One can easily notice that selected samples hold a cross section condition, which is defined later in Chap. 5 and a surrogate model is not required to perform the deduction approach. The same process can be applied to the selection process for the important increment function and therefore, the process is independent of a surrogate model.

In theory, the prediction approach should be considered a fully converged, when Cond. 4.38 is fulfilled. However, in practice, the full model can be considered fully converged if the prediction algorithm neglects all tested increment functions and Cond. 4.38 is fulfilled to a certain degree. Therefore, we set the correction factor $k_{\epsilon_{R-T}}$ to 1 in a case of convergence testing in Step 4 and in a case of convergence testing in Step 13, the correction factor $k_{\epsilon_{R-T}}$ is set to 10.

One further observation should be made from the deduction algorithm. The number of important increment functions increases when the accuracy is increased. This leads to a dramatic increase in samples, especially if the high order increment functions are considered. On the other hand, if only the first order increment functions are considered, the growth is only linear, i.e. very slow compared to what would be expected. This leads to the following conclusion:

Proposition 4.5.1 *The grow of samples in the sense of the Curse of Dimensionality is a problem of the desired accuracy and not dimensionality.*

In other words, one can construct a model using only the first order increment functions and completely avoid the CoD problem[1]. However, in many cases, the accuracy of the model is not sufficient. Therefore, it is necessary to apply a selection process for the higher order increment functions.

Unfortunately, the deduction algorithm can select the non-important increment functions. This is due to a fact that the increment functions are handled as a group, i.e. each test is performed on a selected group of increment functions (see applied example in Step 8). Therefore, it can happen that inside the group is a combination of important and also, non-important increment functions. These increment functions remain in the process and such that they would be sampled. This represents an additional computational burden and therefore, to further reduce the number of increment functions, the prediction algorithm is introduced.

The prediction algorithm is based on the conclusions obtained from DE (see Sec. 2.2). The main advantage of the prediction is the neglection of the non-important stochastic domains without sampling the domain. However, the prediction is never certain and it should not be used as a sensitivity measure. The only application should be on neglection or acceptance of the increment function. Moreover, the prediction algorithm can falsely neglect the influential stochastic domain under specific conditions. For example, consider Eq. (4.43), where the tested increment functions are of the second order. In this case, the prediction fails and provides the wrong solution. However, in combination with the deduction algorithm, it creates a robust selection scheme which prevents false neglections.

[1] In this work, CoD is understood as an exponential growth of samples when new dimensions are added to the problem of interest.

The neglection algorithm is the last part of the selection scheme. It is based on assumption that the underlying derivative is zero. This approach helps to further reduce the number of required samples in a case that the non-important increment functions passed both previous selection processes. To explain the case, consider the following function:

$$f_{Ex-Pr:2}(x_1, x_2, x_3, x_4) =$$
$$x_1 + x_2 + x_3 + x_4 + x_1x_2 + x_3x_4$$

where for each input variable, x_i, is considered a uniform distribution with bounds equal to -1 and 1. This example differs from Eq. (4.42) in the second interaction effect and in this case, the deduction and prediction algorithm selects all the second order increment functions. However, one can easily notice that the large part of the increment functions are zero and such that they do not have to be sampled. In this case, further sampling would only lead to a computational burden and therefore, the neglection algorithm further reduce the required samples.

In theory, the definition holds for all functions. In practice, one can sample the point in a wrong way and neglect an influential function. For example, consider previously defined function (Eq. (4.43)) and the tested increment functions are of the second order. These increment functions are zero along the whole stochastic domain and such that they would be neglected. On the other hand, they are required for the third order increment function and they need to be sampled. However, this problem is extremely rare as it requires specific conditions. Nevertheless, in combination with the deduction approach, it creates a robust selection scheme.

4.6 Conclusion

In this chapter, the selection scheme is introduced. The selection scheme selects the important increment functions and such that prevents the exponential growth in the number of required samples. The selection process is done in three ways. The first way represents the deduction algorithm, which is based on an inverse logic and conclusions obtained from the derivative equation. This step requires some minimal sampling to be performed and these samples are not later used. However, the overall gain in the efficiency (in terms of expensive samples) outweigh the commonly used schemes such as sampling all the lower order stochastic domains at given order.

The second way represents the prediction approach, which predicts the influence of the non-sampled increment functions. The prediction is based on a combination of empirical experience and conclusions obtained in Chap. 2. It brings additional savings in the computational time. However, the prediction is not sufficiently accurate to be used in the sensitivity analysis and such that it should be used only in the decision process. Also, it was found that the prediction can fail for certain types of problems. These failures are caught with the deduction algorithm.

The third way represents the neglection approach, which estimates the influence of the higher order increment function with a small number of samples. The neglection scheme is based on conclusions obtained in Sec. 2.2 and represents an additional step towards improvement in efficiency. However, the neglection scheme can fail in certain rare cases, but this is again caught with the deduction algorithm.

Very important aspect of the prediction theory is the threshold selection. For the deduction algorithm, the threshold is based on an approximation of the neglected increment functions and such that establishing their influence on the final model. This step allows implementing the probability distributions into the neglection process. The threshold for the prediction approach and for the neglection approach is based on an empirical experience.

The selection scheme is completely new way, how to estimate the influence of the higher order interactions. Combination of all three algorithms creates a robust estimator and brings large savings in the expensive function calls. Its importance becomes significant when the number of random variables is higher than 4. In the high dimensional spaces (> 10), the selection scheme becomes a necessity as the required number of samples goes to thousands. Usually one cannot afford such a large pool of samples.

Chapter 5

Adaptive algorithm - The higher order increment functions

In this chapter, the focus is on the development of the adaptive sampling scheme for the higher order increment functions. The adaptive scheme sample the increment functions, which are passed from the selection scheme. Same as in the first order adaptive scheme, the high dimensional adaptive scheme considers the local and global convergence process. In the local process, the convergence of the higher order increment function is checked and in the global process, the convergence of the order of increment functions is tested. However, the final convergence decision still remains on the selection scheme.

The chapter is structured in the following way: The theory of the sampling scheme for the N-D problems is introduced in the first section. In the second section, the numerical application is given. In the third section, the convergence theory is presented, where the local and global convergence processes are explained. The fourth section is given to the simple N-D toy problem, where properties of the sampling scheme are shown. The same section considers the Borehole problem, where the global process is showed and results are compared to the MC simulation. The applied example is followed by a discussion and the last section is given to the conclusion.

5.1 Basic theory of the high dimensional adaptive algorithm

To introduce the theory of the higher order adaptive scheme, let us recall the higher order increment function. The higher order increment function reads:

$$dF_{i...j}(x_i, ..., x_j) = \int_{c_{x_i}}^{x_i} \cdots \int_{c_{x_j}}^{x_j} \frac{\partial f(x)}{\partial x_i, ..., x_j} dx_i \cdots dx_j$$

Each increment function is a separate problem and similarly, as in the case of the 1-D scheme, the adaptive algorithm is applied to each increment function separately. In other words, the adaptive scheme is working only with a given increment function and its underlying stochastic domain. The adaptive algorithm is based on a comparison of surrogate models in each iteration and to compare the surrogate models, the idea of Taylor expansion is used. However, the expansion is done on errors between each iteration, which is similar to the 1-D adaptive scheme.

Let $SdF^{k-1}_{i...j}(x_i, ..., x_j)$ be a surrogate model of the increment function, $dF_{i...j}(x_i, ..., x_j)$, which is constructed with n_{SdF} samples in iteration $k-1$. Then, let $SdF^{k}_{i...j}(x_i, ..., x_j)$ be a surrogate model of the increment function, $dF_{i...j}(x_i, ..., x_j)$, which is constructed with $n_{SdF}+k_{SdF}$ samples in iteration k, where k_{SdF} represents additional samples in the stochastic domain. The comparison of surrogate models is done using the Multi Dimensional Error Comparison (MDEC) function, which reads

$$\epsilon^k_{ND}(x_i, ..., x_j) = \sum_{D_1=0}^{d} \cdots \sum_{D_{n_L}=0}^{d} \frac{1}{max(D_1, ..., D_{n_L})!} \left(\left(\frac{\partial SdF^k_{i...j}(x_i, ..., x_j)}{\partial x_i^{D_1} ... \partial x_j^{D_{n_L}}} - \frac{\partial SdF^{k-1}_{i...j}(x_i, ..., x_j)}{\partial x_i^{D_1} ... \partial x_j^{D_{n_L}}} \right)^2 \right)^{\frac{1}{max(1, D_1, ..., D_n)}} Pos_{ND}(\tilde{x_i}, ..., \tilde{x_j}) \quad (5.1)$$

where d represents the selected maximum order derivative, k represents the given iteration, x_i represents the corresponding variable and $\tilde{x_i}$ represents the normalized corresponding variable x_i, which is defined later. Note that the number of summands in Eq. (5.1) is given by the number of variables in a given increment function. $Pos_{ND}(\tilde{\mathbf{x}})$ represents the multi dimensional position function, which is a measurement of a distance between known samples and it is defined in the following way

$$Pos_{ND}(\tilde{x}_i, ..., \tilde{x}_j) = min\left(\sqrt{\sum_{h=i}^{j}(\tilde{x_h} - K(\tilde{x_h})(\tilde{x_h} - \mathbf{x_{k_h}})^2 - \mathbf{x_{k_h}})^2}\right) \tag{5.2}$$

where the vector operations are considered element-wise, $\tilde{x}_i$ represents the normalized random variable x_i, $\mathbf{x}_{k_i}$ represents the vector of normalized samples of variable i, which is used to create surrogate model $SdF^k_{i...j}(x_i, ..., x_j)$, i.e. $\mathbf{x}_{k_i} = \left(\tilde{x}_{k_{i_1}}, ..., \tilde{x}_{k_{i_{n_{SdF}+k_{SdF}}}}\right)$ and the last part is the shift function $K(\tilde{x}_i)$, which is defined in the following way

$$K(\tilde{x}_i) = k_{shift}\frac{\tilde{x}_i}{\mid \tilde{x}_i \mid} \tag{5.3}$$

where k_{shift} is a shift coefficient, which is set to 0.3 in this work. The normalized random variable is defined in a same way as in the 1-D case and the equation reads

$$\tilde{x}_i = \frac{x_i - \frac{(max(x_i)+min(x_i))}{2}}{\frac{(max(x_i)+min(x_i))}{2}} \tag{5.4}$$

where $min(x_i)$ is the minimum for given distribution of x_i and $max(x_i)$ represents the maximum for given distribution of x_i. Note that in a case of infinite or semi infinite distribution, the distribution is truncated to a finite number of standard deviations. In other words, the stochastic domain is bounded by a minimum and a maximum, which is defined by the user. The normalization process ensures that the value of $\tilde{x}_i$ is bounded between -1 and 1. Example of the position function Pos_{ND} for two random variables is shown in Fig. 5.1.

The MDEC function represents search over the stochastic domain, which takes into account behaviour of the function and also, a position of the known samples. If the maximum derivative order is set to $d = 0$, then the technique is similar to techniques such as Generalization error (56) or Leave-one-out error (56). In Eq. (5.1), the derivative part compares the shapes of interpolants in different iterations, which helps the search algorithm to select the next sample, i.e. the position of the largest change. Note that all derivatives in Eq. (5.1) are taken into account and therefore, the search is done over all interactions parts of the given domain.

The multi dimensional position function assures that MDEC function is zero at the position of a sampled point. It represents an exploration approach to the

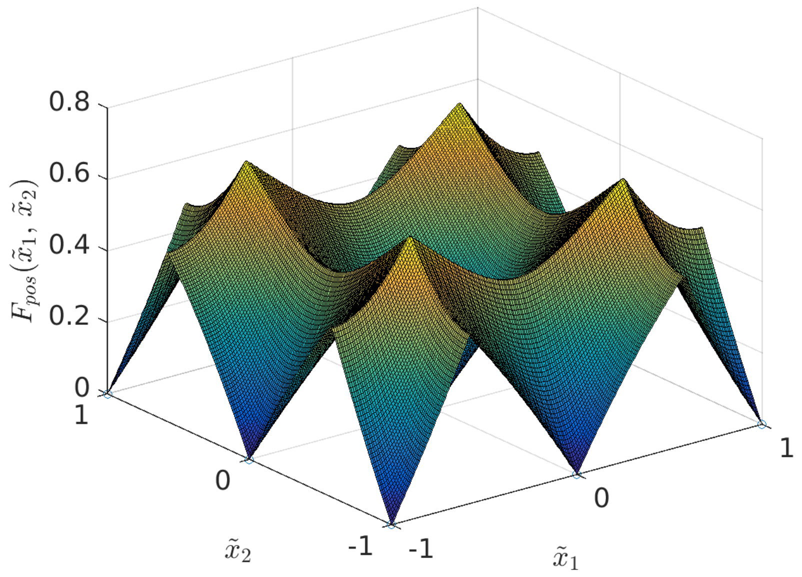


Figure 5.1: The multi dimensional position function, $Pos_{ND}(\tilde{\mathbf{x}})$, created with 9 samples

problem, where the derivative part represents the exploitation approach. A combination of these two approaches creates a natural exploitation and exploration scheme. The scheme is balanced by the nature of the function of interest. In the case of a simple function, the position function helps to spread samples more uniformly around given stochastic domain. This ensures that the stochastic space is rigorously searched and the algorithm is not stacked in one particular area. On the other hand, if the function of interest is complex, the derivative part force the algorithm to sample the problematic part. Therefore, samples in the function of interest are optimally spread around the stochastic domain.

The random variable can have various PDF, which needs to be taken into account. In the proposed scheme, PDF represents a weight function, which emphasizes the region of interest. To implement the probability distribution into the process, it is necessary to modify MDEC function.

To introduce the modification, let $p_i(x_i)$ be the Probability Distribution Function assigned to the random variable x_i. Then, the Joint Probability Distribution Function (JPDF) is defined in the following way

$$p_{i...j}(x_i, ..., x_j) = \prod_{h=i}^{j} p_h(x_h) \tag{5.5}$$

This step allows to take into account the probability distribution for the higher order increment functions. Combining MDEC function (Eq. (5.1)) and JPDF (Eq. (5.5)), the Multi Dimensional Probability Error Comparison (MD-PEC) function can be defined in the following way

$$E_{ND}^{k}(x_i, ..., x_j) = \epsilon_{ND}^{k}(x_i, ..., x_j) p_{i...j}(x_i, ..., x_j) \tag{5.6}$$

The proposed function takes into account all aspects of the uncertainty propagation. The derivative part represents behaviour of the underlying function, while the position function ensures uniformity of samples around the stochastic domain in a case of a simple function. Moreover, JDPF assures that the input probability is taken into account.

The final step is to propose the position of a new sample. One can use the constrained optimization algorithm, however, the search over the stochastic domain is complicated with restrictions. Therefore, an optimization approach is introduced later.

The search for a new sample is not over the whole stochastic domain. First, let us recall DE (Eq. (2.5)), which reads

$$df(\mathbf{x}) = \sum_{i=1}^{n} \frac{\partial f(\mathbf{x})}{\partial x_i} dx_i + \sum_{1 \le i < j \le n} \frac{\partial f(\mathbf{x})}{\partial x_i, x_j} dx_i dx_j + ... + \frac{\partial f(\mathbf{x})}{\partial x_1, ..., x_n} dx_1 ... dx_n$$

The higher order partial derivatives already includes the lower order partial derivatives. In other words, the information from the lower stochastic domains propagates to the higher ones (see Sec. 2.2). Moreover, each proposed sample requires a sample from the lower stochastic domain, i.e. each sample has to hold cross-section condition (see below). Therefore, one can assume that the best position for the new sample will lay in the cross-section of samples from the lower domains. In other words, the best trade off between efficiency (number of samples per the final model) and obtained accuracy (exploration/exploitation search of the stochastic domain) is if the new sample lies in the projection of samples from the lower domains.

The cross-section condition is necessary for all samples in the higher stochastic domains and it reads: Each sample in the higher stochastic domain must have a sample in the lower stochastic domain with the same coordinate, e.g. for the second order increment function consider a sample [1 1] and the central point [0 0], the considered samples in the lower domains are [1 0] and [0 1]. However, in some cases, a number of samples in the lower stochastic domain is not sufficient for the higher stochastic domain. In that case, it is necessary to perform the search over the whole stochastic domain. Therefore, the search over the domain can be divided into two approaches.

The first approach searches the stochastic domain over the selected sub-domain samples. As mentioned above, the best sample is selected from the projection of samples from the lower domains in the stochastic space. This is done in the following way: Select samples from one level lower sub-domains, e.g. for the increment function $dF_{123}(x_1, x_2, x_3)$, the selected samples belonging to the set of increment functions $dF_{12}(x_1, x_2)$, $dF_{13}(x_1, x_3)$, $dF_{23}(x_2, x_3)$. Tensor product of these vectors represents a set of possible positions for a new sample, i.e.

$$V_{i...j} = V_i \otimes ... \otimes V_j \tag{5.7}$$

where V_i represents a vector of samples from the lower stochastic domain. However, this set of samples includes samples, which have one of the coordinates equal to the central point. These samples would not influence the given stochastic domain and therefore, they have to be neglected from the set.

Now, it is easy to process the set of samples, $V_{i...j}$, through MD-PEC (Eq. (5.6)). Sample with the maximum value represents a new position for sampling of the expensive function. This can be mathematically written in the following way

$$\mathbf{x}^k_{new} = \underset{\mathbf{v} \in V_{i...j}}{\arg\max} \quad E^k_{ND}(\mathbf{v}) \tag{5.8}$$

The second approach represents the search over the whole stochastic domain. The search starts, when the set V_i is empty and there are no more unused samples. In other words, all samples from set V_i are already used in the interpolation process. This is done in the following way

$$\begin{aligned} \mathbf{x}^k_{new} &= \underset{\mathbf{x} \in \mathbf{D}}{\arg\max} \quad E^k_{ND}(\mathbf{x}) \\ \text{subject to:} & \qquad \mathbf{x} \neq {}^c\mathbf{x} \end{aligned} \tag{5.9}$$

where $\mathbf{D}$ represents the optimization domain, which is bounded by the input distributions (see Sec. 3.3). The application of the optimization algorithm is not straightforward. The MD-PEC function represents a highly problematic function and direct application of the optimization technique to the problem of interest would lead to a dramatic computational burden. To overcome this problem, a numerical approach is used, which is later introduced in this chapter.

The optimization algorithm is used when the higher order increment functions have more complicated shape than the low order increment functions. In theory, this happens in rare cases, where the partial derivatives have higher influence than the lower ones (see Sec. 2.2). However, in practice, it can happen that the lower order increment functions are converged closely to the allowed threshold. In that case, the high dimensional surrogate model requires additional samples to fulfil the accuracy condition. Note that the optimization process is run on the surrogate models created for the given increment function. This ensures that the computational burden is kept low.

In theory, the optimization process is straightforward. In practice, the analytic form of the proposed algorithm is not well suited for a direct application and therefore, the numerical approach is adopted. The numerical approach is discussed in the following section.

5.2 Numerical application of the high dimensional adaptive algorithm

The application of the adaptive scheme is not straightforward as it needs to take into account the underlying surrogate model, the nature of the problem and also, the convergence criteria. Therefore, in this section, the numerical approach for the high dimensional adaptive scheme is introduced.

In the first step, let us recall notation for the proposed approach. Let $SdF^k_{i...j}(x_i, ..., x_j)$ be the surrogate model of the increment function, $dF_{i...k}(x_i, ..., x_k)$, using $n_{SdF} + k_{SdF}$ samples in iteration k. Let $V_{i...j}$ be a set of possible positions for a new sample, which is construed as a tensor product of all samples from the considered lower stochastic domains, e.g. set $V_{1\ 2\ 3}$ consider samples from sets $V_{1\ 2}$, $V_{1\ 3}$ and $V_{2\ 3}$. The set $V_{i...j}$ consider combination of all samples and it is necessary to neglect all non-influential samples. This includes all the samples with the following attribute:

$$x_t = {}^c x_t \qquad t = i \cdots j \tag{5.10}$$

In the next step, the adaptive scheme separates into two approaches. The first approach consider a non-used samples in set $V_{i...j}$, i.e. some points in set $V_{i...j}$ still does not have a value from the expensive model. Therefore, we can perform the optimization process over samples which holds the cross-section condition. Now, we can assume a vector of inputs $x_{i...j}$, where each input is a sample from set $V_{i...j}$, i.e.

$$\mathbf{x}_{i...j} \in V_{i...j} \tag{5.11}$$

The second approach considers an empty set $V_{i...j}$. In other words, all the samples are used in the interpolation process and it is necessary to perform the search over the entire domain. In this case, one can run the expensive optimization (Eq. (5.9)) algorithm, however, the MD-PEC function is highly non-linear and it would create computation problems. Therefore, the same approach is taken as in the case of 1-D adaptive sampling. The first step is to run the LHS design (9) in the domain of interest. Note that the LHS design is used and not the stratified sampling design (85), which takes into account underlying probabilities. The influence of the input probability is introduced later in the process. As in the first approach, it is necessary to discard all possibilities with the null influence, i.e. application of Cond. 5.10. Next, assume vector $\mathbf{x}_{i...j}$, where each input is a sample from the LHS design mentioned earlier, i.e.

$$\mathbf{x}_{i...j} \in LHS_{i...j} \tag{5.12}$$

where $LHS_{i...j}$ is LHS on the domain $i...j$. When the vector of possible positions of the next sample, $\mathbf{x}_{i...j}$, is defined, the process of selection can start. First, the normalization of all samples in vector $\mathbf{x}_{i...j}$ is done using Eq. (5.4). The normalization is also, applied on the surrogate model $SdF^k_{i...j}(x_i, ..., x_j)$. Next step is to evaluate MDEC function (Eq. (5.1)) for all samples in vector $\mathbf{x}_{i...j}$. The derivatives in the MDEC function can be obtained directly from the surrogate model, such as in a case of Lagrange interpolation, or using the central finite difference scheme.

The last part represents the application of the input probability into decision criteria. Let $\mathbf{rx}^{ND}_{i...j}$ be the joint probability density of random vectors $\mathbf{rx}_i$, which is defined in Sec. 3.3. This is mathematically written in the following way

$$\mathbf{rx}^{ND}_{i...j} = \prod_{h=i}^{j} \mathbf{rx}_h \tag{5.13}$$

Using the joint probability density, it is necessary to construct histogram $h^p_{i...j}$. However, the histogram, $h^p_{i...j}$, is created with a number of bins equal to the number of samples in vector $\mathbf{x}_{i\ldots j}$. The histogram $h^p_{i\ldots j}$ is normalized to obtain the joint probability distribution, $\tilde{h}^p_{i\ldots j}$. The following MD-PEC function is obtained as follows

$$\mathbf{E}^k_{ND} = \boldsymbol{\epsilon}^k_{ND}\tilde{h}^p_{i\ldots j} \tag{5.14}$$

where $\boldsymbol{\epsilon}^k_{ND}$ represents the vector of values obtained from Eq. (5.1). The last part is to obtain the position of a new sample, which is done in the following way

$$x_{new} = \underset{\mathbf{x}_{i\ldots j}}{\arg\max} \quad \mathbf{E}^k_{ND} \tag{5.15}$$

where $\mathbf{x}_{i\ldots j}$ are samples from set $V_{i\ldots j}$ (Eq. (5.11)) or from the LHS design (Eq. (5.12)).

Once, the position of a new sample is obtained, the process of sample proposition is not finished. In the higher stochastic domains, there are other requirements, which needs to be handled. For example, in a case of Lagrange interpolation, it is necessary to sample the stochastic domain in a grid fashion, otherwise, the interpolant starts to oscillate. Therefore, in each iteration, when a new sample is proposed, it is necessary to add samples to fulfil all the conditions required by the interpolant. To simplify the process, the additional samples are distributed such that the tensor grid is constructed.

The second aspect represents samples in the lower stochastic domains. For all samples, it is necessary to fulfil the cross-section condition (see Sec. 5.1). Therefore, in a case of the second approach, it is necessary to obtain all samples belonging to the lower domains, e.g. increment function $dF_{1\ 2\ 3}$ requires samples from increment functions dF_1, dF_2, dF_3, $dF_{1\ 2}$, $dF_{1\ 3}$ and $dF_{2\ 3}$. There are two ways how to obtain required values. The first way is to obtain the required values from the surrogate model constructed in the lower stochastic domain. However, this approach introduces an error into the surrogate modelling process and when, the increment function is of a very high order ($4 <$), the error could be too large. Moreover, the error is cumulative in nature and this is not desired. The second way is simply to obtain the value from the expensive model and this approach is taken in this work.

5.3 Convergence - The high order increment functions

The convergence process for the higher order increment functions is somehow similar to the 1-D process, yet there are differences, which needs to be taken into account. First, let us recall the whole convergence process, which is divided into two levels. The first level represents the convergence of the final model and the second level represents the convergence of the increment function order, e.g. second, third etc... The convergence of the final model is done in the prediction algorithm (see Sec. 4). Therefore, in this part, the focus is given to the convergence of the increment function order.

The convergence process for given order can be separated into two steps. The first step represents the local convergence and the second step represents the global convergence. The high order local convergence focus only on given increment function, while the high order global convergence focus on the current order of the increment functions, i.e. the second order increment functions, the third order increment functions etc... First, the higher order local convergence is discussed.

5.3.1 Convergence - The higher order local convergence

The algorithm is aimed for the uncertainty propagation and therefore, the observed criteria is the mean value and the variance. In Sec. 2.2 is defined the partial mean and the partial variance, which read:

$$\mu^k_{i...j} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} SdF^k_{i...j}(\mathbf{x}_{i...j}) p_{i...j}(\mathbf{x}_{i...j}) dx_i \cdots dx_j \tag{5.16}$$

$$(\sigma^k_{i...j})^2 = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} (SdF^k_{i...j}(\mathbf{x}_{i...j}) - \mu^k_{i...j})^2 p_{i...j}(\mathbf{x}_{i...j}) dx_i \cdots dx_j \tag{5.17}$$

where $SdF^k_{i...j}(\mathbf{x}_{i...j})$ represents the surrogate model in k-th iteration and $p_{i...j}(\mathbf{x}_{i...j})$ represents the joint probability distribution function for a given increment function. The increment function $dF_{i...j}(\mathbf{x}_{i...j})$ is replaced with $SdF_{i...j}(\mathbf{x}_{i...j})$ to emphasize the fact that a surrogate model is used to estimate the statistical properties. The partial mean and the partial variance represent the statistical properties of each increment function. However, to estimate the statistical properties, the numerical approach is used and Eq. (5.16) and Eq. (5.17) are replaced with the numerical equivalent, i.e.

$$\mu^k_{i\ldots j} = \frac{1}{z} \sum_{\mathbf{rx}_s \in \mathbf{rx}} SdF^k_{i\ldots j}(\mathbf{rx}_s) \tag{5.18}$$

$$(\sigma^k_{i\ldots j})^2 = \frac{1}{z-1} \sum_{\mathbf{rx}_s \in \mathbf{rx}} (SdF^k_{i\ldots j}(\mathbf{rx}_s) - \mu^k_{i\ldots j})^2 \tag{5.19}$$

where z represents a number of samples in vector $\mathbf{rx}$, i.e. the vector of random inputs (Eq. (5.13)). In the same way, numerical approach can be used to estimate the total mean (Eq. (2.33)) and the total variance (Eq. (2.34)). Equations for the total mean and the total variance read:

$$\mu^k = f(^{\mathbf{c}}\mathbf{x}) + \sum_{t=1}^{s_D} \mu^k_t \tag{5.20}$$

$$(\sigma^k)^2 = \frac{1}{z-1} \sum_{\mathbf{rx}_s \in \mathbf{rx}} \left(\sum_{t=1}^{s_D} SdF^k_t(\mathbf{rx}_s) - \mu \right)^2 \tag{5.21}$$

where s_D represents a number of the selected increment functions and z represents the number of samples in vector $\mathbf{rx}$, i.e. the vector of random inputs. The selected increment functions, s_D, differs from the 1-D approach. In the 1-D case, the statistical properties (Eq. (3.19) and Eq. (3.20)) are computed from all the 1st order increment functions, where for the N-D case, only the lower order increment functions are considered. For example, if the tested increment function is the third order, then the considered increment functions are the first and the second order plus the considered third order increment function. This step is important as one cannot calculate the total mean and the total variance from the not-converged increment functions. Using all the non-converged increment functions in Eq. (5.20) and Eq. (5.21) would lead to a premature convergence and therefore, to the non-accurate model.

Once the total mean and the total variance are established, one can compute the residual between iterations. The residual of the total mean and the total variance are estimated in the following way

$$R^k_{\mu_{i\ldots j}} = \left| \frac{\mu^k_{i\ldots j} - \mu^{k-1}_{i\ldots j}}{\mu^k} \right| \tag{5.22}$$

$$R^k_{(\sigma_{i\ldots j})^2} = \left| \frac{(\sigma^k_{i\ldots j})^2 - (\sigma^{k-1}_{i\ldots j})^2}{(\sigma^k)^2} \right| \tag{5.23}$$

where k represents an iteration of the process. Note that the total variance and the total mean are observed simultaneously. This is important aspect as both values have to be fully converged to ensure the accuracy of the uncertainty propagation.

Observing only the residuals is not enough. The total variance and the total mean are dependent on all involved surrogate models and if one of the surrogate models is converging slowly, the total number of samples is not optimal. Same as in the 1-D approach, one more convergence condition is applied - the logistic convergence. The logistic convergence is defined as follows

$$LR^k_{\mu_{i\ldots j}} = \left| \frac{\mu^k_{i\ldots j} - \mu^{k-1}_{i\ldots j}}{\mu^k_{i\ldots j}} \right| \tag{5.24}$$

$$LR^k_{(\sigma_{i\ldots j})^2} = \left| \frac{(\sigma^k_{i\ldots j})^2 - (\sigma^{k-1}_{i\ldots j})^2}{(\sigma^k_{i\ldots j})^2} \right| \tag{5.25}$$

Same as in the 1-D approach, the normal and logistic convergence condition holds the following condition: if all involved surrogate techniques are converged, the normal and the logistic convergence holds the following condition

$$LR^k_{\mu_{i\ldots j}} \geq R^k_{\mu_{i\ldots j}}$$

$$LR^k_{(\sigma_{i\ldots j})^2} \geq R^k_{(\sigma_{i\ldots j})^2}$$

To prove this claim, let us focus on the convergence of the expected value (Eq. (5.22) and Eq. (5.24)). First, substitute Eq. (5.22) and Eq. (5.24) into the above condition:

$$\left| \frac{\mu^k_{i\ldots j} - \mu^{k-1}_{i\ldots j}}{\mu^k} \right| \geq \left| \frac{\mu^k_{i\ldots j} - \mu^{k-1}_{i\ldots j}}{\mu^k_{i\ldots j}} \right|$$

Simplifying the above inequality leads to the following inequality:

$$\mu^k \geq \mu^k_{i\ldots j}$$

It is clear that μ^k will be larger than $\mu^k_{i...j}$, because μ^k is composed of $\mu^k_{i...j}$. Therefore, from the inequality, it is clear that the proposed condition has to hold.

On the other hand, if the condition is not valid [1], one of the surrogate models is divergent and it needs more samples in given stochastic domain. In order to decrease the necessary number of samples, the local convergence is established in the following way

$$\begin{array}{c} if \\ LR^k_{\mu_{i...j}} < R^{\mu}_{set_{i...j}} \quad or \quad R^k_{\mu_{i...j}} < R^{\mu}_{set_{i...j}} \\ and \\ LR^k_{(\sigma_{i...j})^2} < R^{(\sigma)^2}_{set_{i...j}} \quad or \quad R^k_{(\sigma_{i...j})^2} < R^{(\sigma)^2}_{set_{i...j}} \\ then \rightarrow stop \end{array} \tag{5.26}$$

where R^{μ}_{set} and $R^{(\sigma)^2}_{set}$ represent the set residual for a local convergence process. This residual is set by the user and in this work, it is set to the desired accuracy (see Chap. 3). However, this residual is modified according to needs of the problem. This is discussed later in this chapter.

The proposed approach ensures that each surrogate model is handled according to its influence to the final model, i.e. if the increment function has a significant contribution to the final model, the samples are added. Conversely, the non-significant increment functions are left only with minimum samples. Moreover, the convergence process is stopped if the local surrogate model is locally converged and further sampling would not bring desired improvement. This ensures that the optimal number of samples is used. The whole convergence process is described in Alg. 4:

[1] Consider a case where the expected value is converging to 0 and the partial expected values are converging to a different value. In this case, the final model is saved with the logistic convergence, i.e. the final solution is convergent.

Algorithm 4 Scheme for the high order local convergence - part 1

```
Initialize the process:
1. Obtain $R^{\mu}_{set_{i...j}}$ and $R^{(\sigma)^2}_{set_{i...j}}$ for each increment function in given order
2. Obtain the maximum order of derivatives $d$
3. Obtain the order of iteration $k$
4. Obtain set $MD - ST$ of the selected increment functions for given order
while set $MD - ST$ contains non-converged increment functions do
    for $t = 1$ to the length of set $MD - ST$ do
        1. Construct the surrogate model $SdF^k_t(\mathbf{x})$ for increment function $dF_t$
        2. Construct a tensor product of all samples from one level lower stochastic domain using Eq. (5.7) and erase all used samples
        if $V_{i...j}$ contains samples then
            1. Assign all samples from set $V_{i...j}$ to vector $x_{i...j}$ (Eq. (5.11))
        else
            1. Run $LHS_t$ in given stochastic domain
            2. Assign all samples from $LHS_t$ to vector $x_{i...j}$ (Eq. (5.12))
        end if
        3. Calculate all derivatives up to the selected order $d$ for each sample in vector $x_{i...j}$ using surrogate model $SdF^k_t(\mathbf{x})$
        4. Calculate the position function, $\mathbf{F}_{ND-pos}$, (Eq. (5.2)) for each sample in vector $x_i$
        5. Calculate all derivatives up to the selected order $d$ for each sample in vector $x_{i...j}$ using surrogate model $SdF^{k-1}_t(\mathbf{x})$
        6. Calculate MD-PEC function(Eq. (5.14))
        7. Obtain position of a new sample (Eq. (5.15))
        8. Calculate the partial mean (Eq. (5.18)) and the partial variance (Eq. (5.19)) for increment function $dF_t(\mathbf{x})$
        9. Calculate the total mean (Eq. (5.20)) and the total variance (Eq. (5.21)) for increment function $dF_t(\mathbf{x})$
        10. Calculate the normal (Eq. (5.22) and Eq. (5.21)) and the logistic residuals (Eq. (5.24) and Eq. (5.25)) for the increment function $dF_t(\mathbf{x})$
        if Cond. 5.26 hold then
            1. Consider the increment function, $t$, from set $MD - ST$ as converged
        end if
    end for
```

Algorithm 4 Scheme for the high order local convergence - part 2

```
        1. Fulfil all necessary interpolation conditions for all the non-converged
           increment functions, i.e. add samples to create a tensor grid
        2. Obtain all the proposed samples from the expansive model, $f(\mathbf{x})$
        3. $k = k + 1$
    end while
    Finalize the process:
    1. Store the residuals of each increment function
    2. Store each surrogate model $SdF_t(\mathbf{x})$
    End
```

5.3.2 Convergence - The high order global convergence

Same as in the 1-D local convergence, the high order local convergence ensures the convergence of the increment function. However, it has the same drawbacks as the 1-D approach, i.e. the sum of the local residuals can be greater then the global residual set by the user. This problem can be written in the following way

$$\sum_{t=1}^{S_D} R^{\mu}_{local_t} > GR^{\mu}_{set} \tag{5.27}$$

$$\sum_{t=1}^{S_D} R^{(\sigma)^2}_{local_t} > GR^{(\sigma)^2}_{set} \tag{5.28}$$

where R_{local_t} represents a lower value of LR_t and R_t. Even with a moderate number of random stochastic variables (~ 6), the number of possible interactions is significant. One can quickly realize that the sum of all local residuals can lead to large errors in the final model. Therefore, an approach for a global convergence is introduced.

The global convergence approach is based on observation of the convergence of the total mean and the total variance of the final model, i.e. the total mean and the total variance of all converged increment functions including all the lower orders. The process to check the global convergence starts when the partial surrogate models are locally fully converged under given threshold. Note that the total mean and the total variance are computed at each iteration k and therefore, the convergence rate of the whole process can be established. To compute the global convergence residuals, the following equations are used

$$GR_{\mu} = \left| \frac{\mu^k - \mu^{k-1}}{\mu^k} \right| \tag{5.29}$$

$$GR_{(\sigma)^2} = \left| \frac{(\sigma^k)^2 - (\sigma^{k-1})^2}{(\sigma^k)^2} \right| \tag{5.30}$$

where k represents the iteration. The same residuals are used in 1-D approach, but in this case, the total mean and the total variance are computed from all the increment functions included in the final model, i.e. from the order 1 up to the current order. The global residuals, i.e. GR_{μ} and $GR_{(\sigma)^2}$, represent the error of the final model. One can quickly realize, that if the global residuals are not under the threshold, the final model is not fully converged.

Let us now focus on the aspect of not fully converged model. Same as in the 1-D case, the convergence process is based on the observation of the partial residuals, $R^{\mu}_{local_t}$ and $R^{(\sigma)^2}_{local_t}$. However, the considered residuals are only of given order, i.e. if the currently solved order is 2, then the considered residuals are only for the second order increment functions. These residuals take into account the global influence of a given surrogate model and also, the local accuracy of a given prediction.

In order to introduce the algorithm, let $MD - STC$ be the set of all increment functions for given order in the final model. The final model considers all increment functions, i.e. from the order 1 up to the current order. The residuals considered in set $MD{-}STC$ are the maximum residuals for each surrogate model, i.e.

$$STC_t = max(R^{\mu}_{local_t}, R^{(\sigma)^2}_{local_t}) \tag{5.31}$$

The maximum residual from the whole set is selected and corresponding increment function is selected for the accuracy improve. The local accuracy (used in the Alg. 4) for the selected surrogate model is modified in the following way:

$$R_{set_t} = \frac{R_{set_t}}{2} \tag{5.32}$$

As discussed above, it is necessary to improve the accuracy of the selected surrogate model. However, one does not know a priori how to modify the residual and therefore, it is empirically selected as a half of the requested value.

The process of the selection ends when the sum of the residuals in set $MD-STC$ is smaller then the global residual GR_{set}, i.e. Eq. (5.27) and Eq. (5.28) are no longer valid. The procedure is described in Alg. 5.

Algorithm 5 Scheme for the selection of the higher order increment functions for improvement

```
Initialize the process:
1. Obtain $R^{\mu}_{set_t}$ and $R^{(\sigma)^2}_{set_t}$ for each increment function for given order
2. Obtain $R^{\mu}_{local_t}$ and $R^{(\sigma)^2}_{local_t}$ for each increment function for given order and create set $MD-STC$
3. Obtain $GR^{\mu}_{set}$ and $GR^{(\sigma)^2}_{set}$
4. Create an empty set $MD-ST$
if $GR_{\mu} > GR^{\mu}_{set}$ or $GR_{(\sigma)^2} > GR^{(\sigma)^2}_{set}$ then
    while Eq. (5.27) and Eq. (5.28) hold do
        1. Select the increment function with the highest residual from set $MD-STC$, i.e. $max(MD-STC)$
        2. Modify $R^{\mu}_{set_t}$ and $R^{(\sigma)^2}_{set_t}$ according to Eq. (5.32) and select the corresponding increment function for the accuracy improvement, i.e. increase number of samples in given domain
        3. Erase the increment function from set $MD-STC$ and store this increment function in set $MD-ST$
    end while
end if
Finalize the process:
1. Store set $MD-ST$ for Alg. 4
2. Store residuals $R^{\mu}_{set_t}$ and $R^{(\sigma)^2}_{set_t}$ for Alg. 4
End
```

5.4 Global process for the higher order increment functions and the starting conditions

The global process for the higher order increment functions differs significantly from the global process introduced in Sec. 3.5. The first difference is that the necessary conditions to start the uncertainty propagation are already established. This restricts the propagation of the probability distributions through the higher order increment functions and the adaptive sampling scheme. Moreover, in 1-D case, the considered increment functions are only the first order, wherein, in this case, the considered increment functions are all the higher order increment functions that are of order $\leq$ the analyzed order, e.g. the second increment functions, the third order increment functions etc...

The first step is to obtain the important increment functions for given order. The selection scheme (see Sec. 4) selects the most important increment functions for the given order. For the selected increment functions, the joint probability distributions are constructed accordingly to Eq. 5.13, where the marginal probability distributions are taken from the 1-D case (see Sec. 3.5). The boundaries of the given stochastic space are also, established in the 1-D approach and for the higher order increment functions, it is just a combination of the one dimensional cases. The desired accuracy, GR_{set}, is set the same as for the 1-D case and, in order to simplify the process, it is the same for the expected value and for the variance, i.e. $GR^{\mu} = GR^{(\sigma)^2}$. The last part is to set the maximum derivative order, d, for the high dimensional adaptive algorithm. It is suggested to keep this number low, due to a large computational burden if the number is too high. Therefore, suggested number is $d = 2$ or $d = 3$.

The difference between the 1-D case and the higher order adaptive scheme is the position of the central point and all samples in the lower stochastic domains. In the 1-D case, the position of the central point has to be selected, however, in the higher order scheme, the position of the central point is already given. Moreover, the samples from the lower stochastic domains are given and therefore, collecting the central point and samples from the lower domains (in a case of the second order increment function, it will be sampled on the abscissas) are considered as the zeroth iteration, i.e. $k = 0$. The first iteration ($k = 1$) is to set the samples on the boundaries of the given domain to circumscribe the interpolation domain. The samples in the first iteration do not have to be positioned on the boundaries of the interpolation domain, it depends on the interpolation technique selected. However, an interpolation is always preferred over an extrapolation in this work.

In the 1-D approach, there is at least one sample positioned according to the adaptive algorithm. This ensures that the increment function is not converged prematurely as the first order increment functions are usually the important ones. However, for the higher order increment functions, this step would lead to a large number of samples with a very low added value, i.e. information provided with these samples would be very low. Therefore, in the first iteration, it is checked, if the influence of the surrogate model is higher then the convergence threshold, i.e.

$$\begin{gathered}
if\\
LR^1_{\mu_t} < R^{\mu}_{set_t} \quad or \quad R^1_{\mu_t} < R^{\mu}_{set_t}\\
and\\
LR^1_{(\sigma_t)^2} < R^{(\sigma)^2}_{set_t} \quad or \quad R^1_{(\sigma)^2} < R^{(\sigma)^2}_{set_t}\\
then \rightarrow stop
\end{gathered}$$

The considered increment function is converged if the condition holds and it is not passed to the adaptive algorithm. This leads to savings in the computational time and also, the number of samples is kept low. However, these increment functions are considered in the global convergence process (see Sec. 5.3.2). The global process for the higher order increment functions is described in Alg. 6.

Algorithm 6 The global process for the higher order increment functions - part 1

```
Initialize the process:
1. Obtain all the input probability distributions from Alg. 3
2. Obtain the statistical properties of the final model constructed in Alg. 3
3. Predict the important second order increment functions using the selection scheme (Sec. 4)
4. Assign the important second order increment functions to set $MD-ST$
if $MD-ST$ is not empty then
    while Stopping criteria not met do
        1. Set the global residual $GR_{set}$ for given order of the increment functions
        2. Assign $GR_{set}^{\mu} = GR_{set}$ and $GR_{set}^{(\sigma)^2} = GR_{set}$
        3. Assign for all $R_{set_t}^{\mu} = GR_{set}^{\mu}$ and $R_{set_t}^{(\sigma)^2} = GR_{set}^{(\sigma)^2}$
        for $t = 1$ to the length of set $MD-ST$ do
            1. Construct the joint probability distribution (Eq. 5.13) for the increment function, $dF_t(\mathbf{x})$, from set $MD-ST$
            2. Construct the surrogate model $SdF_t^0(\mathbf{x})$ for the increment function $dF_t(\mathbf{x})$
            3. Sample the boundaries of given increment function, $dF_t$, from set $MD-ST$
            4. Construct the surrogate model, $SdF_t^1(\mathbf{x})$, for increment function $dF_t(\mathbf{x})$
            5. Calculate the partial mean (Eq. (5.18)) and the partial variance (Eq. (5.19)) for each non-converged increment function
            6. Calculate the total mean (Eq. (5.20)) and total variance (Eq. (5.21))
            7. Calculate the normal (Eq. (5.22) and Eq. (5.21)) and the logistic residuals (Eq. (5.24) and Eq. (5.25)) for increment function $dF_t(\mathbf{x})$
            if Cond. 5.26 hold then
                1. Consider the increment function, $t$, from set $MD-ST$ as converged
            end if
        end for
```

Algorithm 6 The global process for the higher order increment functions - part 2

```
            while $GR_\mu > GR_{set}^\mu$ or $GR_{(\sigma)^2} > GR_{set}^{(\sigma)^2}$ do
                1. Call Alg. 4
                2. Call Alg. 5
            end while
            4. Increase the maximum order of the increment functions
            5. Predict the important higher order increment functions using the
            selection scheme (Sec. 4) and update set $MD - ST$
        end while
    end if
    Finalize the process:
    1. Construct the final model accordingly to Eq. (2.28)
    2. Sample the final model with the MC sampling and obtain the statistical
    properties for the problem of interest
    End
```

5.5 Applied examples for the N-D adaptive scheme

To illustrate the high order adaptive approach, two examples are selected. The first example is used for the high order adaptive algorithm, where its main advantages and disadvantages are discussed. The second example represents the application of the global process, i.e. the whole uncertainty propagation is done. The second example represents the common engineering problem and the results of the uncertainty propagation are then discussed.

5.5.1 Applied example using the adaptive algorithm

In order to illustrate the adaptive approach, let us introduce a simple function. The function of interest is following

$$\begin{multline} F_{ND-test}(x_1, x_2) = \\ 2 + 0.25(x_2 - 5x_1^2)^2 + (1 - 5x_1)^2 + 2(2 - 5x_2)^2 + 7\sin(2.5x_1)\ \sin(17.5x_1x_2) \end{multline} \tag{5.33}$$

where x_i represents a random variable with a uniform distribution and boundaries equal to $[0, 1]$. However, in this case, the focus is given on the increment function

$dF_{12}(x_1, x_2)$. Central point is selected to be at position [0.5 0.5]. The increment function is shown in Fig. 5.2 and it reads

$$\begin{aligned} dF_{12}(x_1, x_2) = \\ -6.128 - 0.25(0.5 - 5x_1^2)^2 - 0.25(-1.25 + x_2)^2 + \\ 0.25(-5x_1^2 + x_2)^2 - 7\sin(2.5x_1)\ \sin(8.75x_1) \\ -6.642\sin(8.75x_2) + 7\sin(2.5x_1)\ \sin(17.5x_1x_2) \end{aligned} \tag{5.34}$$

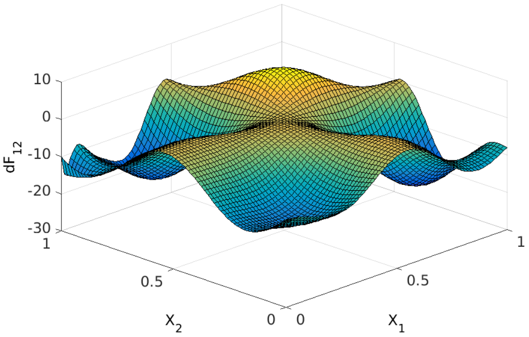
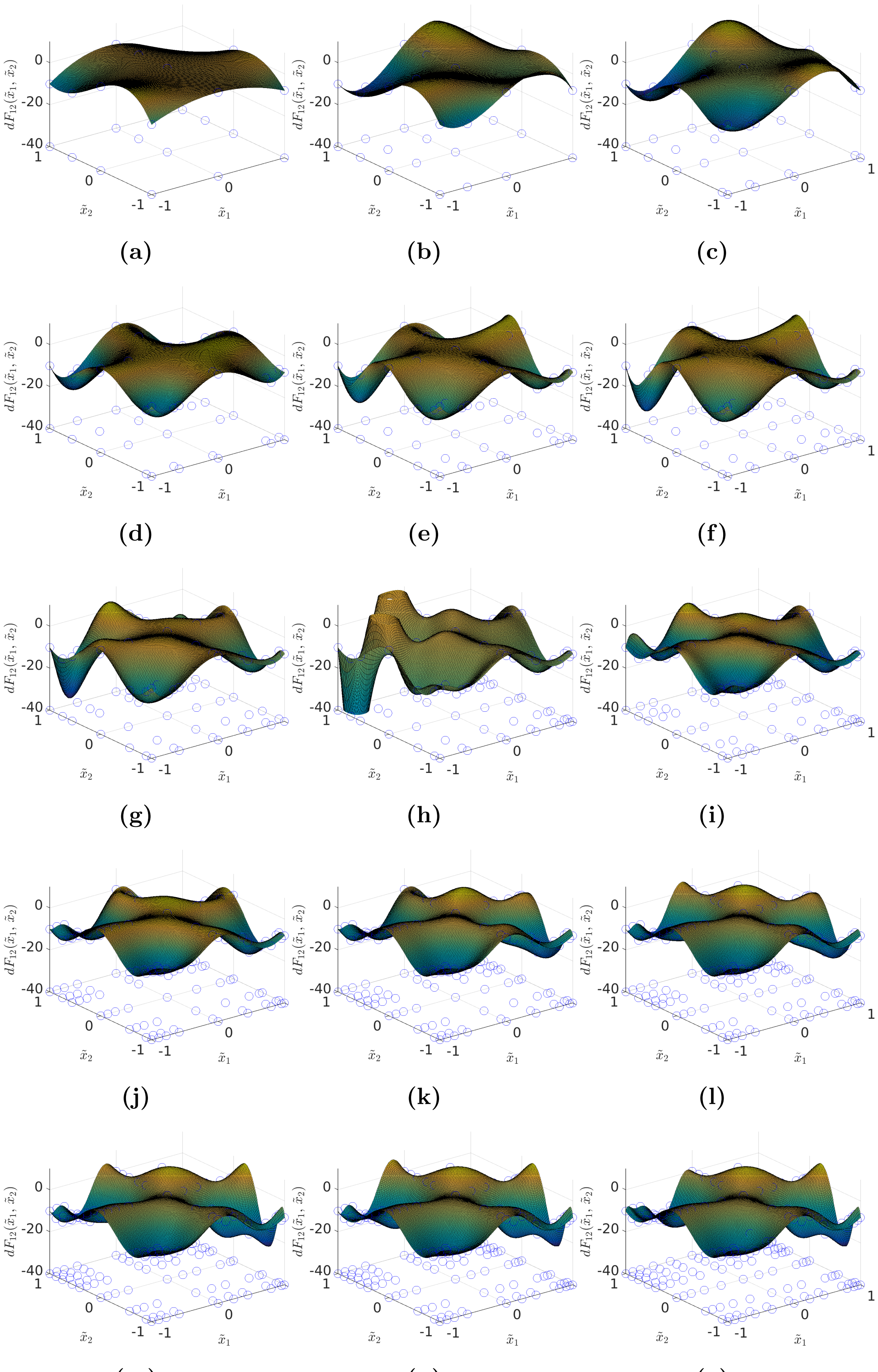


Figure 5.2: Increment function $dF_{12}(x_1, x_2)$

There are several aspects which have to be taken into account in the case of the higher order increment functions. The most common aspect is the natural growth of the increment function from the central point, i.e. the increment is getting stronger, the further we move from the central point. This is a direct consequence of the integration part in the increment function (see Sec. 2.2). However, this aspect is valid only for functions, which can be accurately approximated with a polynomial composed of a low order monomials, e.g. ≤ 4. Therefore, the adaptive scheme should emphasize regions further from the central point in order to improve the accuracy of the surrogate models for the higher order increment functions.

The second aspect is the zeroth value of the increment function around its lower stochastic domains, i.e. all planes and hyper-spaces passing the central point has to be zero. Therefore, samples should be proposed only in the higher domains, i.e. the scheme should avoid sampling the lower stochastic domains. The process of sampling is shown in Fig. 5.3.



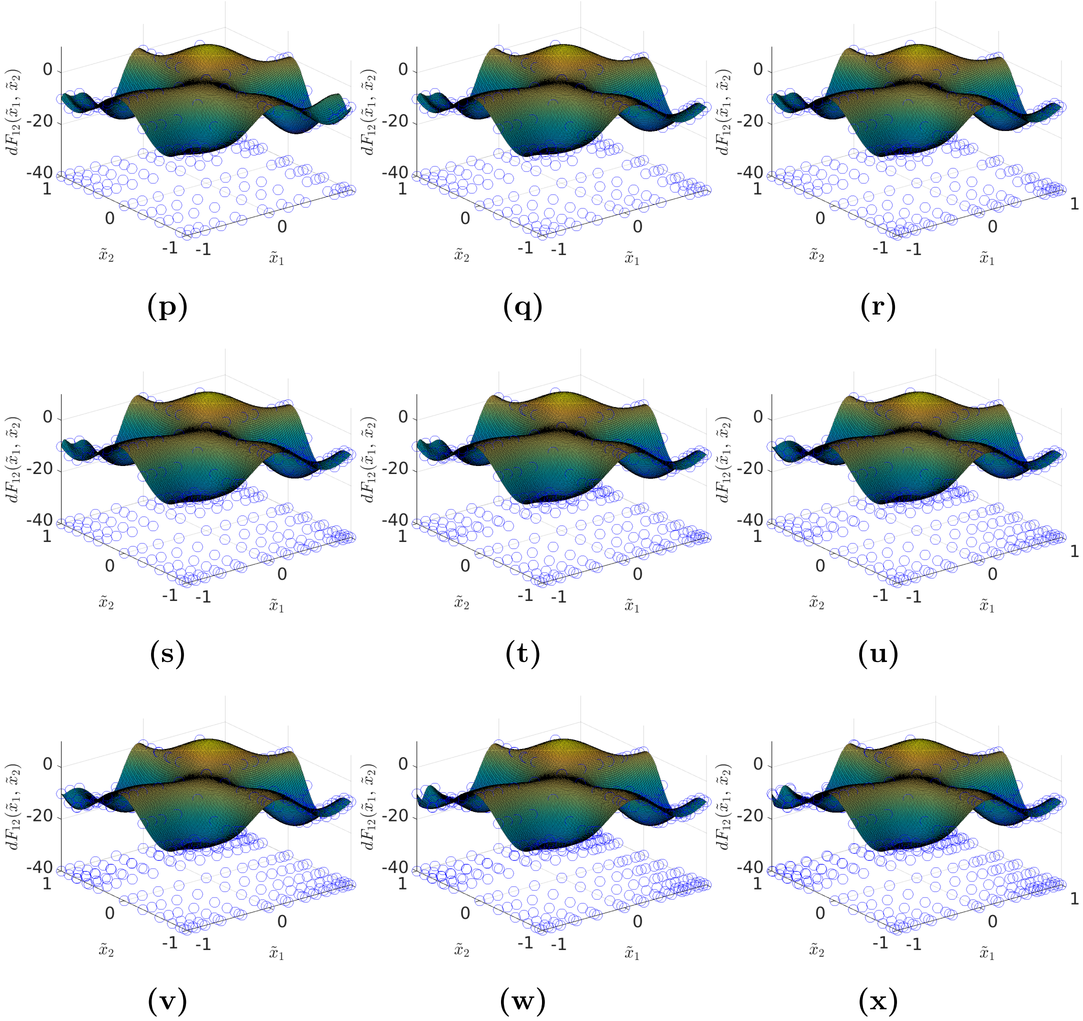
(a) **(b)** **(c)**

(d) **(e)** **(f)**

(g) **(h)** **(i)**

(j) **(k)** **(l)**

(m) **(n)** **(o)**



(p) **(q)** **(r)**

(s) **(t)** **(u)**

(v) **(w)** **(x)**

Figure 5.3: Process of the ND-adaptive sampling for the function of interest $dF_{12}(x_1, x_2)$
Blue circle - Sample from the expensive function

The interpolation technique used for Eq. (5.34) is the multi dimensional Lagrange interpolation (15, 16). Therefore, when the high order local convergence scheme (Alg. 4) proposed a sample, it also samples all the necessary samples to fulfil the tensor condition, i.e. to obtain samples in a grid way. This ensures that Lagrange interpolation does not start to oscillate and provide a reliable interpolation.

The ND-adaptive scheme proposes samples to the positions of the greatest influence. Therefore, samples are mainly spread around the corners of the stochastic domain. The algorithm also takes into account the zeroth influence of the lower stochastic domain and therefore, it does not sample the abscissas or hyperplanes, which belongs to the lower stochastic domains.

Lagrange interpolation has natural tendency to oscillate. The adaptive scheme can recognize it and it proposes samples in the areas of largest oscillations as it can be seen on steps from Fig. 5.3b to Fig. 5.3i. The function has the final trend in Fig. 5.3q and the resulting steps are "polishing" of the function to ensure proper convergence.

The interesting part is the position of samples for the fully converged model. Samples are mainly positioned in the corners of the stochastic domain and this resembles the well known Smolyak Sparse grid approach. In other words, the proposed adaptive scheme behaves similarly to the well known and proven method. On the other hand, this is not something that would be surprising as the HDMR approach using the Lagrange models is closely related to the polynomial chaos approach (see Appendix E). One can suggest using well proven optimal sampling criteria for polynomials such as Clenshaw-Curtis or Gaussian. However, these schemes work well for polynomials, yet for other models such as Kriging, they do not provide optimal results. Later in this work, the multi surrogate approach is introduced, where a combination of various surrogate models is used to approximate the domain of interest. Therefore, we use the adaptive scheme to have technique independent of the surrogate model, i.e. technique which provides optimal sampling scheme regardless the surrogate model used.

The comparison of samples suggested by the adaptive scheme and the Smolyak Sparse Grid is shown in Fig. 5.4.

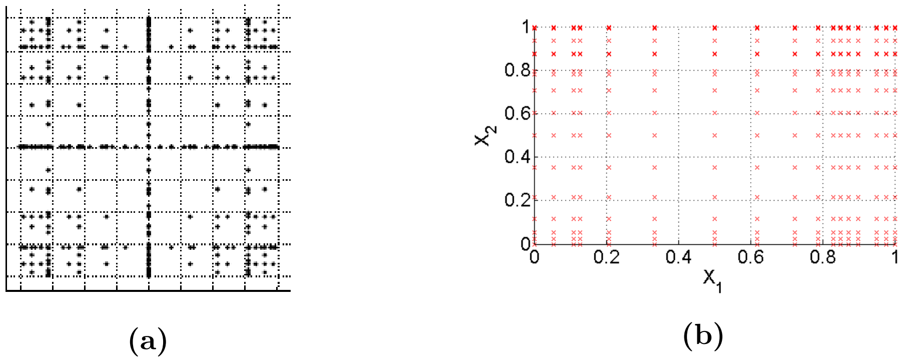


Figure 5.4: Comparison of the Smolyak Sparse grid grid utilizing the Gaussian abscissas (1) (a) and the adaptive scheme (b)

5.5.2 Applied example for the global N-D approach

The function selected for the application of the N-D global process is the well known Borehole problem (86). It represents an 8-D physical problem and the function reads

$$F(x) = \frac{2\pi T_u(H_u - H_l)}{ln\frac{r}{r_w}(1 + \frac{2LT_u}{ln\frac{r}{r_w}r_w^2 K_w} + \frac{T_u}{T_l})} \tag{5.35}$$

where r_w represents the radius of borehole (m), r represents the radius of influence (m), T_u represents the transmissivity of upper aquifer (m2/yr), H_u represents the potentiometric head of upper aquifer (m), T_l represents the transmissivity of lower aquifer (m2/yr), H_l represents the potentiometric head of lower aquifer (m), L represents the length of borehole (m) and K_w represents the hydraulic conductivity of borehole (m/yr). The output is water flow rate in m^3/yr.

Distributions given to each random variable are summarized in Tab. 8.1.

ID	Random Variable	Distribution type	Mean	Standard deviation
x_1	r_w	Normal	0.10	0.0161812
x_2	r	Log-Normal	7.71	1.0056
x_3	T_u	Uniform	63070	115600
x_4	H_u	Uniform	990	1110
x_5	T_l	Uniform	63.1	11.6
x_6	H_l	Uniform	700	820
x_7	L	Uniform	1120	1680
x_8	K_w	Uniform	9855	12045

Table 5.1: Input distributions for the Borehole model

The comparison is made with MC simulation with distributions from Tab. 8.1 sampled directly on the model of interest. The results of MC simulation are summarized in Tab. 5.2. The convergence of the standard deviation and the mean value for various thresholds are summarized in Tab. 5.3. The selected partial mean values and the partial variances for the second case are summarized in Tab. 5.4. The maximum order of derivative is set to $d = 3$.

The histogram obtained via MC sampling is shown in Fig. 5.5. The histograms obtained by the high dimensional adaptive method for various thresholds are summarized in Fig. 5.6. The partial histograms are shown in Fig. 5.8. In

Function calls	Mean	Standard deviation
100000	73.84	28.71

Table 5.2: MC simulation for the Borehole model

Case	Desired accuracy	Function calls	Mean	Standard Deviation	Relative error of Mean	Relative error of S. D.
1	0.01	87	73.85	28.70	1.821e-04	4.736e-4
2	0.001	126	73.84	28.70	2.977e-05	3.198e-4

Table 5.3: Results of the high dimensional adaptive UQ-HDMR approach

Increment function	Partial Mean	Partial Variance	Mean Sensitivity	Variance Sensitivity
r_w	1.8174	531.4032	0.6402	0.6716
H_l	-0.0258	72.1023	0.0090	0.0911
L	0.9143	71.5599	0.3221	0.0904
K_w	0.0030	16.7087	0.0010	0.0211
$r_w.K_w$	0.0045	1.7665	0.0015	0.0022
$r_w.L$	0.0153	7.5374	0.0053	0.0095
$r_w.H_l$	0.0068	7.5901	0.0024	0.0096
$r_w.H_u$	0.0128	7.5983	0.0045	0.0096

Table 5.4: Results of the selected increment functions for the Borehole problem

Fig. 5.7 is showed histogram obtained with only the first order increment functions. Comparing the histogram obtained with the first order increment functions only (Fig. 5.7) and histogram obtained with the full model (Fig. 5.6), one can easily see the larger tail. This proves the conclusions obtained in the Chap. 2.

The comparison graphs of the MC method and the adaptive method are shown in Fig. 5.6.

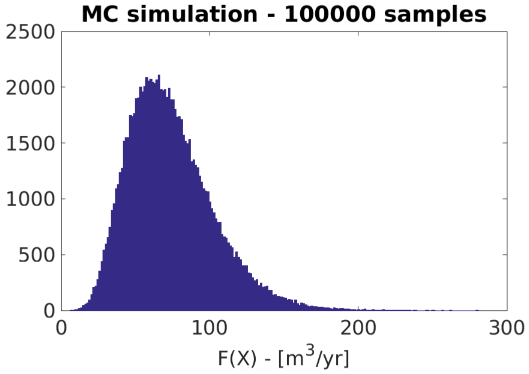


Figure 5.5: PDF obtained by MC simulation

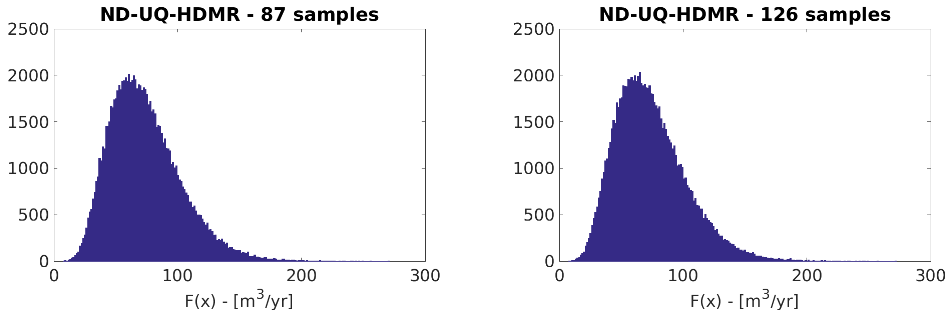


Figure 5.6: PDF obtained by the high dimensional adaptive UQ-HDMR Left: Relative Accuracy 0.01 Right: Relative Accuracy 0.001

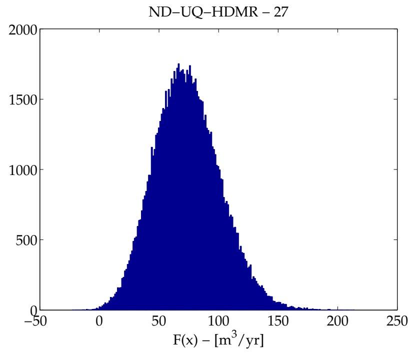
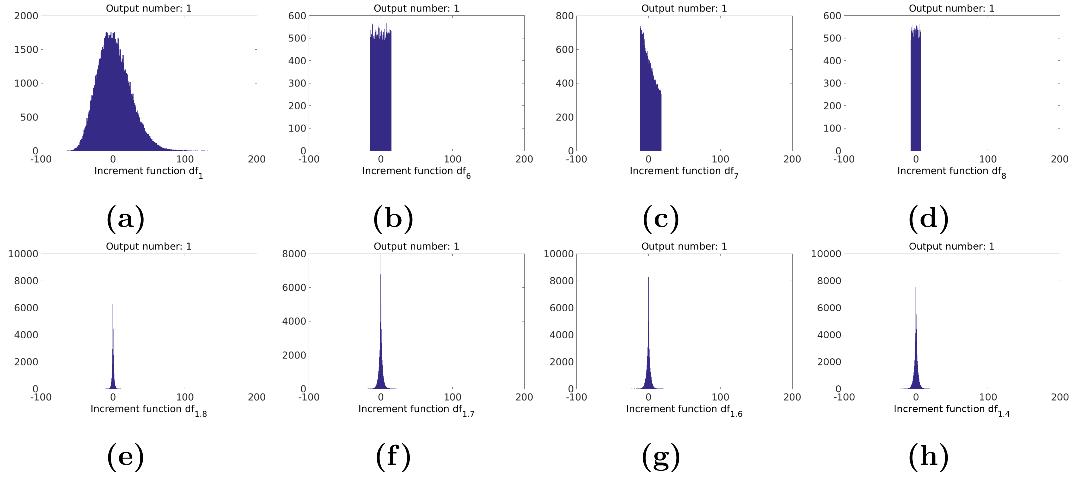


Figure 5.7: PDF obtained with only the $1st$ order increment functions



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 5.8: Histograms for the selected increment functions of the Borehole problem

5.5.3 Discussion about the applied examples

The uncertainty quantification is performed on the Borehole problem. Simplicity and quick evaluation of the Borehole problem make it an ideal test case for engineering applications of the method. The proposed method successfully converged for a low number of samples and provided a very accurate representation of PDF, leading to small errors between MC simulation and the proposed method. The right tail of the final distribution is well described for both residuals (case 1 - $1e-2$, case 2 - $1e-3$). However, a slightly better tail is caught with case 2. This is due to the additional higher order increment functions, namely the 3rd order increment function, dF_{178}. The list of all sampled increment functions is showed in Appendix F.

Variable x_1 is mainly responsible for the final distribution as it can be seen from comparison of Fig. 5.8a and Fig. 5.6. The other variables have a small or negligible effect on the final distribution. The interaction effects represented in Fig. 5.8e - 5.8h are responsible for slight skewness and a longer tail on the right. An interesting comparison is the difference between the converged model (Fig. 5.6) and the model obtained with only the $1st$ order increment functions (Fig. 5.7). One can see that the final model has elongated right tail and therefore, the higher order increment functions are responsible for tails of the final distribution as it is concluded in Sec. 2.2.

One can also notice that the number of samples starts to grow very fast with the higher orders and the additional value of the higher order increment functions is not very large (compared to the value of the first order). Therefore, in practice, one should avoid using a large number of higher order increment functions of not necessary. On the other hand, if the tails of the output distribution are required such as in a case of the reliability study, then the usage of the higher order increment functions is necessary.

In both cases (1 and 2), the method converged to prescribed accuracy, i.e. to the set residual. However, the difference between results (see Fig. 5.6) compared to MC simulation are negligible. Therefore, one can set a philosophical question how the residual should be set.

5.6 Discussion about the high dimensional adaptive scheme

In this chapter, the high-dimensional scheme is presented and applied to the Borehole example. The method extends the 1-D adaptive scheme defined in Chap. 3. However, this approach focus on the interaction effects of given random variables, i.e. the higher order increment functions. The local adaptive scheme focus only on the sampling process of the given increment function, where the global process ensures the selection of the important higher order increment functions. This includes the predictor scheme defined in the previous section.

The high dimensional adaptive algorithm builds on the 1-D adaptive algorithm. However, there are differences between these two. In the 1-D case, the sample selection is free of choice and it is basically an optimization problem. In the higher dimensions, the search is restricted with a requirement of the cross-section condition (see Sec. 5.1). To save the computational time, the projection of samples is considered first. It is assumed that the adaptive algorithm already selected the best position in the lower level and therefore, using the projection of samples leads to the best position in the higher domains. This is a direct consequence of the derivative equation (see Chap. 2). The algorithm switches to a random search over the given domain, when all the lower domain samples are used. However, this is a case of very few problems and very often, it leads to case 2 (see Sec. 2.2), where the higher order derivatives are more complicated than the lower ones. This is usually the case of chemical flows or flows including magnetic field.

The adaptive technique is not restricted to the specific surrogate technique and any surrogate technique can be used. However, this brings an additional step to the sampling process represented by the specific requirement of a selected surrogate model. One does not know a priory what technique is used as a surrogate model and therefore, the adaptive technique cannot be optimal in some cases. In other words, if the surrogate technique requires additional samples in a specific position, the additional samples are not optimally placed. This is a particular case of Lagrange polynomials, where the grid sampling approach is required (for example, see Fig. 5.4b). On the other hand, this can be used as an advantage, when multiple models are introduced in the surrogate process. This is discussed in Chap. 6.

The ND-global process differs from the 1-D global process and the main difference is the usage of the predictor scheme. One quickly realizes that without the predictor scheme, the number of samples required for the higher order increment functions would be extreme. For example, in a case with 5 variables, there are 10 second order interactions. Considering a case, where samples are positioned in corners of given stochastic domain, it means a minimum of 40 additional samples. Many of these samples have a very low influence on the final model and therefore, it would be a large waste of a computational time. The same logic applies for the increment functions in the global algorithm (Alg. 6), before the adaptive scheme is run. This leads to dramatic savings in terms of a computational time, especially in a case of large stochastic spaces, e.g. $n > 10$.

In the most adaptive schemes, the convergence criterion is based on observation of a residual. However, here, the convergence criteria differs from the well known approaches. In this work, the convergence process is separated into three aspects.

The first aspect represents the convergence of the increment function, where residuals of the expected value and the variance are considered. The second aspect is the convergence of the final model for given set of increment functions, i.e. given order of the increment functions. This step prevents from premature convergence and it helps to ensure that the desired accuracy will be met. The third aspect represents the convergence of the order for the increment functions and it is done in the predictor scheme (see Sec. 4). The final model is fully converged, when the predictor scheme neglects all non-important increment functions, i.e. there are no additional increment functions to sample.

5.7 Conclusion

In this chapter, the adaptive sampling approach for the higher order increment functions is introduced. The proposed adaptive sampling is an extension to the 1-D approach, where the EC equation is modified to the N-D case - MDEC. MDEC function takes into account the position of the previous samples, the complexity of the underlying problem and input distributions. However, for the N-D increment functions, it is necessary to respect the requirements of the selected interpolation technique such as grid distribution of samples. This reduces the optimality of proposed samples yet it is a necessary step.

In this chapter is also introduced the convergence process, which is separated into three aspects. The first aspect is the local convergence, where the properties of the increment function are tested. The observed criteria are the statistical properties of given increment function.

The second aspect represents the global convergence, where the convergence of total mean and total variance are checked. The global convergence ensures that the final model is accurate enough and the sum of the local errors does not exceed given threshold. This step is similar to the global convergence process defined in Chap. 3.

In this chapter, the application of the selection scheme to the method is also introduced. The selection scheme represents the third convergence aspect and it is responsible for the final model convergence. This ensures that the number of samples required to construct the final model is kept low. All these aspects make the UQ propagator a very efficient tool.

The proposed technique is shown in two examples. In the first example, an analytic function is considered, where the properties of the adaptive sampling scheme are shown. Using the Lagrange interpolation as a surrogate model, it samples the stochastic domain in a similar way as proven methods such as Smolyak Sparse Grid. This proves the effectiveness of our method.

The second example considers the well known Borehole problem as a test case. In this example, the whole method is tested and showed an excellent agreement with the MC method for a fraction of cost (87 samples). Moreover, the right tail is also very well described. In this example, conclusions obtained in Sec. 2 are proven.

Chapter 6

Multi surrogate modelling

The decomposition process introduced in Chap. 2 allowed to use multiple surrogate models for a single problem. With a proper combination of surrogate models, the global robustness of the developed method can be greatly enhanced. However, it is not known a priori what is the best combination of surrogate models.

In this chapter, the focus is given to the development of the multi surrogate algorithm, which combines various surrogate models to obtain robust interpolant for given increment function. The multi surrogate interpolation technique can be divided into two schemes, where the first scheme measures the overall performance of the selected interpolation technique over the domain and the second scheme measure the local accuracy of the selected interpolation technique. Both schemes measure the efficiency of the surrogate model over time (iterations) and discard the diverging surrogate models. The final model is a combination of the most accurate techniques.

The chapter is structured in the following way: First, the literature review of the multi surrogate techniques is done. In the second section, the basic theory is defined. In the following subsection, the mean weight approach is introduced. Next subsection is given to the local improvement approach, where measurement of the local accuracy is defined. The next subsection explains the application of the local improvement to the overall process. In the third section, the application of the overall interpolation technique is discussed. In the fourth section, the multi surrogate interpolation technique is tested on various test functions. These functions represent various cases, which can be encountered during uncertainty quantification. In the sixth section, a discussion about results and conclusions is given. The last part is given to the final conclusion of this chapter.

6.1 Literature review

In many engineering problems, the usage of surrogate models provided a successful approach, where these models replaced time consuming and expensive computer simulations. Moreover, in many engineering areas such as optimization, sensitivity analysis or uncertainty quantification, the surrogate models represent the only viable solution to obtain an accurate result. However, these models represent a simplification of the real model and therefore, they represent a source of errors.

One of the reliable approximation techniques is the Polynomial Chaos (87, 88, 89, 90, 91), which is also an efficient surrogate model. Another polynomial interpolation scheme: Multi dimensional Lagrange Interpolation (92, 93) is widely used. However, these approaches represent only a small portion of surrogate models used for engineering problems. Interesting work on surrogate models are given in works (49, 50, 52, 57, 94, 95, 96). Review of surrogate models used for computer experiments is given in Chen et al. (97). They provide a general strategy for multiple applications such as electrical engineering or mechanical engineering. In the work of Gorissen et al. (98), surrogate models with the same base (surrogate models using the same basis, e.g. Kriging, Radial Basis Functions (RBF)) are compared on real cases. This provides useful tips for a good surrogate model selection. In the work of Shan and Wang (99), a survey of surrogate models and its applications is given. It also provides some interesting questions, which still remain unsolved.

Many researchers focus on the usage of multiple surrogate models for the uncertainty propagation, in order to obtain a robust and an efficient approach. An interesting application of the surrogate model ensemble is shown in the works of Zhang et al. (100, 101), where a combination of Extended-RBF, RBF and Kriging is done. However, they consider all the models in the process and in the case of a divergent model, the whole process can lead to a catastrophic failure. Another multi surrogate approach is the Weight Average approach (102, 103), where surrogate models are weighted according to their performance. According to the work of Boshop (103), using multiple models can reduce variance. However, in practice, the errors are correlated and the error reduction is not achieved. This was observed in the work of Viana (104), which also provides a rigorous and practical insight into multi surrogate modelling. This work also provides various methodologies for selection of weights. Using multiple surrogates and Dempster-Shafer theory is combined in the work of Muller and Piche (105). In this approach, Dempster-Shafer theory is used to select a combination of the best surrogate approaches. Another approach using an ensemble of surrogate models is described in the work of Ferreira and Serpa (95), where a least square approach is used for a proper combination of surrogate models. A combination of multiple surrogates and global optimization algorithm with an adaptive sampling is given in the work of Viana et al. (106). The Efficient Global Optimization (EGO) is modified to work with a set of surrogates models, making it more efficient optimization approach. In the work of Zhou et al. (107), the evolutionary algorithm is used to combine interpolation and approximation techniques. Moreover, a combination of surrogate models can create a beneficial effect on the final result, i.e. using blessing and curse of uncertainty in the surrogate modelling (for the explanation of a blessing and a curse of uncertainty on surrogate modelling see Zhou et al. (107)).

In previous years, the main focus was given to a single surrogate approach and not so many articles were given to the multi surrogate approach. However, a multi surrogate approach can be an excellent way to have an efficient and a robust surrogate model. Therefore, this part is given to a development of a new surrogate approach, which can combine multiple surrogate models. The important aspect is the neglection approach as in theory, the surrogate models automatically converge. However, in practice, surrogate models suffer from the numerical errors and slow convergence. Therefore, the multi surrogate approach can overcome these problems and it can make the interpolation process more robust and capable of efficiently handle a large number of real problems.

6.2 Idea and theory of multi surrogate modelling

First, let us discuss the basic idea behind the multi surrogate modelling. This problem can be tackled in many ways, where one way is to use ideas from the project management. In the beginning, the project has a target which in our case is the uncertainty propagation. The project manager, i.e. the algorithm, has no knowledge about the problem. Therefore, the project manager hires somebody who knows what to do, i.e. UQ propagation technique. However, the first question which needs to be answered is: who to hire.

In the beginning, the project manager hires only a few workers to get on the right track and understand what the problem is about. In our case, the workers represent surrogate models. However, when the project goes further, the project manager hires more workers or removes some of them, i.e. adding or deleting surrogate models from the problem of interest.

The hiring process is relatively simple. The project manager reads a curriculum vitae and decides who to hire. In the case of surrogate modelling, curriculum vitae represents the properties of the surrogate model, which are obtained from various applications of given surrogate model and also, from the user's experience.

The removal aspect is an important part as some of the workers can perform a bad job. The same is for surrogate models, where some models can diverge and therefore, it is important to delete them from the interpolation process. However, the removal aspect represents a challenging part because the algorithm must know about the performance of the surrogate model, i.e. how efficient is given the surrogate model. This requires time and an effective comparison scheme.

The comparison scheme is a problematic thing as a worker can be below average; however, in a certain area of the project, the worker can be a unique contribution. The same applies to the surrogate models, where for example Kriging model is better than PChip in the overall performance and PChip works better around discontinuity. This makes things difficult and it needs to be taken into account.

Another aspect represents a worker's wage as he/she can perform better than the other workers. The efficient workers have a larger influence on the project and therefore, they must have better wage than the average ones. One can easily see a similarity in surrogate modelling. Some surrogate models perform better than the other ones and this has to be reflected in their influence.

Let us now convert the above idea into a mathematical approach. The project represents the final surrogate model, which can successfully approximate a given problem. As said above, the worker is represented by a surrogate model and the time is represented by an iteration in the sampling process. We are looking for a final surrogate model composed of other surrogate models, which is robust and also efficient, i.e. use as minimum expensive samples as possible and is accurate on a whole range of problems. The mathematical form of the final surrogate model, $sf(\mathbf{x})$, is defined in the following way

$$sf(\mathbf{x}) = \mu(\mathbf{x}) \tag{6.1}$$

where $\mu(\mathbf{x})$ represents the weighted mean of all the considered models and it reads

$$\mu(\mathbf{x}) = \sum_{i=1}^{N_m} wm_i sf_i(\mathbf{x}) \tag{6.2}$$

where wm_i represents the normalized weight coefficient for a given surrogate model, N_m represents the number of selected surrogate models and $sf_i(\mathbf{x})$ represents the surrogate model. This is the core of the multi-surrogate approach. One can understand that the payment for the surrogate model (worker) is represented by a weight coefficient (wage). However, the wm_i coefficient is constant over the entire stochastic domain. The coefficient wm_i represents the average accuracy of the given surrogate model, i.e. the average performance of the worker and it is further explained in the following sections.

As mentioned before, one worker can perform well in a certain range, i.e. one interpolation technique can be very accurate over some parts of the stochastic domain. This needs to be taken into account and the final surrogate model has to be modified. Therefore, Eq. (6.1) is modified in the following way

$$sf(\mathbf{x}) = \mu(\mathbf{x}) + \sum_{i=1}^{N_m} w_{f_i}(\mathbf{x})(sf_i(\mathbf{x}) - \mu(\mathbf{x})) \tag{6.3}$$

where $w_{f_i}(\mathbf{x})$ represents the correction of the model over a certain range of the stochastic domain, i.e. the local weight function. One can understand the local weight as a reward for given work and sum of the corrections represents the local advantages of given surrogate models over the entire stochastic domain. The local weight function is further explained in the following sections.

This is the basic multi surrogate model. In the next subsections, the aim is given to the definition of the weighted mean value approach and to the local improvement.

6.2.1 Weighted mean value approach

In the weighted mean value approach, the basic idea is that the sum of various models gives better prediction over one model. However, only the accurate models have to be used and the inaccurate models have to be neglected from the final model. The selection process is based on an assumption of inaccuracies in given surrogate models.

To illustrate the basic idea of the weighted mean value approach, let us consider a sample in the stochastic domain. Around given sample, a probability distribution can be created to estimate the probability of passing surrogate models. In other words, none of the surrogate models is going to exactly predict the position of the sample, i.e. each surrogate model has an error of interpolation. If multiple surrogate models are considered, one can create a probability distribution as shown in Fig. 6.1. The probability distribution represents how accurately are surrogate models distributed around a given point. In other words, what is the probability that given surrogate model is at a particular position.

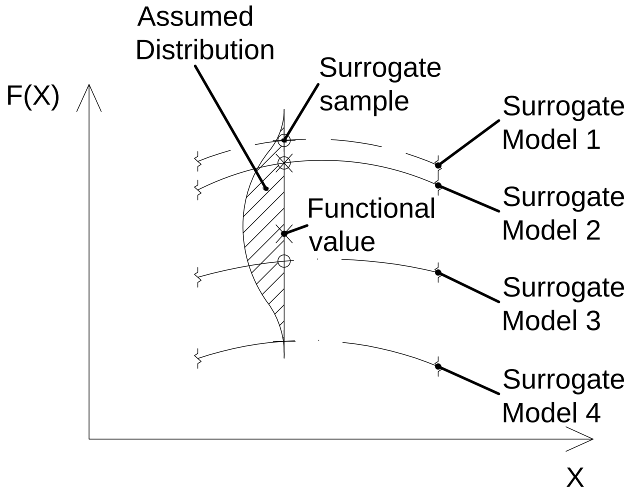


Figure 6.1: Basic idea for the multi surrogate approach

Based on the distribution around given sample, the weights for selected surrogate models can be created. In other words, one can estimate a difference between surrogate model and the test sample and based on these differences, the weights can be created. Moreover, if some surrogate models are on the tails of given distribution, these models can be neglect. However, to assess the validity of given surrogate model, it needs to be done over a number of iterations.

Let us now focus on Eq. (6.2), which represents the weighted mean of the selected surrogate models in a given iteration. For each model considered in a given iteration, the weighted coefficient wm_i is defined in the following way

$$wm_i = \frac{1}{K}\sum_{j=1}^{K}(wmns_j^i) \tag{6.4}$$

where K represents the number of samples in given iteration and $wmns_j^i$ represents the normalized weight of sample j in the current iteration for selected model i. This coefficient is defined in the following way

$$wmns_j^i = \frac{wms_j^i}{\sum_{j=1}^{K}(wms_j^i)} \tag{6.5}$$

where wms_j^i represents the non-normalized weighted coefficient for sample j in a given iteration for selected model i. The normalization process ensures that the sum of weighted coefficients is always 1, which is necessary condition to obtain an interpolation technique. The non-normalized weighted coefficient is obtained using the weight function, which reads

$$wms_j^i = e^{\left(\frac{-\mid y_j^i \mid}{\mid \sigma_{MS} \mid}\right)} \tag{6.6}$$

where y_j^i represents the difference between the function value for sample j and surrogate model i at the previous iteration, i.e.

$$y_j^i = f(\mathbf{x}_j) - sf^i(\mathbf{x}_j) \tag{6.7}$$

where $f(\mathbf{x}_j)$ is the function value for sample j and $sf^i(\mathbf{x}_j)$ is the surrogate value for sample j. Scaling coefficient σ_{MS} in Eq. (6.6) is defined in the following way

$$\begin{aligned}
\sigma_{MS-p1} &= \frac{1}{K}\sum_{j=1}^{K}\frac{1}{N_m}\sum_{i=1}^{N_m}\mid y_j^i \mid \\
\sigma_{MS-p2} &= \frac{1}{K}\sum_{j=1}^{K}\mid f(\mathbf{x}_j) \mid \\
& if\ \sigma_{MS-p2} < \sigma_{MS-p1} \\
\sigma_{MS} &= \sigma_{MS-p2} \\
& else \\
\sigma_{MS} &= \sigma_{MS-p1} \\
& end
\end{aligned} \tag{6.8}$$

where N_m represents the number of active surrogate models, σ_{MS-p1} represents the standard scaling coefficient and σ_{MS-p2} represents the modified scaling coefficient. The standard scaling coefficient σ_{MS-p1} scales the weight function, which makes the weight function independent of the magnitude of the results. In other words, the scaling assures that the errors of the surrogate model are relatively compared to other surrogate models. However, the standard scaling coefficient, σ_{MS-p1}, is prone to the numerical errors. If one of the models is diverging, the standard scaling coefficient becomes too high and slows the neglection process (described later). In that case, the modified scaling coefficient σ_{MS-p2} is used. This allows fast rejection of diverging models.

Let us have a closer look on the weight function (Eq. (6.6)). An example of this function is showed in Fig. 6.2, where it can be seen that it resembles a Gaussian distribution. This reflects the idea of the probability distribution defined earlier in this section. This type of weight function is not the only one, which can be used. Various types of weight functions can be applied; however, there are some restrictions. The weight function has to be equal to 1 if $y_j^i = 0$ and must be monotonically decreasing. Moreover, it has to go to infinity at both sides. In this work, we restrict ourselves only to the proposed function (Eq. (6.6)).

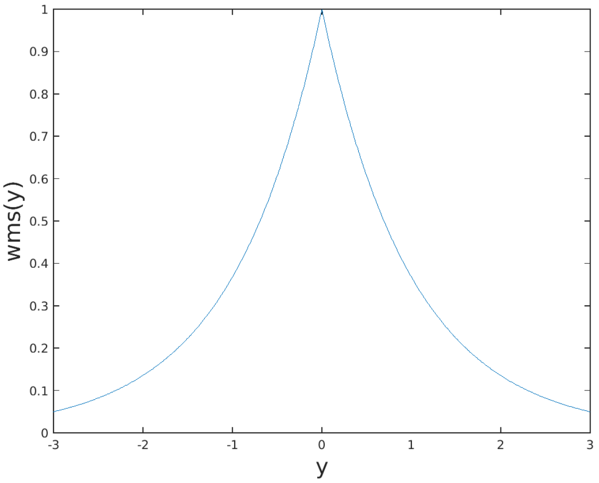


Figure 6.2: Example of the non-normalized weighted coefficient function (Eq. (6.6)) using $\sigma_{MS} = 1$

The models used in the interpolation process are evolving over time, i.e. iterations. However, the performance of each surrogate model can vary over iterations. Therefore, it is necessary to observe the model over time and based on its behaviour during iterations, modify the weight (Eq. (6.4)) assigned in given iteration.

First, let us define a time weight function, which reads

$$Twf(x) = e^{(-|x|)} \tag{6.9}$$

This function is monotonically decreasing and weights from the previous iteration do not have such a high influence as those in the current iteration. It is necessary to obtain weights from the previous and current iteration. In the zero-th iteration, the weights are pre-defined by the user. The modified weights are defined in the following way

$$wmn_i = \frac{\sum_{t=0}^{T_{MS}} Twf(t) wm_i^t}{\sum_{t=0}^{T} Twf(t)} \tag{6.10}$$

where T_{MS} represents the current iteration[1] and wm_i^t represents the weight coefficient defined in Eq. (6.4). The upper script t represents an iteration, i.e. wm_i^t represents the weight coefficient in iteration t for surrogate model i. Using Eq. (6.10), the modified weight coefficient for current iteration is established. The last step is to normalize obtained coefficients, which is done in the following way

$$wm_i = \frac{wmn_i}{\sum_{i=1}^{N_m} wmn_i} \tag{6.11}$$

where wmn_i represents the modified coefficient defined in Eq. (6.10). To explain the process a bit further, consider the third iteration, where 3 samples are added in each step. In each previous iteration, the weighted coefficient is obtained with Eq. (6.11) for each model and for current iteration the weight coefficient is obtained with Eq. (6.4). Note that the sum in Eq. (6.11) is done over all selected (active) model. To explain it a little bit further, consider a following list of available models $[1, 2, 3, 4, 5]$ and a list of active models $[1, 2]$. Therefore, the sum is done over models $[1, 2]$.

Let us now focus on the selection process. One of the main advantages of the proposed approach is the possibility to use only selected models. Moreover, the approach allows adding models later in the interpolation process. The selection process is based on experience or knowledge of the problem. For example, one

[1] Iteration T_{MS} differs from iteration k defined in the previous sections. In other words, the adaptive algorithm is independent from the multi surrogate process. To explain the process further, consider a problem, where the adaptive scheme processed 3 iteration and the multi surrogate approach only one. In this case, $k = k + 3$ and $T = T + 1$.

does know that Lagrange polynomials work better for simpler functions, i.e. lower order polynomials. Therefore, the user selects only Lagrange polynomial for the first few iterations. After few iterations, it is obvious that the function is not simple and therefore, the user adds the Piecewise Cubic Hermite Interpolating Polynomial (PChip) interpolation technique, which is more suited for discontinues functions. In the process of convergence, the algorithm selects, which model performs better and divide the weights accordingly. One can suggest that the best way is to consider all the models from the beginning. However, this is not an optimal approach as multiple models represent slower convergence process. This is discussed in Sec. 6.4.1.

When multiple surrogate models are considered, the algorithm selects, which interpolation technique is the best for given problem. In a case of diverging or completely inappropriate surrogate model, the algorithm neglects the diverging model from the portfolio of considered surrogate models. This represents another advantage of the proposed technique. The advantages of various interpolation techniques are well described in (49, 57, 95, 108, 109). Also, later in this work, properties of various interpolation models are discussed.

The neglection process is defined in the following way. First, the neglection threshold is defined, which reads

$$\epsilon_{Threshold} = \frac{1}{\epsilon_T}\frac{1}{N_m} \tag{6.12}$$

where N_m represents the number of selected surrogate models and ϵ_T is the empirical coefficient, which is set to $\epsilon_T = 2.1$ in this work. Based on the proposed threshold, one can establish the following condition

$$\begin{aligned} &if\ \epsilon_{Threshold} > wm_i \\ &then \implies \textit{neglect the i-th surrogate model} \\ &\textit{from the database} \end{aligned} \tag{6.13}$$

However, it is necessary to apply above condition in an iterative manner. To explain it further, consider a model database that consists of a large number of active surrogate models. The difference between the models in numerical sense would be too large and Cond. 6.13 would neglect a large portion of the surrogate models in one step. Unfortunately, this aspect is problematic even, when $\frac{1}{N_m}$ is introduced in Eq. (6.12). Therefore, after each neglection, the normalization equation (Eq. (6.11)) is applied and the neglection process is repeated.

When the weight is lower than the threshold for a given surrogate model, the global weight, wm_i, is set to 0 for all future iterations. One important aspect should be also noted, when the surrogate model is neglected from the weighted mean approach, the surrogate model is not considered at all. The proposed approach is described in Alg. 7.

Algorithm 7 Scheme for the weight mean coefficients

```
Initialize the process:
1. Obtain iteration $T$
2. Obtain function values ($f(\mathbf{x})$) for all samples in given iteration $T$
3. Obtain the weight values from all previous iterations
4. Obtain a set of active surrogate models
for $i = 1$ to the selected number of surrogate models $N_m$ do
    1. Obtain surrogate values $sf_i(\mathbf{x})$ for all samples in given iteration $T$
    2. Obtain surrogate difference (Eq. (6.7)) for all samples in given iteration $T$
end for
5. Construct the scaling coefficient (Eq. (6.8))
for $i = 1$ to the selected number of surrogate models, $N_m$ do
    1. Construct the non-normalized weights (Eq. (6.6)) for all samples in iteration $T$ using model $i$
end for
6. Normalize the non-normalized weight mean coefficients (Eq. (6.5))
for $i = 1$ to the selected number of surrogate models, $N_m$ do
    1. Calculate the time corrected non-normalized weight coefficient for model $i$ (Eq. (6.10))
end for
8. Normalize the non-normalized weight mean coefficients (Eq. (6.11))
9. Calculate the neglection threshold (Eq. (6.12))
while $\exists\ wm_i\ .\ \epsilon_{Threshold} > wm_i$ do
    1. Discard the surrogate model with the lowest weight mean coefficient, $wm_i$
    2. Normalize the non-normalized weight mean coefficients (Eq. (6.11)) for remaining surrogate models
end while
Finalize the process:
1. Store the mean weight coefficients
2. Update the database of active surrogate models
End
```

6.2.2 Local improvement approach

Once the weighted means are established, one can focus on the local improvements. Let us now extend the idea defined in Sec. 6.2.1, where ensemble of surrogate models create a distribution in a given cross section (see Fig. 6.1). One can assume that the surrogate model will have a similar behaviour in a close neighbourhood around a given cross section. Therefore, the surrogate model with the best approximation can be emphasized in a given neighbourhood. This approach reflects the idea of a worker performing better in a certain part of the project.

Let us now focus on the local improvement in Eq. (6.3), which is represented by

$$w_{f_i}(\mathbf{x})(sf_i(\mathbf{x}) - \mu(\mathbf{x}))$$

The local weight coefficient, $w_{f_i}(\mathbf{x})$, is a function, which is constrained between 0 and 1. However, to have correct final interpolant, it is necessary to establish the following condition: sum of all local weight coefficients is constrained between 0 and 1, i.e.

$$0 \leq \sum_{i=1}^{N_m} w_{f_i}(\mathbf{x}) \leq 1$$

where N_m represents the number of selected surrogate models.

In order to obtain the local weight coefficients, it is necessary to start the process with obtaining the weighted mean model (Eq. (6.1)) from the previous iteration and establishing its value at given sample. Also, it is necessary to obtain functional value for each selected surrogate model in the previous iteration and create a vector of surrogate model values for a given sample, i.e.

$$\mathbf{yl}_j = \left[\mu_j^{T-1}, sf_1^{T-1}(\mathbf{x}_j), ..., sf_{N_m}^{T-1}(\mathbf{x}_j)\right] \tag{6.14}$$

where T represents the current iteration, $\mu^{T-1}(\mathbf{x}_j)$ represents the weighted mean model (Eq. (6.1)) from previous iteration, $sf_i^{T-1}(\mathbf{x}_j)$ represents the surrogate model from the previous iteration and $\mathbf{x}_j$ represents sample in the current iteration. Subtracting the function value at a given sample from vector $\mathbf{yl}_j$ gives the estimation of accuracy. This is mathematically written in the following way

$$\mathbf{Rl}_j = \mathbf{yl}_j - f(\mathbf{x}_j) \tag{6.15}$$

where the vector subtraction is element wise. Once, the accuracy of surrogate models is established, one can see if the surrogate models describe the problem well. There are two possible cases. In the first case, sample $f(\mathbf{x}_j)$ lies far away from all the surrogate models, i.e. vector $\mathbf{Rl}_j$ is either completely negative or completely positive. In this case, all the surrogate models are diverging and it is pointless to emphasize one of them. Therefore, all the local weights, $w^j_{f_i}$, are set to zero, i.e.

$$\begin{aligned} w^j_{\mu} &= 0 \\ w^j_{f_i} &= 0 \end{aligned} \qquad i \in 1...N_m \tag{6.16}$$

where w^j_{μ} represents the local weight for the mean model at sample j and $w^j_{f_i}$ represents the local weight for surrogate model i at sample j.

In the second case, the surrogate models describe the problem accurately and therefore, emphasizing one surrogate model can bring a further increase in the desired accuracy. This is reflected in vector $\mathbf{Rl}_j$, where both signs are present, i.e. there are positive and negative accuracies. However, one wants to consider only surrogate models closest to the true value, $f(\mathbf{x}_j)$ and therefore, only two surrogate models are considered. One from the left (negative accuracy) and one for the right (positive accuracy). To select the closest models, first let us separate vector $\mathbf{Rl}_j$ into two separate vectors, where vector $\mathbf{Rl}^L_j$ consist of only negative accuracies and vector $\mathbf{Rl}^R_j$ consist of only positive accuracies.

With vectors $\mathbf{Rl}^L_j$ and $\mathbf{Rl}^R_j$, one can select the closest surrogate models. The closest surrogate models are the ones, which are closest to the zero and their selection can be mathematically written in the following way

$$MRL_j = max\ component(\mathbf{Rl}^L_j) \tag{6.17}$$

$$MRR_j = min\ component(\mathbf{Rl}^R_j) \tag{6.18}$$

In Eq. (6.17) and Eq. (6.18) two aspects are considered. The first aspect is the accuracy of the selected model and the second aspect represents the selected closest model. In other words, Eq. (6.17) and Eq. (6.18) carry two information, one is the closest model and the second one is the accuracy of the selected model.

The last part is definition of the local weights for the selected surrogate models. The local weight taken in this work are defined as a distance ratio from the selected models. In other words, we emphasize the closest surrogate models proportionally. This is done in the following way

$$w^j_{f_{s:1}} = \frac{MRL_j}{MRR_j - MRL_j} \tag{6.19}$$

$$w^j_{f_{s:2}} = \frac{MRR_j}{MRR_j - MRL_j} \tag{6.20}$$

where $w^j_{f_{s:1}}$ represents the local weight for the surrogate model, which is selected with Eq. (6.17) and $w^j_{f_{s:2}}$ represents the local weight for the surrogate model, which is selected with Eq. (6.18). When the closest models are emphasized in given sample j, the remaining models are automatically set to zero, i.e.

$$w^j_{f_i} = 0 \quad \begin{array}{l} i \neq (f_{s:1}, f_{s:2}) \\ i \in 1...N_m \end{array} \tag{6.21}$$

where $f_{s:1}$ represents the selected model in Eq. (6.17) and $f_{s:2}$ represents the selected model in Eq. (6.18). To explain this step more closer, let us consider the following example. Let us have an ensemble of 4 surrogate models ($i = 1...4$), where set $\mathbf{Rl}^L_j$ consists of surrogate model $i = 1$ and set $\mathbf{Rl}^R_j$ consists of surrogate models $i = [\mu^{T-1}, 2, 3, 4]$. Using Eq. (6.17), the selected surrogate models are $f_{s:1} = 1$ and $f_{s:2} = 3$. Once the models are selected, one needs to establish the local weights for the selected models. This is done using Eq. (6.19) and its values are establish to $w^j_{f_1} = 2/3$ and $w^j_{f_3} = 1/3$. The vector of local weight coefficients then reads $\mathbf{w_f}^j = [0, 2/3, 1/3, 0, 0]$. If the weighted mean model is selected as the closest model, the local weight assigned to the weighted mean model is erased and automatically set to zero.

The proposed approach reflects the idea of the ensemble of surrogate models creating a distribution at a given point. However, at the moment, a local weight is established only at particular sample j. One wants to establish weights over the whole stochastic domain and create a local weight function. This is the topic of next section. The proposed approach is summarised in Alg. 8.

Algorithm 8 Scheme for the local weight coefficients

```
Initialize the process:
1. Obtain the functional value at sample $j$ for the weighted mean model from the previous iteration ($\mu^{T-1}(\mathbf{x}_j)$)
2. Obtain function value at sample $j$ for the expensive function ($f(\mathbf{x}_j)$)
3. Obtain a set of active surrogate models from the previous iteration ($sf_i^{T-1}(\mathbf{x})$)
4. Construct the accuracy vector of surrogate models (Eq. (6.15))
if Set $\mathbf{Rl}_j$ (Eq. (6.15)) contain only negative or positive accuracies then
    1. Assign zero to the local weight of the weighted mean model, i.e. $w_\mu^j = 0$
    for $i = 1$ to the selected number of surrogate models, $N_m$ do
        1. Assign zero to the local weight of surrogate model $i$, i.e. $w_{f_i}^j = 0$
    end for
else
    1. Separate set $\mathbf{Rl}_j$ into set $\mathbf{Rl}_j^L$ and $\mathbf{Rl}_j^R$
    2. Obtain the closest surrogate models from set $\mathbf{Rl}_j^L$ (Eq. (6.17))/ Obtain its accuracy (Eq. (6.17))
    3. Obtain the closest surrogate models from set $\mathbf{Rl}_j^R$ (Eq. (6.18))/ Obtain its accuracy (Eq. (6.18))
    4. Obtain a local weight for surrogate model $f_{s:1}$ (Eq. (6.19))
    5. Obtain a local weight for surrogate model $f_{s:2}$ (Eq. (6.20))
    if $f_{s:1} = \mu_j^{T-1}$ then
        1. Assign zero to the weighted mean model, i.e. $w_\mu^j = 0$
    end if
    if $f_{s:2} = \mu_j^{T-1}$ then
        1. Assign zero to the weighted mean model, i.e. $w_\mu^j = 0$
    end if
    for $i = 1$ to the selected number of surrogate models $N_m$ do
        if $i \neq f_{s:1} \vee i \neq f_{s:2}$ then
            1. Assign zero to the local weight of surrogate model $i$, i.e. $w_{f_i}^j = 0$ (Eq. (6.21))
        end if
    end for
    6. Assembly vector $\mathbf{w_f}^j$
end if
Finalize the process:
1. Store the vector of local weight coefficients ($\mathbf{w_f}^j$) for given sample $j$
End
```

6.2.3 Local improvement in the stochastic domain

Once the local weights are established for given samples, one can start to establish the local weights around the whole stochastic domain. In other words, create a local weight function, which is defined over the whole stochastic domain. To start the process, let us now consider a set of samples $\mathbf{J}$, where each sample consists of a set of local weights $\mathbf{w_f}^j$. These samples are spread over the stochastic domain and they are obtained over multiple iterations. One wants to construct a function, which connects all the samples for a given surrogate model. However, it is necessary to hold the condition defined in the beginning of Sec. 6.2.2, which states that the local weight function has to be constrained between 0 and 1. Moreover, the local weight is known at one particular point in the stochastic domain and using the phenomena "regression paradox[1]" (110), one wants to construct a function, which quickly goes to 0, i.e. goes back to the weighted mean model.

For the moment, let us assume only surrogate model i. First thing is the correlation function, which creates a relationship between samples $\mathbf{J}$ for surrogate model i. The correlation function closely resembles the Kriging approach and there are various types of correlation functions such as Gaussian or linear. Various works on this topic are published, however, we use only the linear correlation function, which is easy to implement and it cannot start to oscillate, i.e. it is bounded between 0 and 1 if the training values are bounded between 0 and 1. Moreover, one does not need to run an expensive optimization algorithm to find the correlation parameter. The linear correlation function reads

$$\theta_i(\mathbf{x}, \mathbf{x_c}) = \sqrt{\| \mathbf{x} - \mathbf{x_c} \|^2} \tag{6.22}$$

where $\mathbf{x_c}$ represents the vector determining center of the correlation function. Using Eq. (6.22) one can construct correlation matrix $\mathbf{R}$, which is K_T x K_T matrix, where K_T represents the number of samples in set $\mathbf{J}$. Each entry in the correlation matrix, $\mathbf{R}$, represents correlation between samples from set $\mathbf{J}$ for given surrogate model i. The correlation matrix then reads

$$\mathbf{R}_i = \begin{bmatrix} \theta_i(\mathbf{x}_1, \mathbf{x}_1) & \theta_i(\mathbf{x}_1, \mathbf{x}_2) & \cdots & \theta_i(\mathbf{x}_1, \mathbf{x}_{K_T}) \\ \theta_i(\mathbf{x}_2, \mathbf{x}_1) & \theta_i(\mathbf{x}_2, \mathbf{x}_2) & \cdots & \theta_i(\mathbf{x}_2, \mathbf{x}_{K_T}) \\ \vdots & \vdots & \vdots & \vdots \\ \theta_i(\mathbf{x}_{K_T}, \mathbf{x}_1) & \theta_i(\mathbf{x}_{K_T}, \mathbf{x}_2) & \cdots & \theta_i(\mathbf{x}_{K_T}, \mathbf{x}_{K_T}) \end{bmatrix} \tag{6.23}$$

[1] regression paradox is an empirical observation, which states that if the observation is an extreme, the next observation will be closer to the mean value.

where $\mathbf{x}_j$ represents the position of sample j from set $\mathbf{J}$ for surrogate model i. Once the correlation matrix is defined, one needs to establish the correlation vector, which represents the relationship between sample of interest and all the training samples. The correlation vector reads

$$r_i(\mathbf{x}) = \begin{bmatrix} \theta_i(\mathbf{x}, \mathbf{x}_1) \\ \theta_i(\mathbf{x}, \mathbf{x}_2) \\ \vdots \\ \theta_i(\mathbf{x}, \mathbf{x}_{K_T}) \end{bmatrix} \tag{6.24}$$

Using Eq. (6.23) and Eq. (6.24) one can construct a local weight function. The local weight function reads

$$w_{f_i}(\mathbf{x}) = r_i(\mathbf{x}) \mathbf{R}_i^{-1} \mathbf{w_f}_i \tag{6.25}$$

where $r_i(\mathbf{x})$ represents the correlation vector for surrogate model i (Eq. (6.24)), $\mathbf{w_f}_i$ represents the vector of local weights for surrogate model i and $\mathbf{R}_i^{-1}$ represents the inverse of correlation matrix $\mathbf{R}_i$ (Eq. (6.23)) for surrogate model i. The final model can be constructed using Eq. (6.3) and Eq. (6.25) and it reads

$$sf(\mathbf{x}) = \mu(\mathbf{x}) + \sum_{i=1}^{N_m} r_i(\mathbf{x}) \mathbf{R}_i^{-1} \mathbf{w_f}_i (sf_i(\mathbf{x}) - \mu(\mathbf{x})) \tag{6.26}$$

Eq. (6.26) represents the final multi surrogate approach, where one can quickly realize that Eq. (6.26) can be rearranged into more convenient matrix form. However, we use the provided form due to the practical programming application.

Example of 1-D and 2-D weight function (Eq. (6.25)) around the stochastic domain is given in Fig. 6.3. In the 1-D case, the local weight function is constructed from 5 samples and it emphasizes Lagrange interpolation technique, which is very lightly trusted around $x_1 = 1570$. In other regions of the stochastic domain, the Lagrange interpolation is not more accurate than the other interpolants and therefore, the interpolant is not emphasized.

In the 2-D case, the local weight function is constructed using a tensor product grid with 9 samples. The local weight function is constructed for the multi dimensional Lagrange interpolation and the algorithm trust the interpolation technique around $x_1 = -0.1$ and $x_2 = \pm 1$. However, this time the algorithm largely emphasize Lagrange interpolation as the other models are far less accurate around this region. On the other hand, the accuracy of the interpolation technique in other regions is not outstanding and therefore, the interpolation technique does not receive special interest.

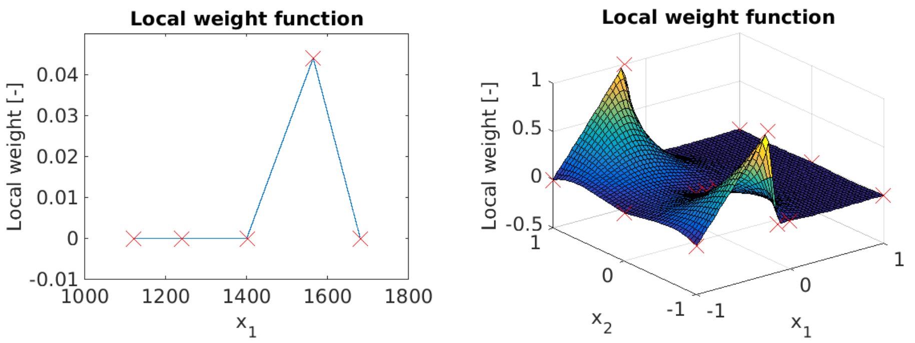


Figure 6.3: Local weight function Left: 1-D case Right: 2-D case

6.3 Application of the multi surrogate approach

The multi surrogate approach can be used as an independent surrogate model for an arbitrary function. However, its application to the increment function is slightly different. One of the main differences between arbitrary function and increment function is the influence of the lower domains, i.e. the increment function is zero at the lower stochastic domains (see Sec. 2.2). Using samples from the lower stochastic domains would slow down the convergence process for the weighted mean approach and therefore, only samples not passing through the central point are considered. To illustrate the problem, let us consider the following increment function: $dF_{i...j}(x_i, ..., x_j)$. The only samples used for the mean weight approach and for the local improvement are the ones, which hold the following condition:

$$x_t \neq {}^c x_t \qquad t = i...j \tag{6.27}$$

where ${}^c x_t$ represents the central point for the stochastic variable t (see Chap. 2).

Another aspect is the adaptive sampling process (see Chap. 3 and Chap. 5). The adaptive scheme sample only one sample in the case of the first order increment functions and few samples in the case of the higher order increment functions in each iteration. However, it was empirically found that using only a few samples for the mean weight approach can lead to a wrong model selection. Therefore, one wants to adapt the weighted mean approach to the selected interpolation techniques, which is done using a simple wait approach. In other words, we wait for several iterations until the mean weight approach is processed. During this waiting period, the local weights are assumed to be 0 and they are reassigned after the mean weights are assigned. The overall structure of the multi surrogate structure is shown in Fig. (6.4).

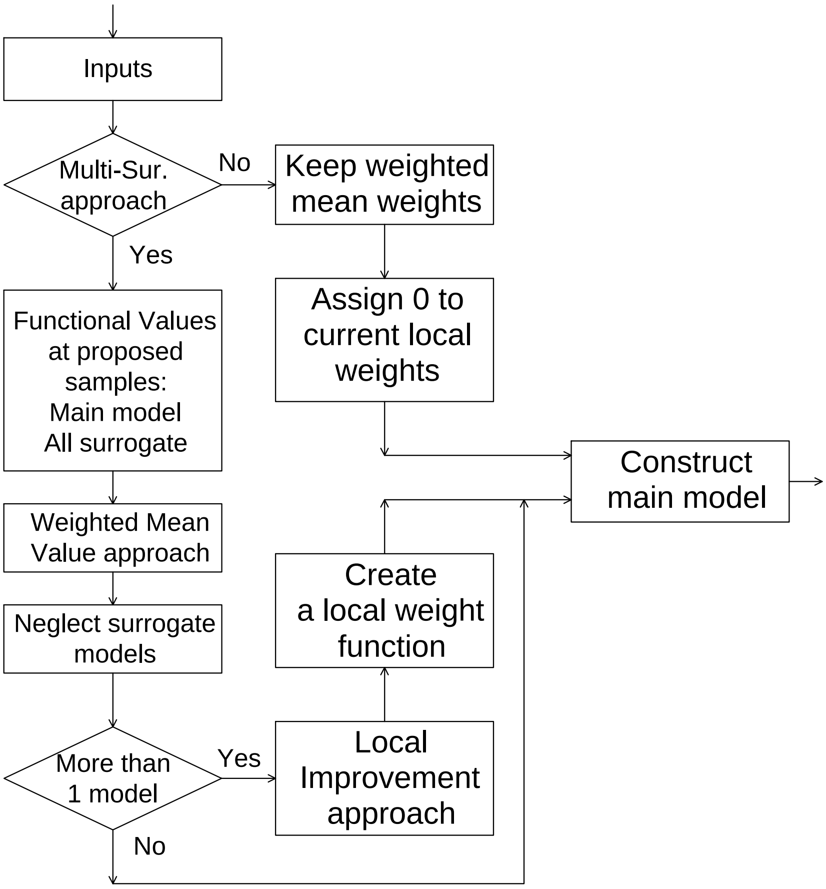

Inputs
Multi-Sur. approach
No
Keep weighted mean weights
Yes
Functional Values at proposed samples: Main model All surrogate
Assign 0 to current local weights
Construct main model
Weighted Mean Value approach
Create a local weight function
Neglect surrogate models
More than 1 model
Yes
Local Improvement approach
No

Figure 6.4: Scheme for the multi surrogate interpolation

6.4 Applied examples for the multi surrogate approach

In order to illustrate the proposed multi surrogate scheme, let us introduce a few simple examples. The first example represents a simple non-linear function, which reads

$$F_{test:1}(x) = (6x-2)^2 \sin(12x-4) \tag{6.28}$$

where x represents the random variable with a uniform distribution and boundaries equal to $[-1,1]$. The same function was introduced in Chap. 3 and it is an interesting example for the multi surrogate approach. In this example, the multi surrogate approach is combined with the 1-D adaptive sampling scheme.

In the first run, we select a large number of surrogate models and all surrogate models start from the first iteration. The list of selected surrogate models is given in Tab. 6.1. The correlation functions for the Radial Basis Functions are showed in Tab. 6.2. Also, in this run, Support Vector Machine (SVM) approximation technique is used. This is not an interpolation technique and it is used for the propose of showing that the proposed technique can work with approximation techniques as well. The input parameters for SVM are shown in Tab. 6.3. Parameters defined in Tab. 6.3 are empirical and their purpose is to avoid a costly optimization algorithm. The convergence of the mean value and the standard deviation for all involved models is given in Fig. 6.5 and Fig. 6.6, respectively. The convergence of the mean model (Eq. 6.1) is also provided, which is given in order to show the influence of the local improvement. Discussion about results is given in the next section.

-	Surrogate model	Mode	Parameters
1	Lagrange Polynomial	-	-
2	Kriging	Gaussian Cor. Fun. / Trend: Pol. 1^{nd}	Opt. par. $\theta = 10$ low b. = 0.1 up b. = 50
3	Kriging	Gaussian Cor. Fun. - Trend: Pol. 2^{nd}	Opt. par. $\theta = 10$ low b. = 0.1 up b. = 50
4	Kriging	Linear Cor. Fun. / Trend: Pol. 1^{nd}	Opt. par. $\theta = 10$ low b. = 0.1 up b. = 50
5	Kriging	Linear Cor. Fun. / Trend: Pol. 2^{nd}	Opt. par. $\theta = 10$ low b. = 0.1 up b. = 50
6	Kriging	Spline Cor. Fun. / Trend: Pol. 1^{nd}	Opt. par. $\theta = 10$ low b. = 0.1 up b. = 50
7	Kriging	Spline Cor. Fun. / Trend: Pol. 2^{nd}	Opt. par. $\theta = 10$ low b. = 0.1 up b. = 50
8	Kriging	Spherical Cor. Fun. / Trend: Pol. 1^{nd}	Opt. par. $\theta = 10$ low b. = 0.1 up b. = 50
9	Kriging	Spherical Cor. Fun. / Trend: Pol. 2^{nd}	Opt. par. $\theta = 10$ low b. = 0.1 up b. = 50

10	Kriging	Gen. Exp. Cor. Fun. / Trend: Pol. 1^{nd}	Opt. par. $\theta = 10$ low b. $= 0.1$ up b. $= 50$
11	Kriging	Gen. Exp. Cor. Fun. / Trend: Pol. 2^{nd}	Opt. par. $\theta = 10$ low b. $= 0.1$ up b. $= 50$
12	PChip	-	-
13	Spline	-	-
14	Radial Basis Function (RBF)	Cor. Fun. 1	-
15	Radial Basis Function (RBF)	Cor. Fun. 2	-
16	Support Vector Machine (SVM)	Gaussian $\lambda = 0.5$	C_{SVM} and ϵ_{SVM}
17	Independent Polynomial Interpolation (IPI)	-	-

Table 6.1: List of surrogate models for Eq. (6.28) in case: 1

Correlation function	-
1	$\sqrt{x^2 + x_c^2}$
2	$e^{\frac{-x^2}{2x_c^2}}$

Table 6.2: List of correlation functions for the Radial Basis Functions

Parameters
$C_{SVM} = 1000max\left(\mid \overline{\mathbf{Y}} + 3\sigma_{\mathbf{Y}} \mid, \mid \overline{\mathbf{Y}} - 3\sigma_{\mathbf{Y}} \mid\right)$
$\epsilon_{SVM} = \frac{1}{100}\frac{\sigma_{\mathbf{Y}}}{\sqrt{K}}$

Table 6.3: Parameters for SVM technique
$\overline{\mathbf{Y}}$ is the mean value of the training set
$\sigma_{\mathbf{Y}}$ is the standard deviation of the training set
K is the number of training samples in the training set

In the second run, the considered function is the same as in the previous case. However, this time the list of provided models is smaller. In this case, the neglect models are the one giving similar behaviour, i.e. from models given similar convergence only one is considered. The list of models is given in Tab. 6.4. The convergence of the mean value and the standard deviation for all the selected models is given in Fig. 6.7 and Fig. 6.8, respectively. The convergence of the

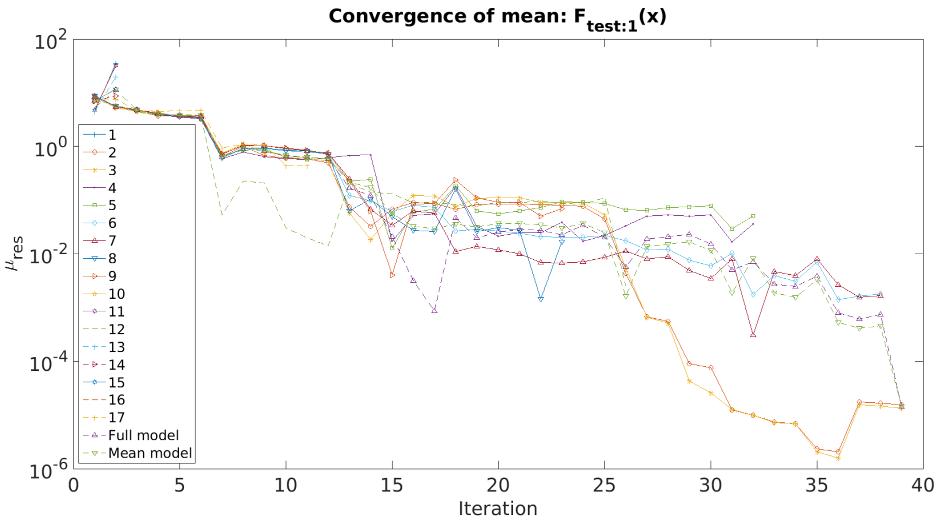


Figure 6.5: Convergence of the mean value for Eq. (6.28)

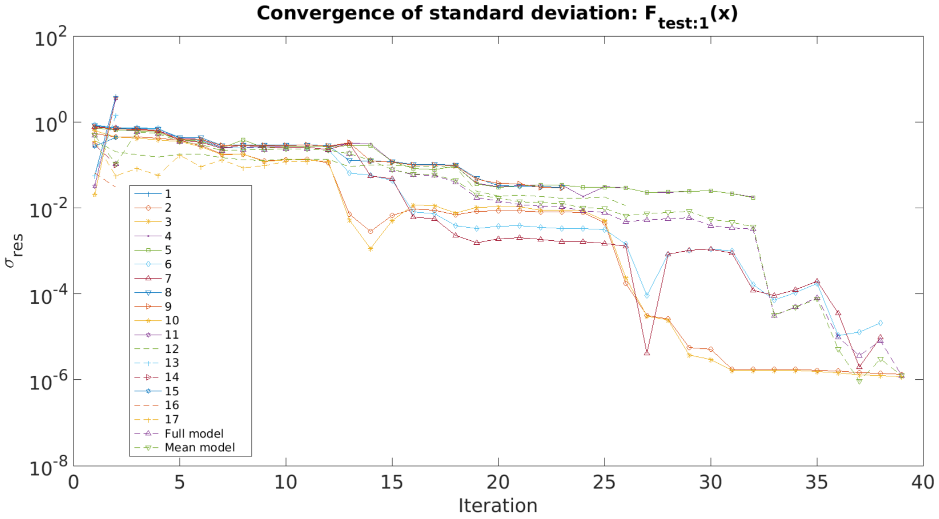


Figure 6.6: Convergence of the standard deviation value for Eq. (6.28)

mean model (Eq. 6.1) is also provided. Discussion about results is given in the next section.

-	Surrogate model	Mode	Parameters
1	Lagrange Polynomial	-	-
2	Kriging	Gaussian Cor. Fun. - Trend: Pol. 2^{nd}	Opt. par. $\theta = 10$ low b. $= 0.1$ up b. $= 50$
3	PChip	-	-

4	Independent Polynomial Interpolation (IPI)	-	-

Table 6.4: List of surrogate models for Eq. (6.28) in case: 2

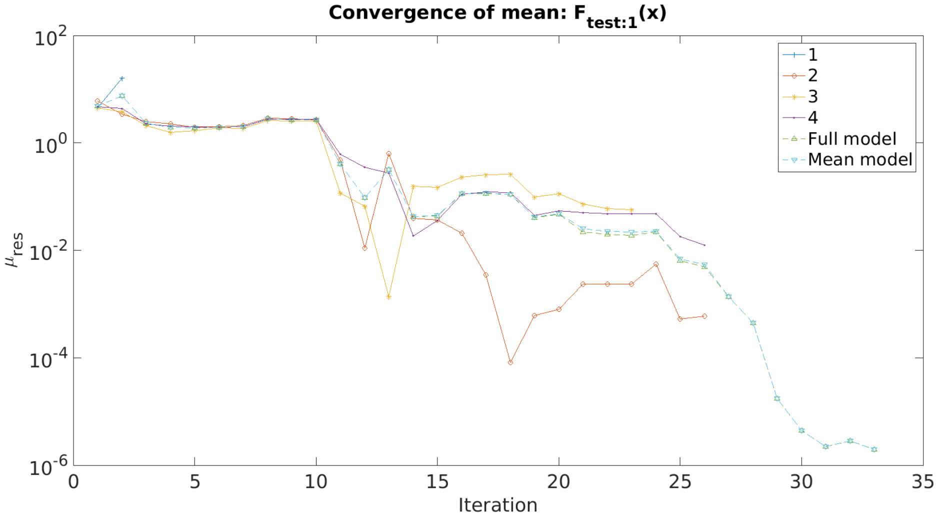


Figure 6.7: Convergence of the mean value for Eq. (6.28)

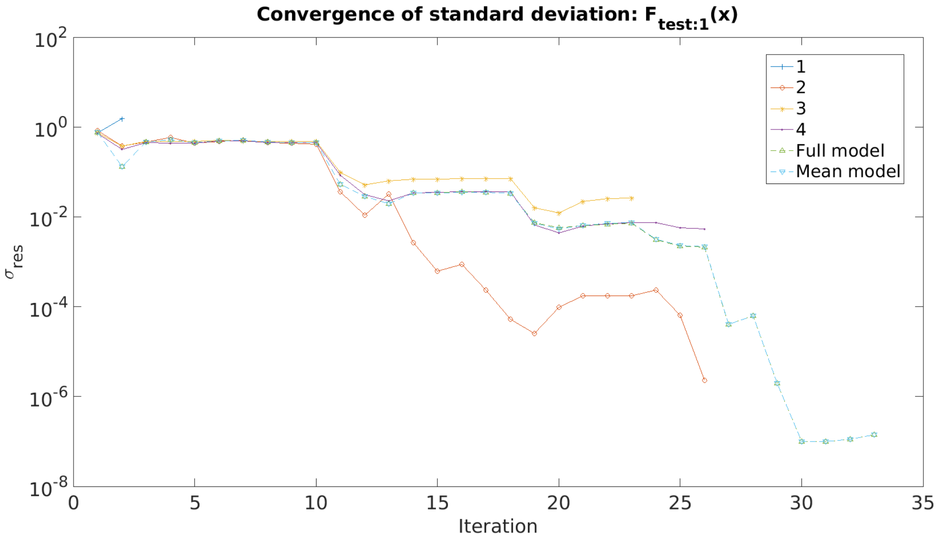


Figure 6.8: Convergence of the standard deviation for Eq. (6.28)

In the third case, a new function is introduced. This function represents a discontinues function and therefore, it represents a serious problem for surrogate models such as Lagrange polynomial or Kriging. The discontinuous function is showed in Fig. 6.9 and its mathematical definition reads

$$F_{test:Jump}(x) = \begin{cases} \sin(x) & x \leq 0.25 \\ \sin(x) + 1 & x > 0.25 \end{cases} \tag{6.29}$$

where x represents the random variable with a uniform distribution and boundaries equal to $[-1, 1]$. In this case, the multi surrogate approach is also combined with the 1-D adaptive scheme and all the surrogate models start from the first iteration. The list of selected surrogate models is given in Tab. 6.5. The convergence of the mean value and the standard deviation for all the selected models is given in Fig. 6.10 and Fig. 6.11, respectively. In this case, the final interpolation is showed in Fig. 6.12, where the final model is showed with samples. Discussion about results is given in the next section.

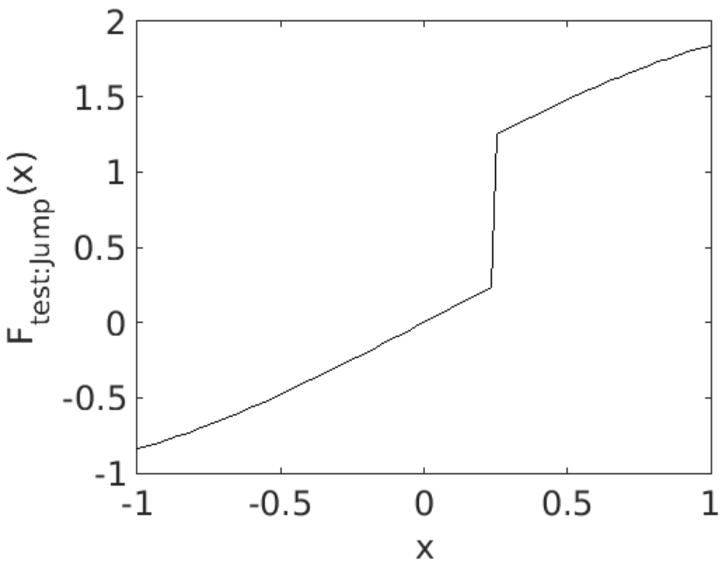


Figure 6.9: Function of interest: $F_{test:Jump}(x)$

-	Surrogate model	Mode	Parameters
1	Lagrange polynomial	-	-
2	Kriging	Gaussian Cor. Fun. - Trend: Pol. 2^{nd}	Opt. par. $\theta = 10$ low b. = 0.1 up b. = 50
3	PChip	-	-

Table 6.5: List of surrogate models for Eq. (6.29) in case: 3

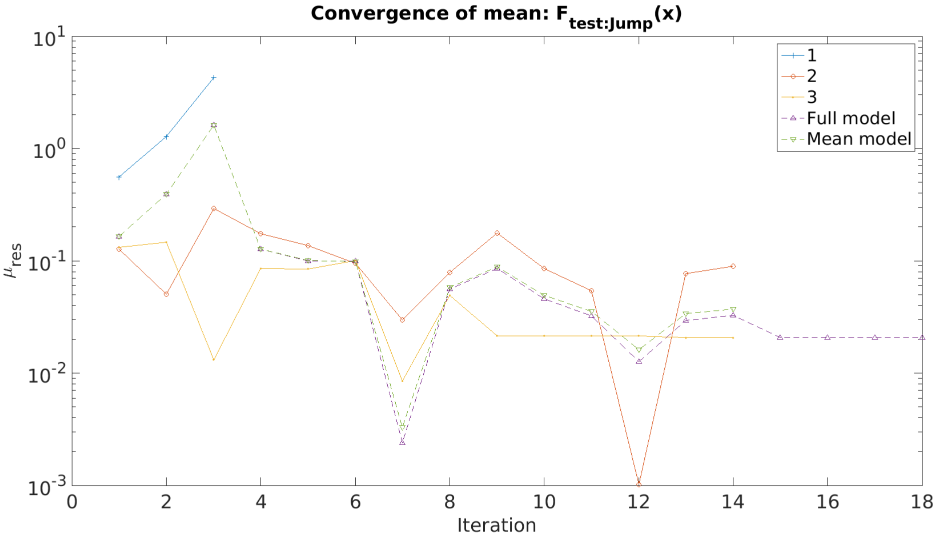


Figure 6.10: Convergence of the mean value for Eq. (6.29)

In the fourth case, a simple function is used, which can be easily interpolated with any interpolation technique. The function is showed in Fig. 6.13 and it reads

$$f(x) = x^3 + x^2 + 30 \tag{6.30}$$

where x represents the random variable with a uniform distribution and boundaries equal to $[-1, 1]$. The list of considered surrogate models is given in Tab. 6.6. Samples are distributed using the 1-D adaptive sampling scheme and all the surrogate models start from the first iteration. The convergence of the mean value and the standard deviation for all the selected models is given in Fig. 6.14 and Fig. 6.15, respectively. The convergence of the mean model (Eq. 6.1) is also provided. Discussion about results is given in the next section.

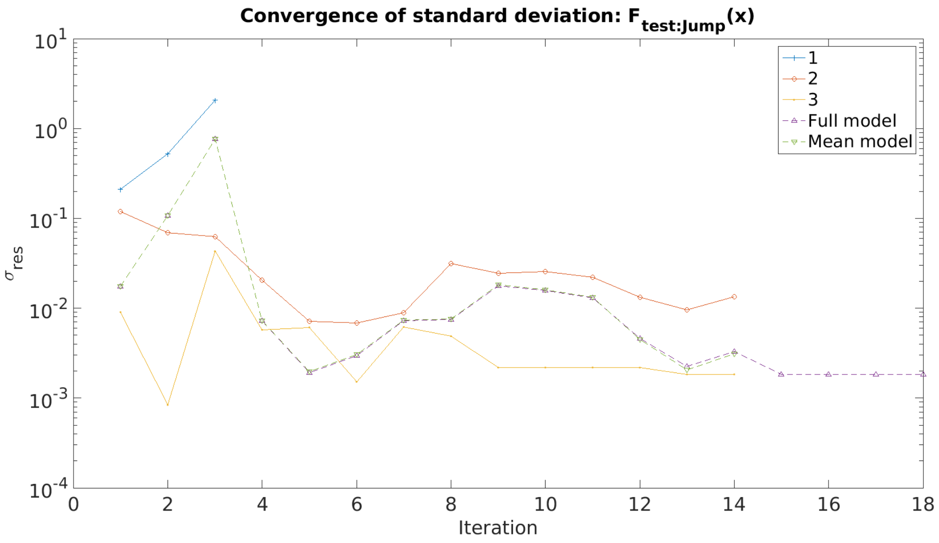


Figure 6.11: Convergence of the standard deviation for Eq. (6.29)

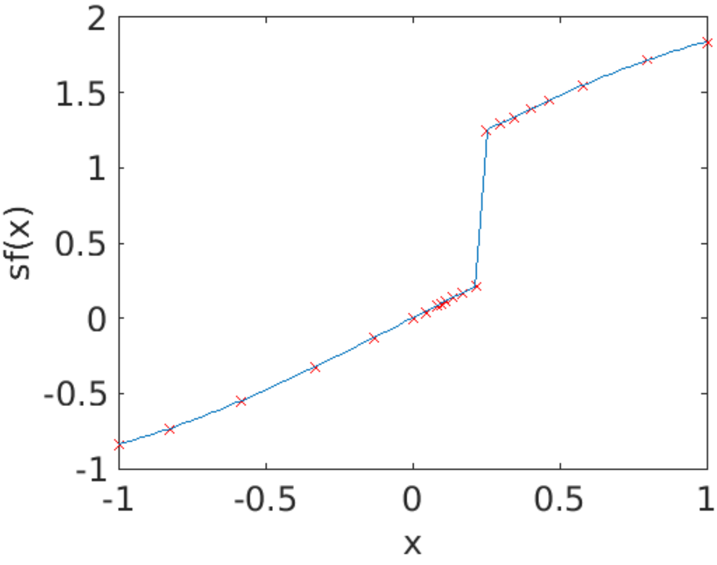


Figure 6.12: The final surrogate model for Eq. (6.29)

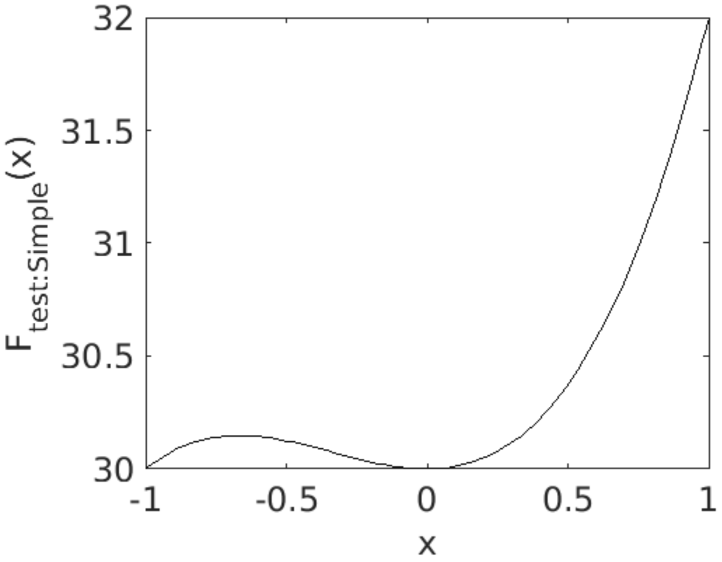


Figure 6.13: Function of interest $F_{test:Simple}(x)$

-	Surrogate model	Mode	Parameters
1	Lagrange Polynomial	-	-
2	Kriging	Gaussian Cor. Fun. - Trend: Pol. 2^{nd}	Opt. par. $\theta = 10$ low b. $= 0.1$ up b. $= 50$
3	PChip	-	-

Table 6.6: List of surrogate models for Eq. (6.30) in case: 4

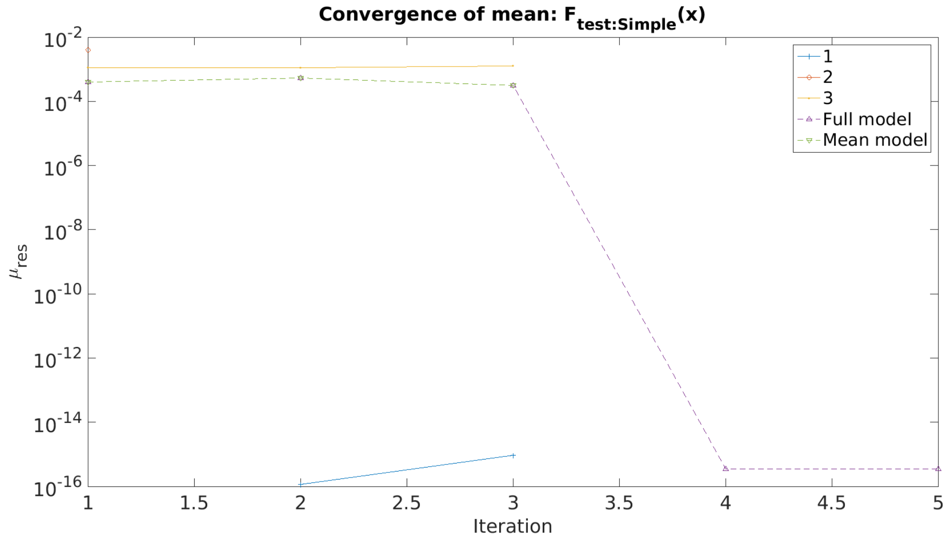


Figure 6.14: Convergence of the mean value for Eq. (6.30)

Note: In the third iteration, the Lagrange polynomial (Model 1) is selected the only model and the full model is composed only of the Lagrange surrogate model. Therefore, the Lagrange polynomial (Model 1) is not showed after the third iteration.

In the fifth case, the previous function (Eq. (6.30)) and the same list of surrogate models are considered. However, in this case, the starting iteration is different for each surrogate model, i.e. a surrogate model is considered active after a prescribed number of iterations. The starting conditions are given in Tab. 6.7. The convergence of the mean value and the standard deviation for all selected models is given in Fig. 6.16 and Fig. 6.17, respectively. Note that in the case of one surrogate model, only the full model is showed. Other models are coincident with the full model. Discussion about results is given in the next section.

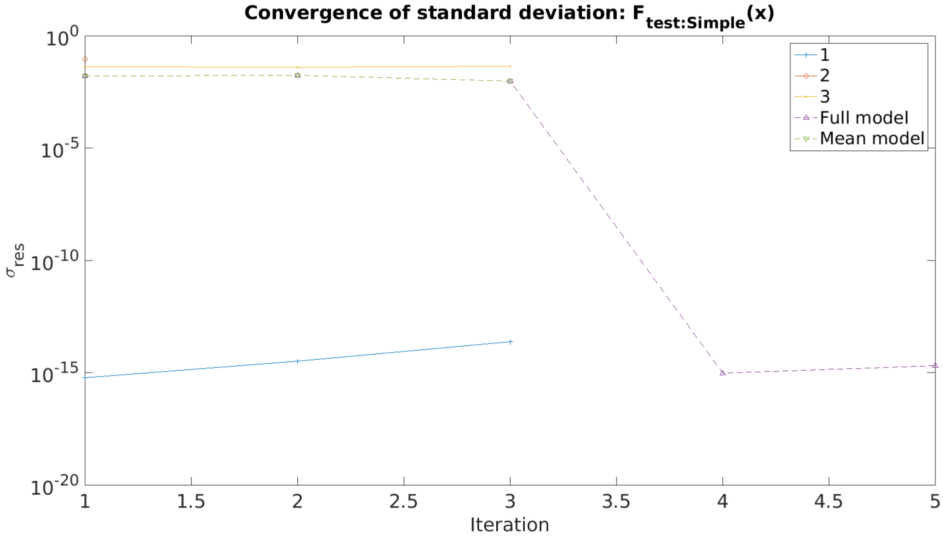


Figure 6.15: Convergence of the standard deviation for Eq. (6.30)
Note: In the third iteration, the Lagrange polynomial (Model 1) is selected the only model and the full model is composed only of the Lagrange surrogate model. Therefore, the Lagrange polynomial (Model 1) is not showed after the third iteration.

Surrogate model	Starting iteration
1	1
2	10
3	6

Table 6.7: Starting condition for the case: 5

6.4.1 Discussion about examples

In the first example (Eq. 6.28), a large number of surrogate models is used, where all the surrogate models start from the first iteration. During the convergence process, the main model is slowly converging and in the process, it is neglecting the non-converging surrogate models or slowly converging surrogate models. In the final iteration, only two the most accurate models are selected and other models are discarded. Regarding the results, the full model predicts better than the mean model for the mean value. On the other hand, it performs slightly worse in the last part of the convergence process. This problem is caused by the local weight function, which is emphasizing the wrong models. This is quickly recognized and in the final iteration, the problem despairs, i.e. both models

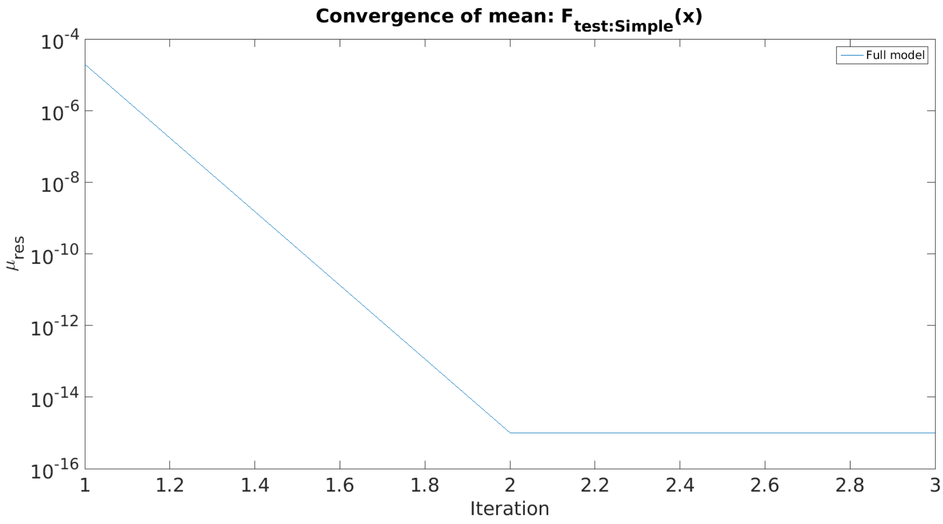


Figure 6.16: Convergence of the mean value for Eq. (6.30)
Note: Only the main model is showed. Other models are not active.

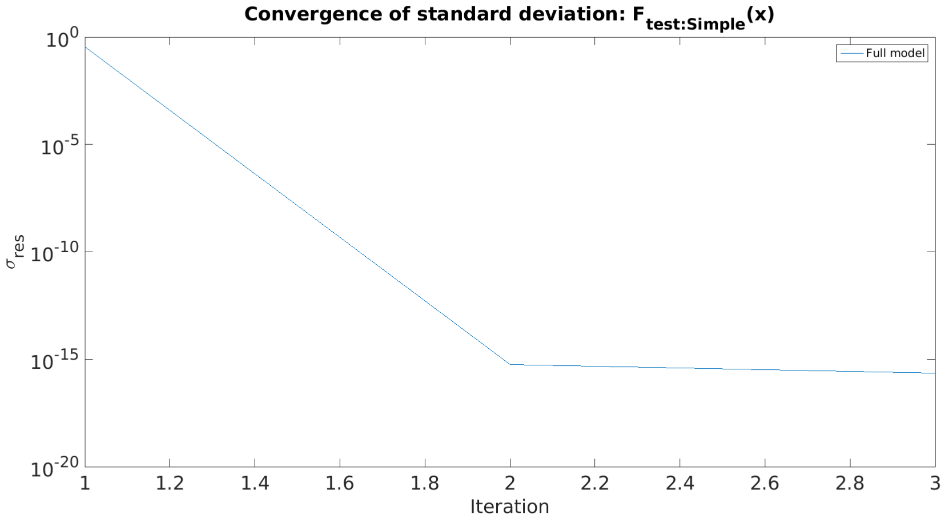


Figure 6.17: Convergence of the standard deviation for Eq. (6.30)
Note: Only the main model is showed. Other models are not active.

converge to the same accuracy. Interesting step is in the 17-th iteration, where the full model is more accurate by orders of magnitude than the other models. This is consistent with the theory in ((103),p. 364), however, it is only valid for a short range. On the other hand, the full model is somehow in the middle of all the models, which is in agreement with findings in other works.

In the second example, a smaller database of models and the same function as in the case 1 is considered. One can see that the final model is converged in fewer iterations and therefore, using a smaller number of expensive function calls. Also, in the 26-th iteration, only one surrogate model is remaining and therefore, only a full model is showed. One can conclude that using a smaller number of surrogate models leads to a faster convergence. Moreover, one can see that using surrogate models with the same basis (for example models 2 and 3 in the first example) only prolong the convergence process and the gain is negligible. This is consistent with an empirical observation, which states that if the manager has multiple persons to manage, he needs more time to finish the project. This statement is called Brooks law (111) and it is a claim about software project management according to which "adding manpower to a late software project makes it later". Moreover, using less surrogate models leads to a less computational burden.

In the third case, the discontinues function is considered and only a small number of surrogate models is used due to the conclusions obtained from the previous examples. It can be clearly seen from convergence of the mean value (Fig. 6.10) and the standard deviation (Fig. 6.11) that the multi surrogate approach is a robust interpolation method. It selects PChip and neglects the diverging ones. This aspect is required in many engineering problems as discontinuities are common in real problems. Moreover, in Fig. 6.12 the final interpolation is showed, where the adaptive algorithm nicely points out the sharp discontinuity. However, better accuracy for the mean value and the standard deviation cannot be achieved as the PChip interpolation technique cannot interpolate such a sharp discontinuity. Nevertheless, provided accuracy is sufficient for the most engineering problems.

In the fourth case and the fifth case, a simple function is considered. This function is mostly encountered in real problems as many problems converge under a low order polynomial. Moreover, from author's experience, most of the increment functions can be interpolated using a low polynomial order. Regarding the fourth case, all surrogate models start from the first iteration. However, in the last case, the interpolation process starts using only one model, which leads to a faster convergence. Therefore, one can conclude that starting with the right surrogate model provides faster convergence. Moreover, the robustness of the multi surrogate approach is not affected. In the case of the diverging surrogate model, another model is added to the database and the best combination of models is selected. In this work, other surrogate models are added to the database after a prescribed number of iterations. The possibility of adding surrogate models to the multi surrogate approach after a certain number of iterations represents a large advantage as it helps to keep the number of samples to the minimum.

6.5 Discussion about the multi surrogate approach

In modern uncertainty quantification problems, the polynomial based methods received a lot of interest in recent years. These models provide a reliable and sufficient approach in many cases. However, it can be easily encountered a problem, where these models are not sufficient (112, 113). Moreover, these problems are common in uncertainty propagation for hypersonic flows and entry propagation. Therefore, to keep the approach efficient yet robust: the multi surrogate approach is developed.

The weighted mean value approach resembles the Weighted Average Surrogate (WAS) technique. However, it can discard or add surrogate models to the problem of interest, which makes the approach very robust and efficient. New samples are tested against active surrogate models after prescribed number of iterations and based on an accuracy of given surrogate models, weights are assigned. However, in some iterations, a surrogate model can be inaccurate, yet in the overall process, the surrogate model is a good interpolant. Therefore, each weight is modified according to the global accuracy obtained over time/iterations.

The multi surrogate approach has some empirical factors, which are based on the author's knowledge and experience. The $\epsilon_{Threshold}$ is purely given by an experience and its modification can lead to a slower/faster convergence. Nevertheless, the proposed empirical approach proved to be reliable and provided accurate results. Another aspect is the time weight function (Eq. (6.9)), which can change the convergence process. In this work, only the proposed function is tested and other functions can provide a better efficiency. The same applies to the weight coefficient function (Eq. (6.6)), which is based on author's experience. Using different weight coefficient functions can give a better performance. However, this needs a larger investigation in the future. Nevertheless, all the proposed functions work well for all the tested examples.

The local weight approach is designed to locally improve the interpolation model and it proves to be working on provided examples. In the case of a larger list of active surrogate models, it provides better accuracy than the mean model. However, when the number of surrogate models is low, the additional accuracy is not high. Nevertheless, it is considered as a valuable additional tool in the surrogate model building. The basic idea of the local weight approach is that the surrogate model with the best accuracy is assumed to give the best accuracy in a close neighbourhood. Therefore, emphasizing given model can bring additional accuracy for the final surrogate model.

The local weight approach is composed of two aspects. The first aspect is the weights obtained from one particular sample. These weights represent an accuracy of given surrogate models at given particular point. However, the weighted mean model is also considered, which in many cases is the most accurate model from the ensemble of models. The second aspect is the interpolation of weights around the stochastic domain, which resemble the Gaussian process. In our case, the linear correlation function is assumed, but other correlation functions can be used. Nevertheless, one needs to hold all the restrictions. The linear correlation function is used due to its simplicity and lack of the hyper-parameter, which needs to be established by an expensive optimization technique.

The application of the proposed technique is straightforward for common problems. However, in the case of application to the cut-HDMR approach, one needs to consider only samples, which belongs to the domain of interest. The samples from lower domains are automatically 0 and therefore, these samples would slow down the weight estimation process. Another aspect is the waiting period, which is based on empirical experience. Using a small number of samples for the weight estimation can lead to a wrong weight estimation and therefore, it is suggested to use a larger number of samples. In this work, the number of test samples is driven by the number of waiting iterations. In other words, the adaptive scheme sample the stochastic domain with few samples and after a prescribed number of iterations, the weights are estimated. During the waiting period, the mean weights are kept and the new local weights are automatically 0. The local weights are recomputed, when the algorithm is used.

Based on the results from the test cases and empirical experience, several rules for the multi surrogate modelling can be established:

- Keep the number of active surrogate models to the minimum

- The considered surrogate models should be modelled in a different way, i.e. avoid using various modifications of the same technique such as Kriging with the linear trend function and Kriging with the second order polynomial trend function etc...
- Start using surrogate models when it's appropriate
    - Polynomials for the first few iterations
    - Piecewise interpolation for discontinues problems, i.e. after several iterations
    - Kriging models for highly oscillatory functions, i.e. after a large number of iterations

6.6 Conclusion

In this chapter, the multi surrogate approach is developed. The implementation of the multi surrogate interpolation technique is a necessary step in order to make the method robust and capable to handle all engineering problems. Combination of various surrogate models can make our method an effective and robust uncertainty propagation tool.

The developed multi surrogate interpolation method resembles the Kriging model; however, it has several distinctions. The mean weight approach reflects the trend function in Kriging model. It is composed as a weighted mean of the active surrogate models. The local improvement approach resembles the spatially autocorrelated error; however, in our case, the final interpolant is composed as a sum of the local improvement functions for all the active surrogate models.

One of the main aspects, which distinguishing our technique from currently available techniques, is the possibility to add or discard the interpolation techniques from the pool of active models. However, this requires an iterative process, i.e. behaviour of each surrogate model is observed over time. Based on these observations, diverging surrogate models are discarded and the final interpolant is constructed from the most accurate models.

However, it was found that using large pool of surrogate models leads to a prolonged convergence and such that to increase in the expensive function calls. This is not desired behaviour and therefore, it is suggested to add models to the active pool after a certain number of iterations and accordingly to the solved problem. This step ensures that the model is robust yet it remained an effective tool for most commonly encountered problems.

Nevertheless, the developed multi surrogate interpolation technique can work effectively without a prior knowledge of the behaviour of the selected interpolation techniques, i.e. each problem can start with a large pool of surrogate model. However, the convergence is not optimal.

Chapter 7

Independent Polynomial Interpolation technique

In this chapter, the focus is given to a development of a new surrogate model, which can handle discontinuities in the function of interest. The proposed technique resemble spline method, however, it is built differently and offers a greater possibility for future modifications. The basic idea is to have a fully discontinuous model, which can be continuous on over a defined range. The continuous solution is provided with a specific selection of a differencing scheme.

The chapter is structured in the following way: The first section is given to a literature review, where various techniques for discontinuous interpolation are discussed. In the second section, the theory of the Independent Polynomial Interpolation (IPI) is defined. The third section is given to the boundary conditions, which are required by IPI. In the fourth section, the numerical application is discussed and the fifth section is given to the applied examples, where the performance of IPI is tested. The last two sections are given to the discussion about IPI and conclusion.

7.1 Literature review

One of the largest problems in the surrogate modelling are discontinuities in the function of interest. A sharp change in the function of interest cannot be handled by many surrogate models and many Partial Differential Equations (PDE) are discontinuous in nature. For example, the Burgers' equation (114) has a discontinuity, which appears over time and this makes the problem even more complicated. Unfortunately, this is a common problem in many engineering examples.

One of the first approaches, which tries to handle discontinuous functions is the well known piecewise linear interpolation (115). It converges in all cases, however, the convergence process is very slow. Another technique, which is commonly used is the spline technique (115, 116). This technique has a large portion of variations and it was successfully used in many problems. Interesting interpolation techniques is the cubic spline (117), which consider a polynomial of degree ≤ 3 and it requires continuity of the first and second derivative. This is an advantage in a case of continuous or oscillatory functions, however, for sharp discontinuity, this represents a serious flaw. Modification of the technique, which handles better sharp discontinuities is the Shape-Preserving Piecewise Cubic polynomial (PChip) (115, 118). This modification assumes a zero-th derivative in the local optimum obtained from a set of samples. PChip works very well for functions with discontinuities, however, in a case of the oscillatory function, this assumption makes the interpolation not so accurate as the cubic spline. Another interesting version of a spline is the Akima spline (119). In this case, derivatives are constructed in a geometric mean way. Unfortunately, for the Akima spline and the PChip, the N-Dimensional approach is still missing. Other very interesting modifications of spline technique such as Not-a-knot, B-splines, and Bezier curves are introduced in the work of Bartels et al. (116) and Levy (117).

Another interesting approach for discontinuous functions are wavelets (112, 120, 121), which can approximate discontinuous functions very well. On the other hand, in the case of continuous problems, they converge in much slower rate than the polynomial chaos (112). Also, wavelets suffer from the Curse of Dimensionality (CoD), however, this can be overcome to a certain point with an adaptive scheme. Interesting combination of polynomials and wavelets is given in the work of Lee et al. (122), where the proposed combination is applied to a filtering problem. Application of wavelets enhanced with an importance sampling to the uncertainty quantification problem is given in the work of Schiavazzi et al. (123). In given work, the approach is tested on the famous Kraichnan-Orszag problem, which represents a coupled system of non-linear ODEs.

Another approach, which tries to handle discontinuous problems are the Multi-Element generalized Polynomial Chaos (ME-gPC) (124). The proposed technique combines gPC with a segmented approach i.e., it separates the stochastic space into smaller domains, which are then interpolated with the low order generalized Polynomial Chaos (gPC). This technique is very robust, however, due to the region segregation, it suffers from CoD. The adaptive approach to the ME-gPCE is considered in the work of Li and Stinis (125) and Wan and Karniadakis (126, 127), where the sub-domains are selected on a base of the local influence. This helps to mitigate CoD, however, it still represents a large issue for ME-gPC. Nevertheless, ME-gPC provides very robust Uncertainty Quantification (UQ) technique. The application of ME-gPC on a real case is in the work of Kewlani and Iagnemma (128), where Mobile Robot Dynamics is considered. This work provides time comparison between common gPC and ME-gPC.

An interesting variation of gPC is the Iterative generalized Polynomial Chaos (i-gPC). This approach handles problems with discontinuities in the function slightly better and it also provides a good convergence in the case of continuous function. Another approach called the simplex method (129) creates a local small order polynomial over simplexes (N-D triangles). The method adaptively samples the stochastic domain to adapt to the discontinuities in the function. Comparison of i-gPC, wavelets, and simplex method is given in the work of Lucor et al. (130).

In the work of Ma (35) is provided a combination of HDMR modelling and adaptive algorithm for discontinuous problems. The proposed method works well for a variety of discontinuous problems, however, it requires a large number of samples. The Pade-Legendre and its application to the fluid dynamics simulation is given in the work of Chantrasmi (48). Pade approximation uses a ratio of two polynomials instead of one polynomial of a finite order. This helps to solve oscillation problems around discontinuities. Application of Pade-Legendre approximation to the Gibbs problem is done in the work of Hesthaven et al. (131). Another variation of Pade approximation is given in the work of Tampos et al. (132), where Chebyshev polynomials are used.

In recent years, the discontinuous interpolation gains a large interest. Many engineering problems suffer from discontinuities in the function and the well known non-intrusive gPC is not able to handle these problems. In this part, a new interpolation technique is developed, which tries to handle the problem of discontinuity in the function of interest.

7.2 Theory of Independent Polynomial Interpolation technique

The Independent Polynomial Interpolation (IPI) idea slightly differs from the other approximation theories. The basic idea is that the known sample influences only a small area in its neighbourhood and with other known samples in a close neighbourhood of the sample, one can construct a local polynomial. Therefore, the interpolation technique is constructed using a small number of samples and its domain of influence is only local. This limits possible oscillation or divergence of the final model. In other words, if one of the models starts to diverge, its influence is only local. One can see that the same motivation lies behind the spline interpolation.

To explain the mathematical approach, let us consider the following model

$$sf_{IPI}(\mathbf{x}) = \sum_{r=1}^{T_c} \overline{w_r(\mathbf{x})} pf_r(\mathbf{x}) \tag{7.1}$$

where T_c represents the number of samples in given stochastic domain, $\overline{w_r(\mathbf{x})}$ represents a weight function for a given sample and $pf_r(\mathbf{x})$ represents the independent surrogate model. In this work, only polynomials are considered for $pf_r(\mathbf{x})$. The final model, $sf_{IPI}(\mathbf{x})$, is constructed as a sum of local polynomials, where each polynomial have a restricted area defined by the weight function, $\overline{w_r(\mathbf{x})}$.

Let us first focus on weight function $\overline{w_r(\mathbf{x})}$. The weight function assumes that the influence of the sample slowly fade away and it is zero in the next sample. This requires a gridded sampling, i.e samples have to be distributed in a grid way (see Fig. 7.1). Once the samples are distributed, one can construct the following weight function

$$\overline{w_r(x)} = \begin{cases} 1 - \frac{x - {}^{cp}x}{{}^{c2}x - {}^{cp}x} & {}^{cp}x < x < {}^{c2}x \\ \frac{x - {}^{c1}x}{{}^{cp}x - {}^{c1}x} & {}^{c1}x < x < {}^{cp}x \\ 1 & x = {}^{cp}x \\ 0 & x \geq {}^{c2}x \\ 0 & x \leq {}^{c1}x \end{cases} \tag{7.2}$$

where ${}^{cp}x$ represents the central sample, ${}^{c1}x$ represents the sample on the left and ${}^{c2}x$ represents the sample on the right. An example of the weight function is

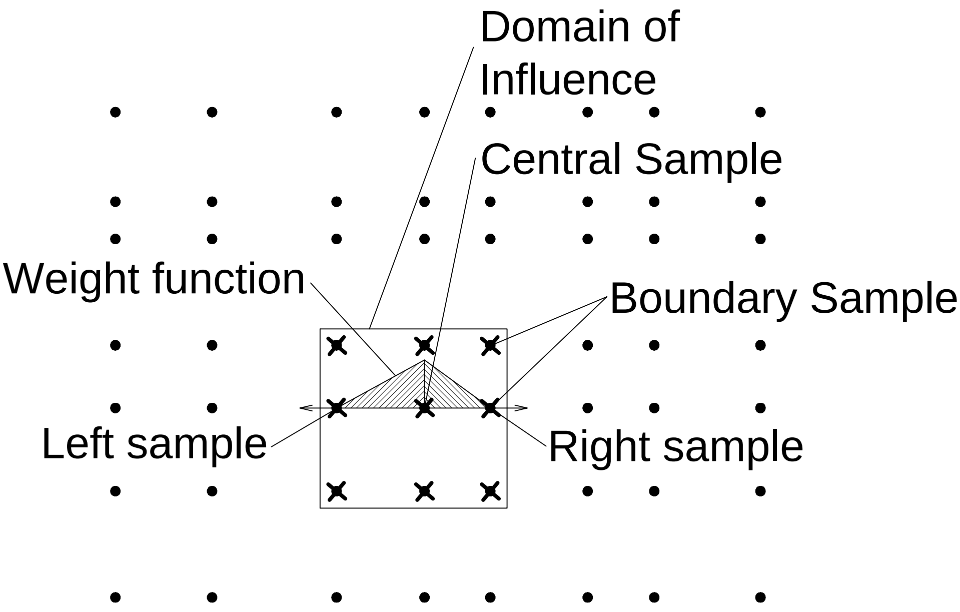


Figure 7.1: The grid sampling and the domain of influence for IPI

showed in Fig. 7.1. The weight function in N-D system is defined in the following way

$$\overline{w_r(x_1, x_2, ..., x_n)} = \overline{w_r(x_1)}\ \overline{w_r(x_2)}...\overline{w_r(x_n)} \tag{7.3}$$

where n represents the number of the stochastic variables. One can quickly realize that samples, which lies outside of the weight function have zero influence on the interpolation process. This ensures that the polynomial is only local. Moreover, the proposed weight approach ensure that the final weight is always 1 over the whole stochastic domain, i.e.

$$1 = \sum_{r=1}^{T_c} \overline{w_r(\mathbf{x})} \tag{7.4}$$

This condition ensures that the stochastic domain is properly interpolated.

Let us now define the polynomial function, $pf_r(\mathbf{x})$. In the proposed technique, there are two types of polynomial functions. The first one the interior region (common known samples) and the second one is defined for the boundaries of the stochastic domain. At the moment, let us focus only on the interior region, where the polynomial is constructed from the central point and the closest samples. In this case, the number of samples for given polynomial is given as 3^n, where n is the number of dimensions. The easiest solution is to construct a polynomial only using given samples. However, we want to provide a continuous solution as many real life cases are continuous. We impose a derivative condition on each boundary sample and therefore, each boundary sample consist of a derivative and its value. The key aspect is construction of the required derivative, which is discussed later. Another aspect represents a boundary sample, which lies in the corner of the domain of influence (see Fig. 7.1). For these samples, a cross derivative is pre-described and the application of derivatives is showed in Fig. 7.2. Using all available information and to create an interpolation approach, one can construct an univariate polynomial of the 4-th order, which is extended to n-dimensional space via the tensor product. In other words, we construct $4n$ order polynomial. This polynomial is stable, i.e. it does not start to oscillate and it provides enough smoothness for the most cases. For example, the 4-th order polynomial in 1-D reads

$$pf_r(x_1) = c_C + c_4x_1^4 + c_3x_1^3 + c_2x_1^2 + c_1x_1 \tag{7.5}$$

and for the 2-D case, it reads

$$\begin{gathered} pf_r(x_1, x_2) = c_0 + c_1x_1^4 + c_2x_1^3 + c_3x_1^2 + c_4x_1 + c_5x_2^4 + c_6x_2^3 + c_7x_2^2 \\ +c_8x_2 + c_9x_1^4x_2^4 + c_{10}x_1^4x_2^3 + c_{11}x_1^4x_2^2 + c_{12}x_1^4x_2 + c_{13}x_1^3x_2^4 + c_{14}x_1^2x_2^4 + c_{15}x_1x_2^4 + \\ c_{16}x_1^3x_2^3 + c_{17}x_1^2x_2^3 + c_{18}x_1x_2^3 + c_{19}x_1^3x_2^2 + c_{20}x_1^3x_2 + c_{21}x_1^2x_2^2 \\ +c_{22}x_1^2x_2 + c_{23}x_1x_2^2 + c_{24}x_1x_2 \end{gathered} \tag{7.6}$$

where c_i are unknown coefficients yet to be obtained. To construct all the basis of local polynomials, the tensor product expansion is used and polynomials for a larger number of the stochastic variables are constructed accordingly. Using the polynomial and its derivatives, one can construct Vandermonde matrix (133). With all the samples in the domain of influence (Fig. 7.2) and given Vandermonde matrix, one can construct a set of linear equations. For the 1-D polynomial example showed earlier, the Vandermonde matrix reads

$$\begin{bmatrix} 1 & {}^{cp}x & {}^{cp}x^2 & {}^{cp}x^3 & {}^{cp}x^4 \\ 1 & {}^{c1}x & {}^{c1}x^2 & {}^{c1}x^3 & {}^{c1}x^4 \\ 1 & {}^{c2}x & {}^{c2}x^2 & {}^{c2}x^3 & {}^{c2}x^4 \\ 0 & 1 & 2\ {}^{c1}x & 3\ {}^{c1}x^2 & 4\ {}^{c1}x^3 \\ 0 & 1 & 2\ {}^{c2}x & 3\ {}^{c2}x^2 & 4\ {}^{c2}x^3 \end{bmatrix}$$

where ${}^{cp}x$ represents the central sample, ${}^{c1}x$ represents the sample on the left and ${}^{c2}x$ represents the sample on the right. Note that the last two rows in the Vandermonde matrix represents the polynomial for the derivative condition imposed on the boundary samples. Vandermonde matrix for high dimensional spaces, e.g. 2-D and 3-D, are defined accordingly.

To obtain the unknown coefficients in the set of linear equations, one needs to solve the following equation

$$\mathbf{c}_r = \mathbf{V}_r^{-1} \times \mathbf{Y}_r \tag{7.7}$$

where $\mathbf{V}_r$ represents the previously described Vandermonde matrix for polynomial pf_r and $\mathbf{Y}_r$ represents the vector of sample values and sample derivatives for polynomial pf_r. Note that only samples in given domain of influence are used, i.e. each polynomial is solved independently. However, using simple inversion of Vandermonde matrix in Eq. (7.7) is prone to the numerical errors. Therefore, in this work, the Moore-Penrose pseudo-inverse (134) is used.

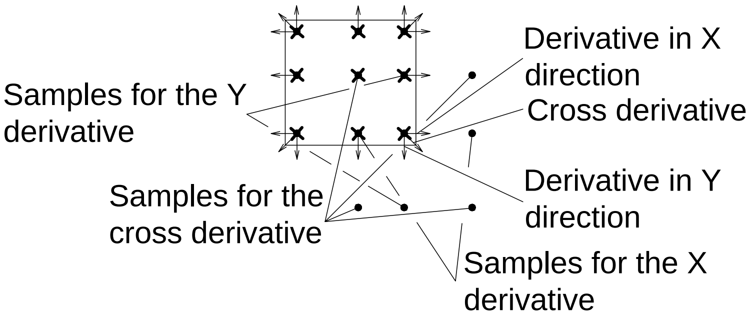

Derivative in X direction
Samples for the Y derivative
Cross derivative
Derivative in Y direction
Samples for the cross derivative
Samples for the X derivative

Figure 7.2: The local polynomial and its domain of influence for IPI

Once all the local polynomials are fully established, one can create a final interpolation model using Eq. (7.1). How to obtain derivatives for a construction of the polynomial is a topic of the next section.

7.2.1 Derivatives of the domain of influence

In many interpolation techniques, which require derivatives, a tricky approach is used. For example, spline (116, 135) use neighbourhood samples to establish derivatives. For Piecewise Cubic Hermite Interpolating Polynomial (PChip) (115), a sort of weighted finite differences is used. In this case, the neighbourhood samples are considered. Contrary to this, Kriging enhanced with derivatives requires derivatives directly from the code of interest. These derivatives are usually obtained with a finite differences method or the expensive code provides derivatives directly. In this work, derivatives are obtained from neighbourhood samples and a central finite differencing scheme is used to provide a continuous model. To illustrate the method, the following form of 1-D derivative is used:

$$\frac{\partial f({}^{p}x_i)}{\partial x_i} \approx \frac{f({}^{1}x_i) - f({}^{2}x_i)}{({}^{p}x_i - {}^{1}x_i) + ({}^{2}x_i - {}^{p}x_i)} \tag{7.8}$$

where ${}^{p}x_i$ represents the sample of interest, ${}^{1}x_i$ represents sample on the left of the sample ${}^{p}x_i$ and ${}^{2}x_i$ represents sample on the right of the sample ${}^{p}x_i$. The higher order partial derivatives are computed accordingly and graphical representation of the proposed scheme is shown in Fig. 7.2. Using the central differencing scheme, we assume locally smooth problem around each sample. One can consider a different approach for the derivative scheme, however, in that case, the final model is not continuous. Nevertheless, this step can be advantageous in various aspects such as a discontinuity in the function of interest.

7.3 Boundary conditions

There are two boundary aspects, which needs to be addressed. The first aspect represents the boundary derivative as one cannot use the central differencing scheme for the boundary samples of the stochastic domain. The second aspect represents the boundary polynomial as one cannot use the full 4-th order polynomial. The boundary polynomial has to be lower and it has to respect the number of provided samples (see Fig. 7.3). Let us first address the boundary derivative.

For the boundary derivative, one can quickly realize that slightly different approach have to be taken. One can construct the required derivative accordingly to the position of the central sample (${}^{p}x_i$). Note that the central sample for given local polynomial, ${}^{p}x_i$, represents the centre of the considered polynomial and it differs from the central sample, ${}^{c}x_i$, defined in the previous chapters. If there are only samples on the left or on the right, one has to switch to the forward or backward differencing scheme. In multiple dimensions, this leads to a combination of the central and the forward/backward differencing scheme. To illustrate the approach closer, one can consider sample ${}^{p}x_i$. If there exist samples on the left (${}^{1}x_i$) and on the right (${}^{2}x_i$), one use these samples. On the other hand, if one side is missing (sample ${}^{1}x_i$ or sample ${}^{2}x_i$), one replace the missing sample with the central point (${}^{p}x_i$). Clear example is showed in Fig. 7.3, where to establish the derivatives in x and y directions for the central sample (the corner sample), only the forward scheme is used. To establish the cross derivative, the only combination of the forward differencing scheme is used, which leads to a construction of the domain of influence from 4 samples (as showed in Fig. 7.3). The proposed approach is summarized in Alg. 9.

Algorithm 9 Algorithm for obtaining derivatives at the boundary samples

```
Initialize the process:
1. Consider sample of interest ${}^{p}\mathbf{x}$
2. Obtain the requested directions of the derivative
for $i = 1$ to all requested directions do
    if Samples $({}^{p}x_1, ..., {}^{1}x_i, ..., {}^{p}x_n) \wedge ({}^{p}x_1, ..., {}^{2}x_i, ..., {}^{p}x_n)$ exist then
        1. Consider the central differencing scheme in a given direction, i.e. use samples $({}^{p}x_1, ..., {}^{1}x_i, ..., {}^{p}x_n)$ and $({}^{p}x_1, ..., {}^{2}x_i, ..., {}^{p}x_n)$ to estimate the required derivative
    else
        1. Consider the forward/backward differencing scheme in a given direction, i.e. use samples $({}^{p}x_1, ..., {}^{1}x_i, ..., {}^{p}x_n)$ and $({}^{p}x_1, ..., {}^{p}x_i, ..., {}^{p}x_n)$ or $( {}^{p}x_1, ..., {}^{p}x_i, ..., {}^{p}x_n)$ and $( {}^{p}x_1, ..., {}^{2}x_i, ..., {}^{p}x_n)$ to estimate the required derivative
    end if
end for
3. Construct the requested derivative
Finalize the process:
1. Store the constructed derivative
End
```

Another aspect represents the boundary polynomial. One cannot construct the full polynomial as described in the previous section (see Sec. 7.2). In order to construct the boundary polynomial, it is necessary to take into account the position of the sample. In other words, the order of polynomial depends on the position of the central sample.

The first step is to obtain the number of samples, which lie in a close neighbourhood. This is done with the following equation

$$N_{samples} = \prod_{i=1}^{n}(k_i + 1) \tag{7.9}$$

where n represents the number of stochastic variables and k_i represents the number of neighbourhood samples, e.g. $k_i = 2$ if there is a sample on the left and right, $k_i = 1$ if there is a sample only on the right or on the left. Once the number of samples is known, the order of the polynomial can be established. Note that if $N_{samples} = 3^n$, one can use the full polynomial described in the previous section (see Sec. 7.2). Otherwise, Alg. 10 has to be used.

Algorithm 10 The polynomial basis creation algorithm for the boundary samples

```
Initialize the process:
1. Consider sample of interest ${}^p\mathbf{x}$
for $i = 1$ to the number of stochastic variables $n$ do
    if Sample on the left $({}^px_1, ..., {}^1x_i, ..., {}^px_n)$ and on the right $({}^px_1, ..., {}^2x_i, ..., {}^px_n)$ exists then
        1. Construct polynomial basis $x_i^P$ for $P = 1...4$
        2. Store polynomial basis into set $PS$
    else
        1. Construct polynomial basis $x_i^P$ for $P = 1...3$
        2. Store polynomial basis into set $PS$
    end if
end for
1. Construct a tensor product of all polynomial basis in set $PS$
2. Create the final polynomial
Finalize the process:
1. Store the constructed polynomial
End
```

To illustrate the proposed algorithm, let us consider the following example. Assume a 2-D stochastic space and a corner sample (see Fig. 7.3). Using Alg. 10, the final polynomial reads

$$\begin{gathered} pf_i(x_1, x_2) = c_0 + c_1x_1^3 + c_2x_1^2 + c_3x_1 + c_4x_2^3 + c_5x_2^2 \\ +c_6x_2 + c_7x_1^3x_2^3 + c_8x_1^2x_2^3 + c_9x_1x_2^3 + c_{10}x_1^3x_2^2 + c_{11}x_1^3x_2 + c_{12}x_1^2x_2^2 \\ +c_{13}x_1^2x_2 + c_{14}x_1x_2^2 + c_{15}x_1x_2 \end{gathered} \tag{7.10}$$

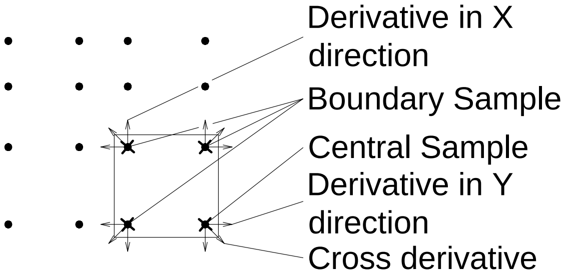


Figure 7.3: IPI boundary approach for the 2-D stochastic domain

7.4 Numerical application of the Independent Polynomial Interpolation

IPI has straightforward application. First step represents the construction of derivatives. For each sample in the stochastic domain one need to construct all the required derivatives, e.g. for $n = 3$ the following set of derivatives is required: $\frac{\partial f(\mathbf{x})}{\partial x_1}$, $\frac{\partial f(\mathbf{x})}{\partial x_2}$, $\frac{\partial f(\mathbf{x})}{\partial x_3}$, $\frac{\partial f(\mathbf{x})}{\partial x_1 \partial x_2}$, $\frac{\partial f(\mathbf{x})}{\partial x_1 \partial x_3}$, $\frac{\partial f(\mathbf{x})}{\partial x_2 \partial x_3}$, $\frac{\partial f(\mathbf{x})}{\partial x_1 \partial x_2 \partial x_3}$. We are considering only mixed derivatives up to the order of n. For samples inside the domain, the central differencing scheme (Eq. (7.8)) is used and for boundary samples, Alg. 9 is used. Once for every sample has calculated all necessary derivatives, the independent polynomials can be established. However, in the first part, the domain of influence is created (see Fig. 7.1) for each sample in the stochastic domain. For each domain of influence, the polynomial is established, where for the interior samples, the common polynomial is used and for the boundary samples, Alg. 10 is used. Also, for each sample, the weight function is constructed (Eq. (7.2) and Eq. (7.3)). Once the polynomials are created and the derivatives are established, the algorithm can estimate the unknown coefficients for given polynomials using Eq. (7.7). The final model (Eq. (7.1)) is than established as a sum of all the polynomials over the whole stochastic domain. The whole process is described in Alg. 11.

Algorithm 11 Independent Polynomial Interpolation - part 1

```
Initialize the process:
1. Obtain samples in the stochastic domain, $\mathbf{X}$
2. Obtain responses $\mathbf{Y}$ for samples obtained in step 1. Only responses are obtained as derivatives are constructed in the following steps.
for $r = 1$ to the number of samples $T_c$ do
    1. Consider sample $\mathbf{x}_r$
    2. For sample $\mathbf{x}_r$ calculate the number of samples in a close neighbourhood, $N_{samples}$ (Eq. (7.9))
    if $N_{samples} = 3^n$ then
        1. Use the central differencing scheme for all required derivatives, e.g. for $n = 2$ the following derivatives are required: $\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \frac{\partial f(\mathbf{x})}{\partial x_1 \partial x_2}$
    else
        1. Call Alg. 9 for each required derivative, e.g. for $n = 2$ the following derivatives are required: $\frac{\partial f(\mathbf{x})}{\partial x_1}, \frac{\partial f(\mathbf{x})}{\partial x_2}, \frac{\partial f(\mathbf{x})}{\partial x_1 \partial x_2}$
    end if
end for
for $r = 1$ to the number of samples $T_c$ do
    1. Consider sample $\mathbf{x}_r$
    2. Construct the local domain of influence for sample $\mathbf{x}_r$, e.g. for $n = 2$ and interior sample $\mathbf{x}_r$ see Fig. 7.1
    3. Construct a set of local samples $\mathbf{X}_r$ and their responses, $\mathbf{Y}_r$. Note that samples $\mathbf{X}_r$ and their responses , $\mathbf{Y}_r$ are subset of samples $\mathbf{X}$ and their responses $\mathbf{Y}$ obtained in step 1.
    4. Construct a set of derivatives for samples $\mathbf{X}_r$. These derivatives are constructed in the previous for loop
    5. For sample $\mathbf{x}_r$ calculate the number of samples in a close neighbourhood, $N_{samples}$ (Eq. (7.9))
    if $N_{samples} = 3^n$ then
        1. Construct the full 4-th order polynomial
    else
        1. Call Alg. 10
    end if
```

Algorithm 11 Independent Polynomial Interpolation - part 2

```
        6. Construct the Vandermonde matrix, $V_r$, using only samples from the
        domain of influence
        7. Compute the unknown coefficients for given polynomial (Eq. (7.7))
        8. Construct the local polynomial and store it into set $LP$
        9. Construct the weight function (Eq. (7.2) and Eq. (7.3)) and store it into
        set $WF$
    end for
    Finalize the process:
    1. Using set $LP$ and $WF$, construct the final surrogate model, (Eq. (7.1))
    End
```

The IPI represents a standalone surrogate model and such that it can be used independently. However, it is necessary to couple IPI with the adaptive scheme proposed in the previous chapters. IPI has a requirement to have samples in a grid way, which is the same requirement as for the Lagrange polynomials. This aspect is necessary due to the selected weight function; however, the nature of the weight function does not require equidistant spacing. An example of the grid sampling with various spacing between samples is given in Fig. 7.1. Therefore, when the adaptive scheme (Chap. 5) propose a sample, it is necessary to obtain additional samples in order to fulfil the grid sampling condition (Step 1. in Alg. 4 of Chap. 5).

7.5 Applied examples

To illustrate IPI, let us consider a set of examples. The first example is a continuous 1-D function, which reads

$$F_{test}(x) = (6x-2)^2 \sin(12x-4) \tag{7.11}$$

where x represents the random variable with a uniform distribution and boundaries equal to $[-1, 1]$. This function is a non-linear function and it is showed in Fig. 3.2. The same example was used for the adaptive scheme. However, in order to provide comparison of various interpolation techniques, the adaptive sampling is replaced with the Clenshaw-Curtis nodes. Function $F_{test}(x)$ is interpolated with interpolation techniques showed in Table. 7.1. These techniques represent the piecewise interpolation techniques and they can handle most of the interpolation problems. The convergence of the interpolation techniques is observed with $L2$ criteria, which reads

$$L2 = \frac{1}{N_{MC}} \sqrt{\sum_{i=1}^{N_{MC}} \left(F_{test}(\mathbf{x}_i) - sf(\mathbf{x}_i)\right)^2} \tag{7.12}$$

where $sf(\mathbf{x})$ represents the selected interpolation technique, $F_{test}(\mathbf{x})$ represents the original function and N_{MC} represents the number of tested samples, which is set to 100000. The interpolation is showed in Fig. 7.4 and the convergence history is showed in Fig. 7.5, where x-abscissa represents the number of training samples. The results are discussed in the following section.

-	Interpolation technique
1	IPI
2	Spline
3	PChip

Table 7.1: List of interpolation techniques used for function $F_{test}(x)$

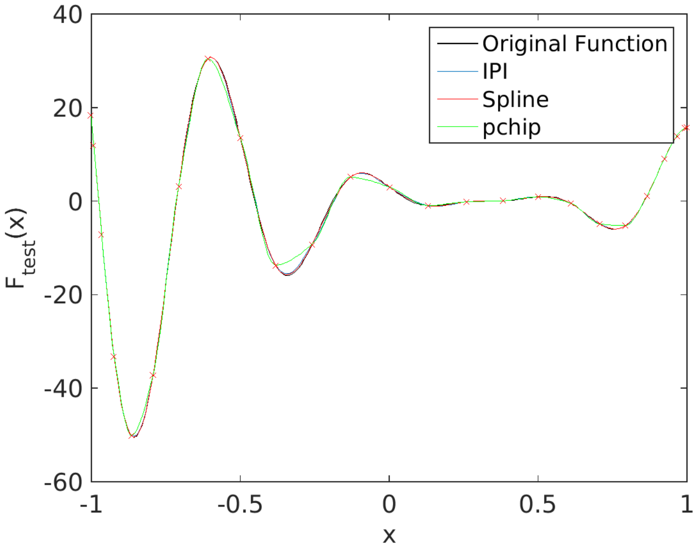


Figure 7.4: Interpolation of Eq. (7.11)

In the second example, the discontinuous function is considered and it reads

$$F_{test:Jump:1D}(x) = \begin{cases} x\sin(5x) & x \leq 0.25 \\ \frac{1}{x}\sin(5x) & x > 0.25 \end{cases} \tag{7.13}$$

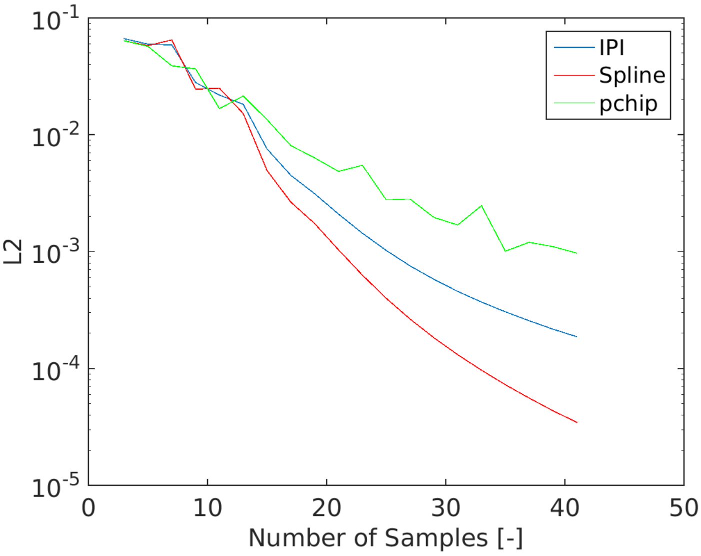


Figure 7.5: Convergence of $L2$ criteria for Eq. (7.11)

where x represents the random variable with an uniform distribution and boundaries equal to $[-1, 1]$. The proposed function is showed in Fig. 7.6 with other interpolation functions. The convergence history is showed in Fig. 7.7, where x-abscissa represents the number of training samples. The results are discussed in the following section.

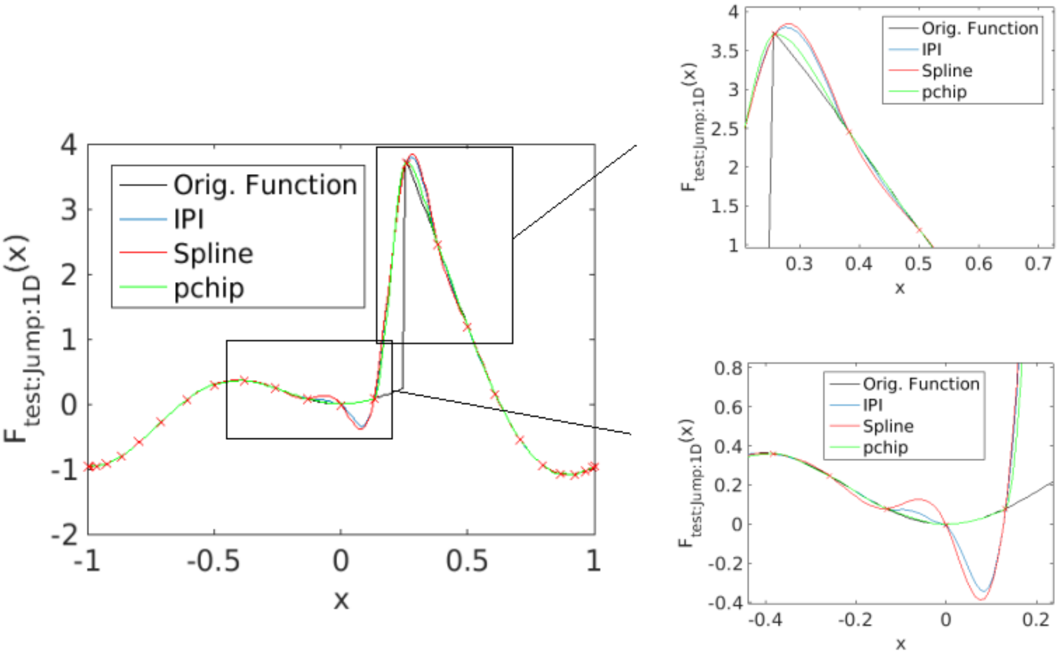


Figure 7.6: Interpolation of Eq. (7.13)

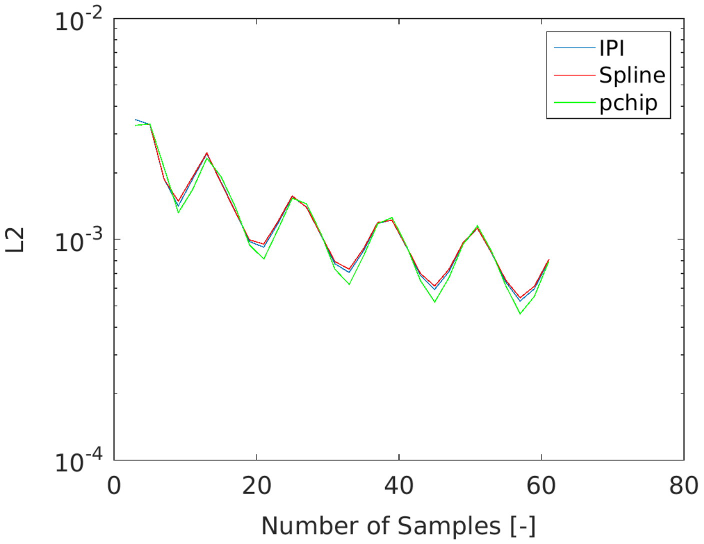
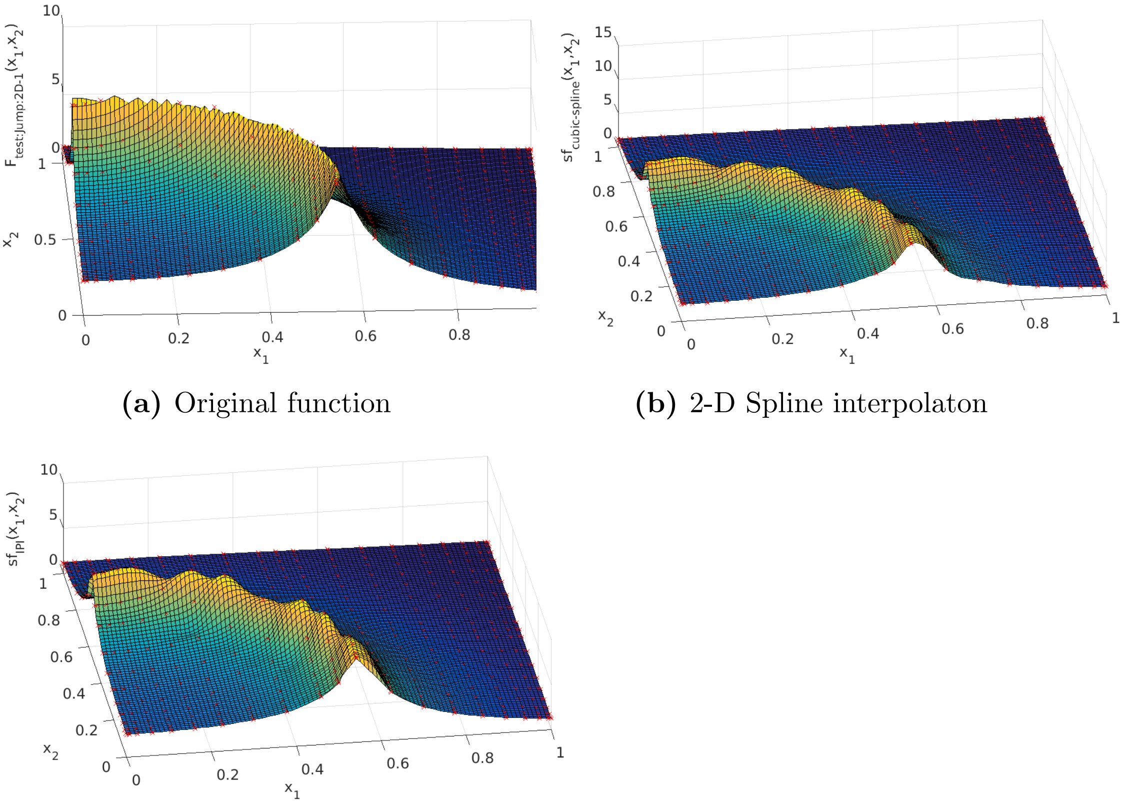


Figure 7.7: Convergence of $L2$ criteria for Eq. (7.13)

The third example is the 2-D discontinuous function, which reads

$$F_{test:Jump:2D-1}(x_1, x_2) = \frac{1}{|0.3 - x_1^2 - x_2^2| + 0.1} \tag{7.14}$$

where x_i represents the random variable with an uniform distribution and boundaries equal to $[0, 1]$. The proposed function is discontinuous and it was taken from the work of Ma (35). All the functions are showed in Fig. 7.8 and the convergence process is showed in Fig. 7.9, where x-abscissa represents the number of training samples. The results are discussed in the following section.



(a) Original function

(b) 2-D Spline interpolaton

(c) IPI interpolation

Figure 7.8: Interpolation of Eq. (7.14)

The fourth example represents a typical step function. The proposed function is 2-D and it reads

$$F_{test:Jump:2D-2}(x_1, x_2) = \begin{cases} 4 & x_1 \leq 0.25,\ x_2 \leq 0.75 \\ 2 & 0.5 < x_1 < 0.85,\ 0.25 < x_2 < 0.7 \\ 1 & x_1 \geq 0.85,\ x_2 \geq 0.7 \\ 3 & x_1 \geq 0.75,\ x_2 \leq 0.25 \\ 0 & \text{For all other } x_1 \text{ and } x_2 \end{cases} \tag{7.15}$$

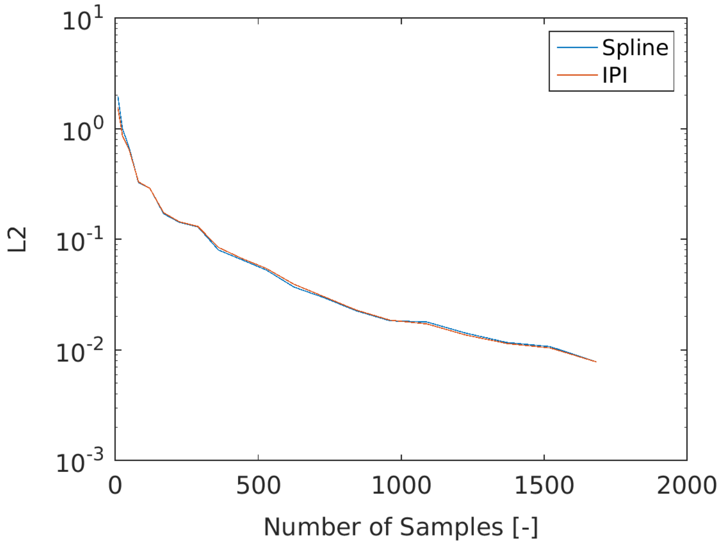


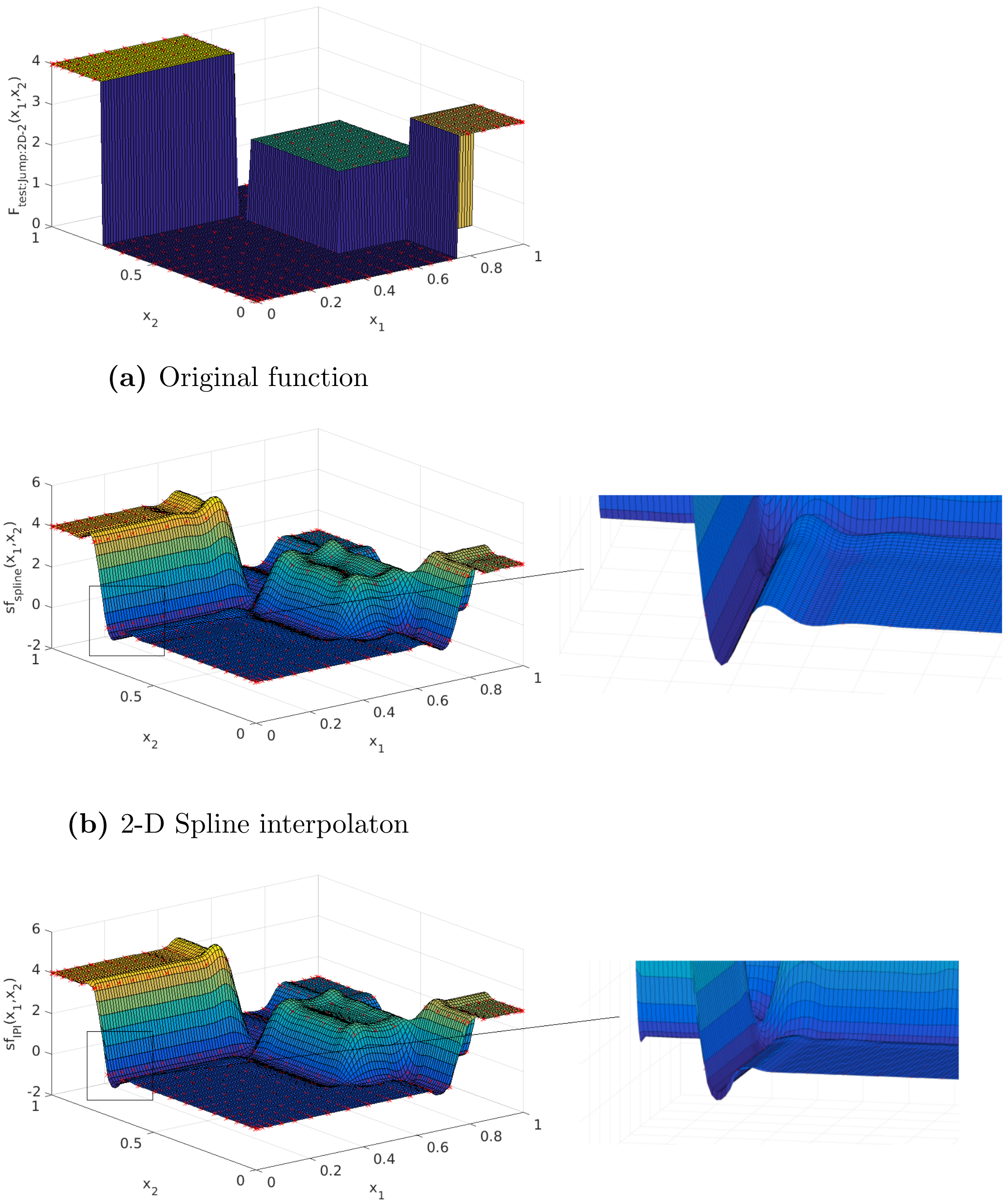
Figure 7.9: Convergence of $L2$ criteria for Eq. (7.14)

where x_i represents the random variable with an uniform distribution and boundaries equal to $[0, 1]$. All the functions are shown in Fig. 7.10 and the convergence process is shown in Fig. 7.11, where x-abscissa represents the number of training samples. The results are discussed in the following section.

7.5.1 Discussion about applied examples

In the first case, a fully continuous function is considered. It can be seen from Fig. 7.4 that IPI performs similarly to spline and in certain regions it has better behaviour than PChip. However, in the convergence process (see Fig. 7.5), IPI is between the spline function and PChip. The main aspect is the small oscillation of the function, which spline handles better due to the adopted higher order derivative. Nevertheless, IPI proves to be an efficient interpolation. On the other hand, PChip suffers from inaccuracies in the interpolation process.

In the second case, a 1-D discontinuous function is considered. From the convergence history (see Fig. 7.7), it can be seen that all functions have similar convergence history. However, around samples, 20, 26, 42 and 48 (see Fig. 7.7), the spline performs slightly worse than IPI due to the discontinuity and PChip have the best accuracy. The higher order continuity is a disadvantage in this case. Interesting part is the discontinuity itself (see Fig. 7.6), where IPI performs better



(a) Original function

(b) 2-D Spline interpolaton

(c) IPI interpolation

Figure 7.10: Interpolation of Eq. (7.15)

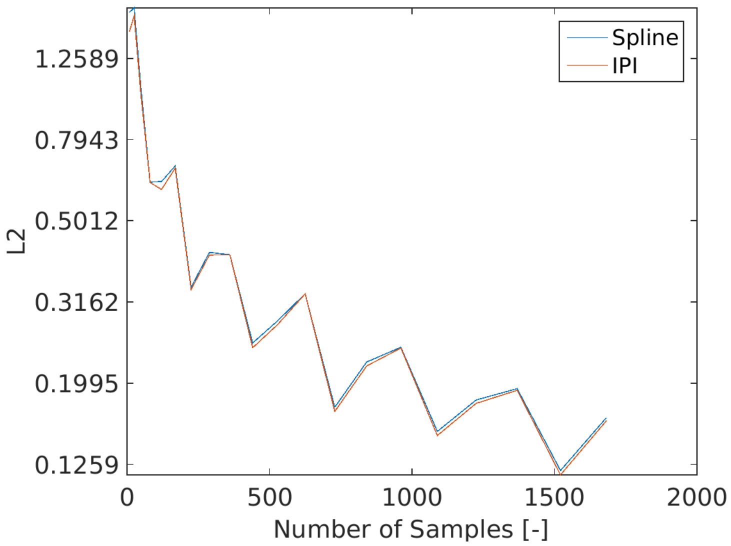


Figure 7.11: Convergence of the $L2$ criteria for Eq. (7.15)

than the spline, i.e. IPI stops to oscillate earlier (see provided details in Fig. 7.6). Note that the adaptive algorithm defined in the previous sections focuses mainly on the discontinuity (see Fig. 6.12) and therefore, IPI would provide better result compared to spline due to the smaller oscillations around discontinuities.

In the third case, smooth function with a discontinuity is showed (see Fig. 7.8a). Both interpolation functions exhibit similar properties, which is showed in the convergence process (see Fig. 7.9). The interpolants slowly converge to the function of interest, however, due to the sampling scheme used, the discontinuity is not properly interpolated. Nevertheless, IPI shows good behaviour around discontinuity, i.e. smooth change to the continuous regions.

In the fourth case, the 2-D step function is considered. Multiple steps in the stochastic space are introduced to test the behaviour of the proposed technique (see Fig. 7.10a). From Fig. 7.10c and Fig. 7.10b, it can be seen that IPI behaves much better around discontinuities. Oscillations are suppressed much earlier rather than with a spline and also, oscillations are not so large as in the case of the spline. This is reflected in the convergence process (see Fig. 7.11), where IPI have slightly better accuracy.

7.6 Discussion about IPI

The proposed interpolation technique represents a new approach to the piecewise interpolation techniques. It is based on a sum of weighted local polynomials, where the weight is slowly diminishing from the central sample. In other words, the central sample represents the center of the domain of influence and the behaviour of the function is defined by samples in a close neighbourhood. In this work, the domain of influence is considered only neighbourhood samples (see Fig. 7.1). In future works, one can focus on using various domains of influence, i.e. larger portions of samples in the domain of influence. This could lead to a higher order polynomials, which could be advantages in a case of continuous function. However, this needs larger research.

Another aspect is the local interpolating polynomial. In this work, the focus is only on polynomials, however, nature of the proposed technique allows to use combinations of various techniques such as a combination of polynomials of various orders or a combination of polynomials and Kriging models. Using a various combination of techniques can bring higher accuracy and more importantly, higher robustness. Nevertheless, due to the insufficient time, this part was not closely examined.

The finite differencing scheme used to establish the derivatives is an important aspect. In this work, we use a combination of the central differencing scheme and the forward/backward differencing scheme, but the different differencing scheme can improve the convergence around the discontinuities. For example, in a case of a sharp discontinuity inside the 1-D stochastic domain (see Fig. 7.6), changing the central derivative to the left hand side derivative for the sample on the left from the discontinuity and to the right hand side derivative for the sample on the right from the discontinuity, it can lead to a completely discontinues surrogate model and such that provide a better accuracy. In other words, IPI can be fully discontinues surrogate model. However, this still remains an open question and possible future research.

One of the main advantages over the common spline functions is the solution to the proposed local polynomials. Each polynomial represents a separate problem and therefore, a solution to the set of linear equations (Eq. (7.7)) is simpler. Therefore, an extension to N-dimensional case is simple and easy. In other words, to construct a surrogate model, which can accurately approximate high dimensional (> 4) problems, is easier than in a case of spline according to author's knowledge. On the other hand, the pre-processing solution takes more time as it needs to find a N-number of solutions to the linear set of equations.

In the tested examples, the IPI behaves same as the spline interpolation. Moreover, it behaved better in the step discontinuous functions around discontinuity (see Fig. 7.10). Contrary to this, in a fully continuous and oscillatory case, the IPI behaved slightly worse than the spline. One can conclude that IPI stands between the spline and PChip. Moreover, one can modify the local polynomials to simulate the PChip behaviour, i.e. to assign zero-th derivative condition on the central sample if all neighbourhood samples have a lower/higher value. Unfortunately, this was not implemented due to the insufficient time.

7.7 Conclusion

In this chapter, a new interpolation technique is introduced. It was developed in order to have an interpolation technique, which would suit our applications. The developed technique resembles the well-known spline method, however, it has several distinguishments.

The basic idea of the proposed method is to construct interpolant as a sum of local interpolation techniques. In this work, the local interpolation technique is selected the N-D polynomial, however, the nature of the method allows to combine various interpolation techniques such as Kriging or N-D spline. Each local interpolation technique is bounded by the weight function, which decreases from the central point of the local polynomial until it reaches zero. In other words, the influence of the local polynomial is strongest at the center and diminish with distance. This ensures that the interpolation technique is only local.

Using weights to establish domain does not ensure the continuity of the interpolant. The continuous interpolant is achieved with the derivative boundary condition on each local polynomial. In this work, the central differencing scheme constructed from the neighbourhood samples is used. Combination of a various differencing scheme can make the interpolant discontinuous, which is desired in many engineering problems. However, this aspect was not explored deeper and remains an open topic.

The technique is tested on various problems and in all tested cases, it proves to be competitive interpolant. It stands between the commonly used spline and PChip. However, compared to these two techniques (and many other), our interplant offers larger flexibility. Unfortunately, this flexibility is not investigated due to time constraints and remains for future research.

Chapter 8

Multi fidelity approach

In this chapter, the multi fidelity module is introduced. The multi fidelity is designed to be a separate module and such that it works as an extension to the previously defined method. It is based on a well proven additive correction, which is here extended to the HDMR approach.

In order to measure the effectiveness of the low fidelity models, the power ratio theory is introduced. It compares the accuracy and timing of given low fidelity models and such that creates a criterion for selection of the low fidelity model.

The accuracy of the low fidelity model is measured with the error prediction scheme. The error prediction scheme is based on the linear extrapolation of the known difference between obtained samples. The prediction error predicts the error of the mean value and the variance for the low fidelity model and in combination with the power ratio theory, it creates a robust selection process for the low fidelity model.

The chapter is structured in the following way: The first section is given to the literature review, where various works on the multi fidelity modelling are discussed. In the second section, the power ratio is defined. The third section is devoted to the correction scheme for the low fidelity model and also to the selection process of the low fidelity model. The fourth section focus on the construction and application of the prediction error function. The fifth section is the numerical application of the multi fidelity module and it is divided into two subsections. The first subsection is focused on the first order increment functions and the second subsection is focused on the higher order increment functions. The sixth section is given to the applied example, where the performance of the multi fidelity method is tested and compared to the MC simulation. The final sections are given to the discussion about the multi fidelity scheme and to the conclusion.

8.1 Literature review

In the recent years, the multi fidelity approach gained a lot of interest. Its ability to reduce the computational time significantly with a minimal loss of accuracy can spread the use of uncertainty quantification methods to a wider range of problems. Mainly industry, which needs a fast and reliable solution, is interested in the multi fidelity approach.

One of the first birds in the multi fidelity modelling for computer codes using the surrogate approach is issued in Kennedy and O'Hagan (136). The autoregression approach for the multiple low fidelity models is used in given work and it is followed by many other researchers (137, 138, 139). In the work of Kuya et al. (137), a combination of the real data and the mathematical model using the co-Kriging is introduced. The co-Kriging is combined with an optimization technique in the work of Forrester et al. (140), where it is applied to the generic transonic civil aircraft wing. Moreover, in this work, advantages of co-Kriging are discussed and it provides a nice explanation of the co-Kriging interpolation. Interesting combination of the Latin HyperCube sampling and the co-Kriging interpolation is given in the work of Xiong et al. (141). The proposed work also incorporates a cross validation to estimate the validity of interpolation. The combination of a simple Kriging and a low fidelity model is done in the work of Lee et al. (142), where the Kriging is used to correct the low fidelity model to emulate the expensive model. The proposed method is applied to various aerospace problems.

Global optimization method using the multi fidelity approach and the sequential sampling is given in the work of Son et al. (138). In given work, the multi fidelity approach is taken from the work of Kennedy and O'Hagan (136), however, the scale parameter (defined in the work of Kennedy and O'Hagan (136)) is replaced with an approximation function. Another modification of the work of Kennedy and O'Hagan (136) is given in the work of Qian et al. (139), where a stationary Gaussian process and the scale parameter are assumed to be an approximate function. The multi fidelity approach is successfully applied to an optimization problem of a heat exchanger. A combination of Kriging and a genetic optimization algorithm is given in the work of Zhu et al. (143). In given work, the Kriging surrogate model is used to interpolate differences between the fidelity models. In the work of Robinson et al. (144) is showed the multi fidelity approach for optimization, where various optimization techniques are introduced including the trust region or the space mapping approach. The optimization techniques are applied to the wing design problem. Multi fidelity methods for multidisciplinary design are introduced in the work of March (145). This very interesting work considers constrained and unconstrained multi fidelity optimization using trust region approach and the optimization technique is later on modified to the multidisciplinary problems. The proposed optimization method is applied to the structural/aerodynamic optimization problem. The multi fidelity approach for optimization under uncertainty is given in the work of Ng and Willcox (146) and Ng (147), where the control variate estimator is modified to use the multi fidelity models. This approach proved to be very efficient, on the other hand, it requires a large number of function calls (for the low fidelity and for the high fidelity) and therefore, it cannot be used in CFD applications or DSMC codes. Nevertheless, in the work of Ng (147), the proposed approach considers also the timing (in terms of computational time) of given fidelity models and this is not considered in any other work according to author's knowledge.

A very important method used for the multi fidelity optimization is the space mapping method first introduced in the work of Bandler et al. (148). The space mapping method modifies the high fidelity parameter space to be an image of the low fidelity parameter space. However, the space mapping approach requires an optimization algorithm to be applied to the low fidelity model and this is not feasible for expensive computational problems. Nevertheless, the space mapping approach gained a lot of interest among researchers (149, 150, 151, 152, 153, 154). Interesting introduction into the space mapping theory is given in the work of Bakr et al. (155). In the proposed work, the space mapping optimization is clearly explained and a combination of classical optimization methods and space mapping optimization is showed. Very interesting work on the space mapping is given in the work of Robinson (156). In given work, new Surrogate Based Optimization (SBO) methods are established, where the number of input parameters can vary between the fidelity models. Review of the space mapping optimization techniques and its progress up to the year 2004 is given in the work of Bandler et al. (157).

Currently, the multi fidelity approaches used in industry are shown in Adams et al. (84). Dakota software represents a robust approach to the non-intrusive sensitivity analysis, optimization and uncertainty quantification. This software package can be connected with other mathematical models, which are handled as a black box and it is tested in various cases, where it proved its reliability. It considers the multi fidelity approach using the polynomial chaos. Another work combining the polynomial chaos and the multi fidelity approach is shown in the work of Ng and Eldred (4). The method is successfully applied to many examples including the horn acoustics problem. Various techniques for SBO are discussed in the work of Eldred and Dunlavy (158) and the efficiency of the multi fidelity approach is shown. In given work, the additive and multiplicative surrogate correction[1] is used to correct the low fidelity model. The application of the Dakota software using multi fidelity polynomial chaos for an uncertainty propagation on a wind turbine is given in the work of Padron et al. (159). An example of the application of the multiplicative correction for the trust region optimization is discussed in the work of Alexandrov et al. (160).

8.2 Power ratio theory

The basic question in the multi fidelity modelling is the comparison of fidelity models. In other words, how to select the best model or combination of models for a given problem, where one wants to obtain the best ratio of speed and accuracy. Based on the comparison, select the best model for a given problem. However, the decision process is very complicated. In this work, the focus is given on the selection of the low fidelity model, i.e. we assume that the high fidelity model is defined by the user and a list of the low fidelity models is given to the algorithm.

We define a new approach, where a decision is made on a basis of accuracy and timing of given low model. The decision process is based on power ratio P^{MF}, which represents the efficiency of given low fidelity model against the high fidelity model. The power ratio is defined in the following way

$$P^{MF} = \frac{A^{MF}}{T^{MF}} \tag{8.1}$$

where A^{MF} represents the accuracy of the low fidelity model against the high fidelity model and T^{MF} represents the time ratio of the low fidelity model and

[1] The additive and multiplicative corrections are discussed in the following sections.

the high fidelity model. The accuracy, A^{MF}, is defined in the following way

$$A^{MF} = e^{-Q^{MF}} \tag{8.2}$$

where Q^{MF} represents the quantity of interest, e.g. an error of the mean value or an error of the standard deviation. The quantity of interest Q^{MF} has to be positive and therefore, the accuracy ratio ranges from 0 to 1. The time ratio is defined in the following way

$$T^{MF} = \frac{t^L}{t^H} \tag{8.3}$$

where t^L represents the computational time of the low fidelity model and t^H represents the computational time of the high fidelity model. The time ratio is never 0 and it is always positive.

The power ratio has some important properties. Closely examining Eq. (8.1), it can be easily understood that if the time of the low fidelity model is higher than the high fidelity, no matter the accuracy, it is worthless to use the low fidelity model. To extend this approach, the model which is extremely inaccurate is also, useless as it would need to be corrected all the time. In other words, extremely inaccurate low fidelity model is useless no matter how fast it is. Therefore, there exists a separation line, which separates the efficient low fidelity models from the inefficient ones. Based on this aspect, one can construct a diagram showed in Fig. 8.1 and create a separation line which connects 0 and 1. In other words, a line which separates the efficient models from the inefficient ones. Obviously, models which lie in the region of inefficiency are not worthy to use. Therefore, the following condition can be established

$$\begin{gathered} if \\ P^{MF} < 1 \\ \textit{The low fidelity model is inefficient} \rightarrow \textit{stop using it} \end{gathered}$$

One can understand that the power ratio basically represents the efficiency of the low fidelity model. In other words, the low fidelity model with the higher power ratio should provide better efficiency. On the other hand, one can protest, that models slower yet more accurate should be a better choice. It is necessary to realize that the accuracy of a given model (discussed later) can always be corrected. On the other hand, one cannot correct the computational time for the low fidelity model. This leads us to a conclusion that the main constraint for the low fidelity modelling is the computational time. The power ratio takes all this into account and therefore, the low fidelity model with the highest ratio is selected.

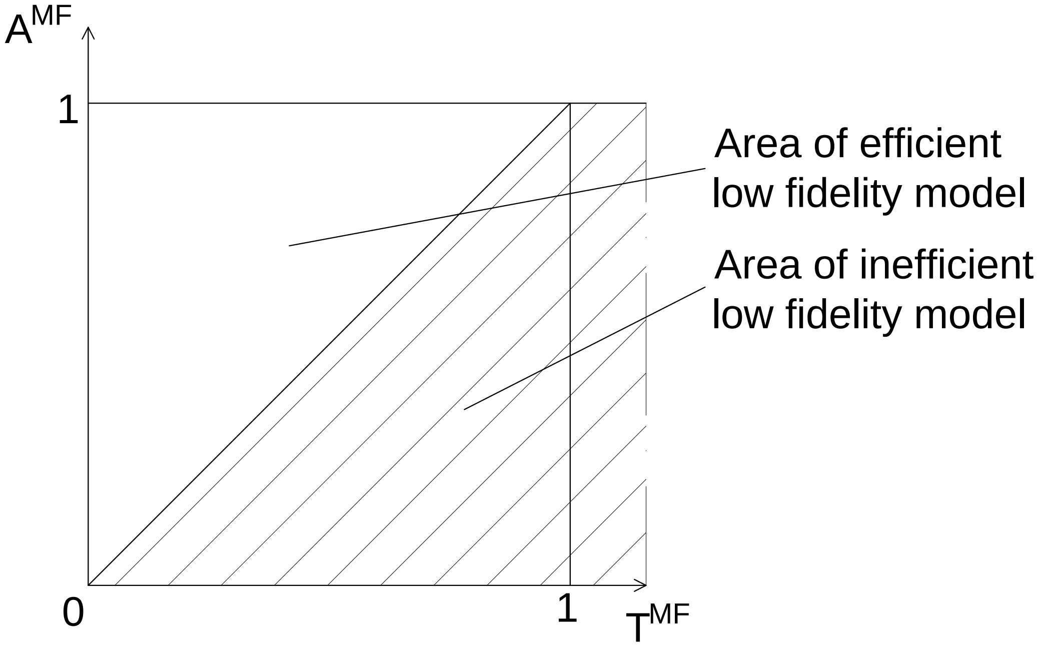

A^{MF}
1
Area of efficient low fidelity model
Area of inefficient low fidelity model
0
1
T^{MF}

Figure 8.1: The low fidelity model efficiency diagram

The power ratio represents the basic stone of multi fidelity modelling in this work. It compares the low fidelity models and provides a sort of selection criterion. However, it only defines, which model selects and not when. The topic of application of the power ratio and the process of model selection is discussed later.

8.3 Basic theory of the multi fidelity approach

One of the main advantages of the HDMR approach is the separation of the stochastic domain into smaller sub-domains. Each sub-domain represents a separate problem and therefore, it can be handled as an independent interpolation problem. In other words, one can apply the multi fidelity approach separately to each increment function. However, it is necessary that the proposed multi fidelity approach has the same properties as the function of interest, i.e. the HDMR approach must be able to be applied to the multi fidelity scheme as well.

In order to correct the low fidelity model, it is necessary to construct a sort of correction function. However, before the correction function is defined, let us discuss some basic aspects of the multi fidelity modelling. Currently, there are considered two approaches for the multi fidelity corrections, where the first one is the additive correction

$$\epsilon(\mathbf{x}) = f^{HF}(\mathbf{x}) - f^{LF}(\mathbf{x}) \tag{8.4}$$

and the second one represents the multiplicative correction

$$B(\mathbf{x}) = \frac{f^{HF}(\mathbf{x})}{f^{LF}(\mathbf{x})} \tag{8.5}$$

where $f^{HF}(\mathbf{x})$ represents the high fidelity model and $f^{LF}(\mathbf{x})$ represents the low fidelity model. We consider only the additive correction, because the integral form of DE can handle the additive correction, while the multiplicative correction represents a problem. In other words, it is necessary to perform several analytic operations (described later), which are problematic in the case of multiplicative correction.

The high fidelity model, $f^{HF}(\mathbf{x})$, and the low fidelity model, $f^{LF}(\mathbf{x})$, can be separated using the HDMR approach as both models are independent functions. Therefore, one can conclude that the additive correction (Eq. (8.4)) can be separated into the increment approach as well. The mathematical form then reads:

$$d\epsilon(\mathbf{x}) = dF^{HF}(\mathbf{x}) - dF^{LF}(\mathbf{x}) \tag{8.6}$$

where $dF^{HF}(\mathbf{x})$ represents the high fidelity increment function and $dF^{LF}(\mathbf{x})$ represents the low fidelity increment function. Moreover, due to application of the additive correction, each increment function follows Eq. (8.6). This is important aspect as all the conclusions about increment functions (see Chap. 2) are valid for the correction function. The derivation of the increment function is given in the next section.

8.3.1 Correction of the low fidelity increment function

To derive the additive increment function defined in the previous section, let us remind the additive correction function (Eq. (8.4)), which reads

$$\epsilon(\mathbf{x}) = f^{HF}(\mathbf{x}) - f^{LF}(\mathbf{x}) \tag{8.7}$$

where $f^{HF}(\mathbf{x})$ represents the high fidelity function and $f^{LF}(\mathbf{x})$ represents the low fidelity function. Eq. (8.7) is called Fidelity Error Function (FEF) through this work. More useful in the multi fidelity modelling are the derivatives of given functions, i.e. comparing the derivatives of given fidelity models. Therefore, application of derivative to Eq. (8.7) leads to the following equation

$$\frac{\partial \epsilon(\mathbf{x})}{\partial x_i ... \partial x_j} = \frac{\partial f^{HF}(\mathbf{x})}{\partial x_i ... \partial x_j} - \frac{\partial f^{LF}(\mathbf{x})}{\partial x_i ... \partial x_j} \tag{8.8}$$

Let us now recall the increment function (Eq. (2.26)), which reads

$$dF_{i...j}(x_i, ..., x_j) = \int_{^c x_i}^{x_i} ... \int_{^c x_j}^{x_j} \frac{\partial f(\xi_i, ..., \xi_j, {}^c x_{...})}{\partial \xi_i ... \partial \xi_j} d\xi_i ... d\xi_j \tag{8.9}$$

Application of Eq. (8.9) to Eq. (8.8) and with little rearrangement, one can obtain the following result

$$\int_{^c x_i}^{x_i} ... \int_{^c x_j}^{x_j} \frac{\partial f^{HF}(\xi_i, ..., \xi_j, {}^c x_{...})}{\partial \xi_i ... \partial \xi_j} d\xi_i ... d\xi_j = \\ \int_{^c x_i}^{x_i} ... \int_{^c x_j}^{x_j} \frac{\partial f^{LF}(\xi_i, ..., \xi_j, {}^c x_{...})}{\partial \xi_i ... \partial \xi_j} d\xi_i ... d\xi_j + \int_{^c x_i}^{x_i} ... \int_{^c x_j}^{x_j} \frac{\partial \epsilon(\xi_i, ..., \xi_j, {}^c x_{...})}{\partial \xi_i ... \partial \xi_j} d\xi_i ... d\xi_j \tag{8.10}$$

Following the same approach used in Sec. 2.3, one can replace the integral form into more algebraic shape and Eq. (8.10) can be rearranged into the following form:

$$dF^{HF}_{i...j}(x_i, ..., x_j) = dF^{LF}_{i...j}(x_i, ..., x_j) + d\epsilon_{i...j}(\xi_i, ..., \xi_j) \tag{8.11}$$

where $dF^{LF}_{i...j}(x_i, ..., x_j)$ represents the low fidelity increment function and $d\epsilon_{i...j}(\xi_i, ..., \xi_j)$ represents the Fidelity Error Increment Function (FEIF). FEIF represents the additive correction of the increment function (Eq. 8.6) and through the chapter, this description is used. As mentioned in the previous section, FIEF follows all rules for the increment functions.

In Chap. 2, we defined statistical properties of an increment function. Using the partial expected value (Eq. (2.33)) and the partial variance (Eq. (2.34)) from Sec. 2.3, one can establish the multi fidelity expected value and the multi fidelity partial variance. The mathematical form of the multi fidelity partial mean value and the multi fidelity partial variance reads

$$\mu_t = \frac{1}{z}\sum_{j=1}^{z}(dF_t^{LF}(\mathbf{x}_j) + d\epsilon_t(\mathbf{x}_j)) \tag{8.12}$$

$$\sigma_t^2 = \frac{1}{z-1}\sum_{j=1}^{z}(dF_t^{LF}(\mathbf{x}_j) + d\epsilon_t(\mathbf{x}_j) - \mu_t)^2 \tag{8.13}$$

where z represents the number of samples of MC simulation applied to the surrogate model of the low fidelity increment function, $dF_t^{LF}(\mathbf{x})$, and the Fidelity Error Increment Function, $d\epsilon_t(\mathbf{x}_j)$. The approximation of the Fidelity Error Increment Function, i.e. the surrogate model for FEIF, is discussed later.

Let us now closely examine the statistical properties of the multi fidelity increment function. Eq. 8.12 can be separated into two aspects - the partial expected value of the low fidelity model and the partial expected value of FEIF. This can be done without loss of generality and calculate the expected value using only FEIF, one can define the partial expected value of FEIF, which reads

$$\mu_{t_\epsilon} = \frac{1}{z}\sum_{j=1}^{z} d\epsilon_t(\mathbf{x}_j) \tag{8.14}$$

where t represents the selected increment function. This function has a large meaning for the multi fidelity modelling because it is measuring the difference between the expected value of the high fidelity model and the low fidelity model. Therefore, observing the mean value for FEIF can give the user great insight into differences of given fidelity models. This aspect is discussed later in this chapter, where the prediction algorithm for the fidelity errors is established.

On the other side, the multi fidelity partial variance cannot be completely additive without loss of generality. To explain this statement, let us assume uncorrelated variance of the low and high fidelity model, i.e.

$$Cov(dF_t^{LF}(\mathbf{x}), dF_t^{HF}(\mathbf{x})) = 0 \tag{8.15}$$

The high and low fidelity models are completely independent and hence, there is no correlation between them. In other words, Eq. (8.7) is valid only if Cond. 8.15 holds. Using above condition and Eq. (8.7), one can establish the partial variance for FEIF, i.e.

$$\sigma_{t_\epsilon}^2 = \frac{1}{z-1}\sum_{j=1}^{z}(d\epsilon_t(\mathbf{x}_j) - \mu_{t_\epsilon})^2 \tag{8.16}$$

The variance of FEIF represents the deviation from the mean error, which represents how much FEIF changes over the stochastic domain. In other words, how large are differences between models. This function is discussed later in this chapter, where the prediction algorithm for the fidelity errors is established.

8.3.2 Selection of the low fidelity increment functions

In the previous chapters, one of the main aspects of the proposed approach is to consider the increment function as a separate problem. One can easily extend this approach to the multi fidelity modelling, where only important increment functions are corrected. In other words, it is necessary to correct the low fidelity model with samples from the high fidelity model only when the increment function has a significant effect on the final result. The basic question is: which increment functions have to be corrected and how.

One cannot assume anything about the low fidelity model. Therefore, for the first order increment functions, it is necessary to obtain the central point and the boundary samples, which help us to create the first order FEIF and obtain basic knowledge about the problem. It is necessary to do this step for all the low fidelity models as one needs to understand the efficiency of each low fidelity model. However, considering the results obtained from the multi surrogate modelling (see Sec. 6.4.1), one should keep the number of low fidelity models low, e.g. one or two.

To illustrate the problem in a better perspective, consider the following problem: Let us assume a single fidelity approach, where it is necessary to obtain k^{MF_H} samples, e.g. $k^{MF_H} = 100$, to have a fully convergent solution. Each sample cost t^{MF_H} time units, e.g. $t^{MF_H} = 10$. Now, let us consider the multi fidelity approach, where number of the low fidelity models is n^{MF_L}, e.g. $n^{MF_L} = 1000$ and each of these low fidelity models cost t^{MF_L} time units, e.g. $t^{MF_L} = 1$. One can quickly realize that the single fidelity approach is already finished, while the multi surrogate approach knows only the accuracy of all the models in the central point. It could not create FEIF to establish the fidelity differences. With the illustrated problem, one can construct the following condition for a number of the low fidelity models:

$$\sum_{i=1}^{n^{MF_L}} t_i^{MF_L} < k^{MF_H} t^{MF_H}$$

In other words, the number of low fidelity models is limited by the complexity of function, i.e. how many samples are required to have a convergent model. This is reflected in the efficiency of the multi fidelity modelling because when the user sets the convergence residual low, the uncertainty propagation will converge early and the algorithm will not have enough time to use the low fidelity models. Moreover, one needs other samples in different positions in the stochastic space, which are used to create FEF. FEF needs to be created for each low fidelity model and this increases the overall computational time. Therefore, it is suggested to keep the number of the low fidelity models as low as possible.

Once, the boundary samples are obtained, the statistical properties can be obtained and the algorithm can start selecting the models required for the uncertainty propagation. The numerical application of the selection process is discussed later. Before we define the selection process, let us have a closer look at the higher order increment functions.

For the higher order increment function (≥ 2), the process is slightly different. Correcting all the higher order increment functions would lead to a large number of the high fidelity function calls and this is not an efficient approach. However, one cannot assume that all the higher order interactions are included in the low fidelity model. In other words, the low fidelity model can be simplified and a large number of the higher order increment functions can be neglected from given model. This is also, author's experience. Moreover, this could represent a serious flaw in UQ propagation if the problem has strong interaction effects such as in the hypersonic flows, where the chemical reactions take place. These reactions are highly active and have a key influence on the final shape of PDF. Therefore, it is necessary to ensure that the higher order interactions are also, included in the low fidelity model.

The way how to discover the interaction effects inside the considered model is the prediction algorithm (see Chap. 4). The prediction algorithm is exactly the same as in the previous chapters, however, this time, it is applied to the high fidelity model and all the low fidelity models. One needs to compare the low fidelity model to the high fidelity model and obtain a common set of functional increment functions, i.e. constructing a logical conjunction of two predicted sets (one for the high fidelity and one for the low fidelity). This is mathematically written in the following way

$$M^{i^{MF}}_{Low:Consider} = M^{i^{MF}}_{Low} \wedge M_{high} \tag{8.17}$$

where $M^{i^{MF}}_{Low}$ represents the set of selected increment functions from the low fidelity model i^{MF} and M_{high} represents the set of selected increment functions from the high fidelity model. Set $M^{i^{MF}}_{Low:Consider}$ considers all the low fidelity increment functions, which can be used for uncertainty propagation. However, set $M^{i^{MF}}_{Low}$ can consider also low fidelity increment functions, which are not in set M_{high}. These increment functions must not be considered because it is necessary to obtain a correction to the high fidelity model and not vice versa.

Once the prediction algorithm selects the important increment functions, the algorithm can decide, which increment functions needs to be corrected with the high fidelity model. In the first step, it is necessary to sample the boundary samples with the low fidelity model and construct the basic statistical properties. We sample the boundary samples in order to circumscribe the given stochastic domain. The low fidelity model selected for the boundary samples is the one with the highest power ratio, i.e. the predicted values are combined with the power ratio (Eq. (8.1)), which is done in the following way

$$P^{MF} = \frac{e^{-(\sigma_p^{HF} - \sigma_p^{LF^i})}}{T^{MF^i}} \tag{8.18}$$

where σ_p^{HF} represents the statistical properties of the high fidelity model and $\sigma_p^{LF^i}$ represents the statistical properties of the i-th low fidelity model. The statistical properties are obtained from the linear model of residuals (Eq. (4.34)) and only the standard deviation is considered in this work.

When the boundary samples are obtained, FEIF can be created and the statistical properties can be established. Based on the sensitivity results obtained from the first step, one can select, which increment functions needs to be re-interpolated, i.e. corrected with the high fidelity model. For uncertainty quantification problems, the mean value and the variance is used and the mathematical form reads

$$\begin{array}{l} if\ R_{\mu_t}^{MF_L^{i^{MF}}} \leq R_{set_t}^{\mu} \vee R_{(\sigma_t)^2}^{MF_L^{i^{MF}}} \leq R_{set_t}^{(\sigma)^2} \\ \textit{Consider only the low fidelity samples} \\ \textit{else} \\ \textit{Re-interpolate the low fidelity samples} \\ \textit{end} \end{array} \tag{8.19}$$

where $R_{\mu_t}^{MF_L^{i^{MF}}}$ represents the residual of the mean value for the i-th low fidelity model (see Eq. (3.21)), $R_{(\sigma_t)^2}^{MF_L^{i^{MF}}}$ represents the residual of the variance for the i^{MF}-th low fidelity model (see Eq. (3.22)) and R_{set_t} represents the residual for a local convergence process (see Eq. (3.25)). Cond. 8.19 is tested in each iteration and for each increment function considered. For the sake of clarity, in Cond. (8.19) is not showed the current iteration and given increment function. If the proposed condition is valid, the increment function is selected for re-interpolation, i.e. correction with the high fidelity model. Note that, in order to reduce the computational burden, only samples in the low fidelity model are used for the proposed condition.

8.4 Construction and application of the fidelity error function

In the previous section, the Fidelity Error Function is established and its application to the HDMR approach is also introduced. The incremental form of FEF reads

$$\epsilon(\mathbf{x}) - \epsilon({}^{\mathbf{c}}\mathbf{x}) = \sum_{i=1}^{n} \int_{{}^{c}x_i}^{x_i} \frac{\partial \epsilon(\text{¸})}{\partial \xi_i} d\xi_i + \\ \sum_{1 \leq i < j \leq n} \int_{{}^{c}x_i}^{x_i} \int_{{}^{c}x_j}^{x_j} \frac{\partial \epsilon(\text{¸})}{\partial \xi_i, \xi_j} d\xi_i d\xi_j + ... + \int_{{}^{c}x_1}^{x_1} ... \int_{{}^{c}x_n}^{x_n} \frac{\partial \epsilon(\text{¸})}{\partial \xi_1, ..., \xi_n} d\xi_1 ... d\xi_n \tag{8.20}$$

and the Fidelity Error Increment Function (FEIF) reads

$$d\epsilon_{i...j}(\mathbf{x}) = \int_{{}^{c}x_1}^{x_1} ... \int_{{}^{c}x_n}^{x_n} \frac{\partial \epsilon(\text{¸})}{\partial \xi_1, ..., \xi_n} d\xi_1 ... d\xi_n \tag{8.21}$$

Let us now closely examine equations above. There are various ways how to construct Eq. (8.21), however, in this work a point wise approach is taken. Samples from the low fidelity model and the high fidelity model are collected at particular positions and compared. To introduce the approach, let us consider a vector of error fidelity responses, which reads

$$\boldsymbol{\epsilon} = \mathbf{f}^{\mathbf{HF}} - \mathbf{f}^{\mathbf{LF}} \tag{8.22}$$

where $\mathbf{f}^{\mathbf{HF}}$ represents the vector of responses from the high fidelity model and $\mathbf{f}^{\mathbf{LF}}$ represents the vector of responses from the low fidelity model. Both responses are obtained using the same sample matrix $\mathbf{X}$. Using Eq. (8.20) and the vector of fidelity errors $\boldsymbol{\epsilon}$, one can construct a set of FEIF samples using the approach discussed in Sec. 2.3, e.g. for $n = 3$, the second order fidelity error increment function reads

$$d\epsilon_{ij}^{z_s} = \epsilon(x_i^{z_s}, x_j^{z_s}, {}^c x_{\dots}^{z_s}) - \epsilon(x_i^{z_s}, {}^c x_j^{z_s}, {}^c x_{\dots}^{z_s}) - \epsilon({}^c x_i^{z_s}, x_j^{z_s}, {}^c x_{\dots}^{z_s}) + \epsilon({}^c x_i^{z_s}, {}^c x_j^{z_s}, {}^c x_{\dots}^{z_s}) \tag{8.23}$$

where $x_i^{z_s}$ represents sample z_s from the input matrix, $\mathbf{X}$, for stochastic variable i and ${}^c x_{\dots}^{z_s}$ represents the central sample for given stochastic variables. Using all the obtained samples, one can construct surrogate model $Sd\epsilon_{i\dots j}(\mathbf{x})$.

The surrogate model, $Sd\epsilon_{i\dots j}(\mathbf{x})$, can be created using various types of surrogate models such as Lagrange interpolation, PChip or IPI. However, author's suggestion is to use the piecewise interpolation models (For example, IPI). These models prevent oscillations and do not diverge over the whole stochastic domain.

Unfortunately, the surrogate model represents a source of errors for the multi fidelity modelling. However, if the underlying function does have a complicated shape, the adaptive scheme (See Sec. 3 and Sec. 5) automatically place samples in the problematic regions. Therefore, improving the accuracy of the surrogate model and also reducing the fidelity error. On the other hand, if the underlying function is not complicated, the interpolation scheme should have no problems to interpolate the problem accurately. Again, let us remind that we assume same trends between the low fidelity model and the high fidelity model[1].

[1] Under the term, "same trends" for the low and high fidelity models is understood that both models have similar derivatives in all directions for given increment function. Also, possible discontinuity or oscillatory behaviour in the stochastic domain is presented in both fidelity models.

8.4.1 Prediction approach for FEIF

The interpolation technique used for the surrogate model, $Sd\epsilon_k(\mathbf{x})$, represents a source of error. One is never sure what is the behaviour of fidelity models between known samples. However, one can expect the same behaviour in a close neighbourhood of a known sample and create a sort of prediction around a given sample. Therefore, prediction approach for possible errors is developed.

To establish the prediction, let us have a closer look on Eq. (8.8). The difference between derivatives of given fidelity models can highlight areas of large disagreements. In other words, when the difference between derivatives is known at one particular sample, one can predict the difference in a close neighbourhood. Taking the absolute value of Eq. (8.8) and application of the linear approximation, one can extrapolate the differences between fidelities. Therefore, we can construct a Linear Fidelity Error Prediction (FEP^{Linear}) function, which is mathematically written in the following way

$$\begin{aligned} FEP^{Linear}(x_i, ..., x_j) &= \epsilon_{{}^{p}\mathbf{x}} + \left| \left(\frac{\partial \epsilon(\mathbf{x})}{\partial x_i ... \partial x_j} \Big|_{\mathbf{x} = \ {}^{p}\mathbf{x}} \right) \right| (\mathbf{x} - \ {}^{p}\mathbf{x}) \\ &= \left| \left(\frac{\partial f^{HF}(\mathbf{x})}{\partial x_i ... \partial x_j} \Big|_{\mathbf{x} = \ {}^{p}\mathbf{x}} - \frac{\partial f^{LF}(\mathbf{x})}{\partial x_i ... \partial x_j} \Big|_{\mathbf{x} = \ {}^{p}\mathbf{x}} \right) \right| (\mathbf{x} - \ {}^{p}\mathbf{x}) \end{aligned} \tag{8.24}$$

where ${}^{p}\mathbf{x}$ represents the sample, where both derivatives (low fidelity and high fidelity) are known and $\epsilon_{{}^{p}\mathbf{x}}$ represents the constant error term at position ${}^{p}\mathbf{x}$. Note that usually the correction is known at point ${}^{p}\mathbf{x}$ and therefore, the constant error term is equal to 0.

However, using only linear extrapolation is not efficient and one can extend this approach to involve surrogate model $Sd\epsilon_t(\mathbf{x})$. Therefore, using definite integral for given sample ${}^{p}x_i$, one can obtain an increment to the prediction at particular point along given dimension, i.e.

$${}^{p}dSFEP_{i...j}(x_i, ..., x_j) = \int_{{}^{p}x_i}^{x_i} ... \int_{{}^{p}x_j}^{x_j} \left| \frac{\partial Sd\epsilon_{i...j}(\boldsymbol{\xi})}{\partial \xi_i ... \partial \xi_j} \right| d\xi_i ... d\xi_j \tag{8.25}$$

where $Sd\epsilon_{i...j}(\text{¸})$ represents the previously defined surrogate model for increment function $i...j$ and ${}^{p}x_i$ represents the sample position, where both fidelity values (low and high) are known. Note that in Eq. (8.25), the surrogate model for the increment function is used instead of Eq. (8.24). This is possible due to the properties of the integral form of DE, namely Cond. 2.27.

One can realize that Eq. (8.25) resemble an increment function, where the central point is sample ${}^{p}\mathbf{x}$. However, the absolute value prevents direct transformation to the algebraic form. Unfortunately, direct application of Eq. (8.25) is not practical and it would lead to an extreme computational burden. Therefore, a trick is used to transform Eq. (8.25) into the increment function format. The absolute value is set in front of the definite integral, i.e.

$$ {}^{p}dSFEP_{i...j}(x_i, ..., x_j) = \left| \int_{{}^{p}x_i}^{x_i} ... \int_{{}^{p}x_j}^{x_j} \frac{\partial Sd\epsilon_{i...j}(\boldsymbol{\xi})}{\partial \xi_i ... \partial \xi_j} d\xi_i ... d\xi_j \right| \tag{8.26} $$

This modification is not mathematically correct, yet it is necessary to obtain fast solution. One can transform Eq. (8.26) into more appropriate shape, i.e.

$$ {}^{p}dSFEP_{i...j}(x_i, ..., x_j) = |dSd\epsilon_{i...j}(\mathbf{x})(x_i, ..., x_j)| \tag{8.27} $$

To obtain prediction in all directions, one needs to compute error increment in all directions. Using Eq. (8.20) and Eq. (8.27), one can construct the final solution for given sample as a sum of all increment predictions, i.e.

$$ \begin{aligned} {}^{p}FEP_t(x_i, ..., x_j) &= {}^{p}Sd\epsilon_t({}^{p}x_i, ..., {}^{p}x_j) + \sum_{i_2=i}^{j} {}^{p}dSFEP_{i_2}(x_{i_2}) \\ + \sum_{i \le i_2 < j_2 \le j} &{}^{p}dSFEP_{i_2 j_2}(x_{i_2}, x_{j_2}) + ... + {}^{p}dSFEP_{i...j}(x_i, ..., x_j) \end{aligned} \tag{8.28} $$

where $Sd\epsilon_t(x_i, ..., x_j)$ represents the surrogate model of FIEF for increment function t, ${}^{p}x_i$ represents the sample, where the value for the low fidelity and the high fidelity model is known and ${}^{p}SFEP_t(x_i, ..., x_j)$ represents the error prediction for given sample ${}^{p}\mathbf{x}$. Using the whole expansion (Eq. (8.28)) is still very expensive in the adaptive process and therefore, we restrict the error prediction only for the first order increment functions, i.e.

$$ {}^{p}FEP_t(x_i, ..., x_j) = {}^{p}Sd\epsilon_t({}^{p}x_i, ..., {}^{p}x_j) + \sum_{i_2=i}^{j} {}^{p}dSFEP_{i_2}(x_{i_2}) \tag{8.29} $$

In the previous parts, we worked with a predictive error for one sample ${}^{p}\mathbf{x}$. In other words, creating the prediction using only one known sample. However, the stochastic domain is composed of many samples, where the correction is known, i.e. samples used to construct surrogate model $Sd\epsilon_t(\mathbf{x})$. More importantly, one can realize that Eq. (8.28) is zero at sample ${}^{p}\mathbf{x}$. This is consistent with the theory as we want the prediction error to be zero at sample, where the correction is known. Therefore, one needs to construct Eq. (8.29) for all samples in given stochastic domain. The error prediction is done as a minimum of all neighbourhood samples, where only samples closest to the position of the prediction are considered. For example, for 1-D problems, the closest samples are one on the left and one on the right of the requested position of the prediction. At particular position, one can estimate the error as a minimum of error prediction from all neighbourhood samples. This is mathematically done in the following way

$$FEP_t(x_i,...,x_j) = min(\ ^{p_1}FEP_t(x_i,...,x_j),...,\ ^{p_{sn}}FEP_t(x_i,...,x_j)) \tag{8.30}$$

where superscript p_m represents sample $\mathbf{x}_m = [x_i,...,x_j]$, where Eq. (8.29) is constructed. For 1-D case, Eq. (8.30) can be simplified as follows

$$FEP_t(x_i) = min(\ ^{p_1}FEP_t(x_i),\ ^{p_2}FEP_t(x_i)) \tag{8.31}$$

where $^{p_1}FEP_t(x_i)$ represents the error prediction function for sample on the left and $^{p_2}FEP_t(x_i)$ represents the error prediction function for sample on the right. In higher dimensions, the tensor product sampling is adopted, which simplifies the sample neighbourhood approach. The example of a close neighbourhood approach for the tensor product sampling is showed in Fig. 8.2. Also, an illustrated example of Eq. (8.30) is provided for 1-D (Fig. 8.3) and for 2-D (Fig. 8.4).

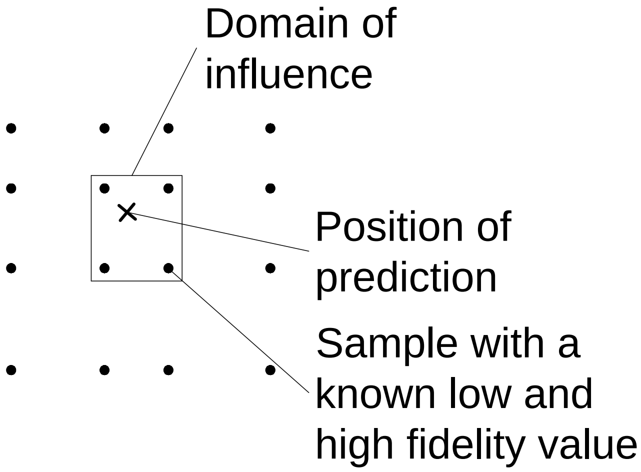


Figure 8.2: The tensor product sampling and the area of influence

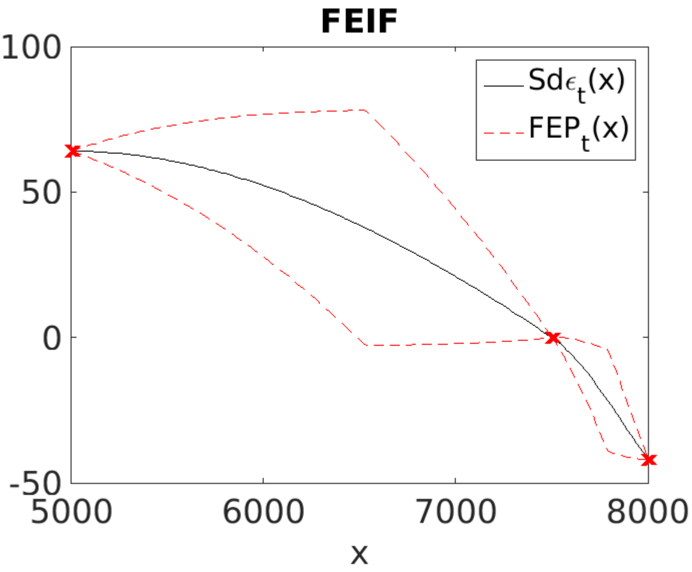


Figure 8.3: Example of 1-D prediction for FEIF

Note: In this 1-D example, 3 samples are considered, where the correction is known (${}^{p_1}x = 5000$, ${}^{p_2}x = 7500$, ${}^{p_3}x = 8000$). For the region between x=5000 and x=7500, $FEP_t(x)$ is constructed using ${}^{p_1}FEP_t(x)$ and ${}^{p_2}FEP_t(x)$. For the region between x=7500 and x=8000, $FEP_t(x)$ is constructed using ${}^{p_2}FEP_t(x)$ and ${}^{p_3}FEP_t(x)$.

Once the prediction model is created, one can establish the probable error of the mean value and the variance, respectively. Using Eq. (8.14) and Eq. (8.16), one can replace the fidelity error increment function with Eq. (8.30). This leads to the following predicted error of the mean value,

$$\mu_{FEP_t} \approx \frac{1}{z}\sum_{j=1}^{z} FEP_t(\mathbf{x}_j) \tag{8.32}$$

and the predicted error of the variance,

$$(\sigma_{FEP_t})^2 \approx \frac{1}{z-1}\sum_{j=1}^{z}(FEP_t(\mathbf{x}_j) - \mu_{FEP_t})^2 \tag{8.33}$$

where z represents the number of samples in MC simulation and t represents the selected increment function. The predicted mean value represents the probable error if the current model is used and the predicted variance represents the oscillation around the predicted mean, i.e. how much the error deviates. In other words, using the low fidelity model and the surrogate model for correction of the low fidelity model, the predicted error in the statistical properties is given by Eq. (8.32) and by Eq. (8.33).

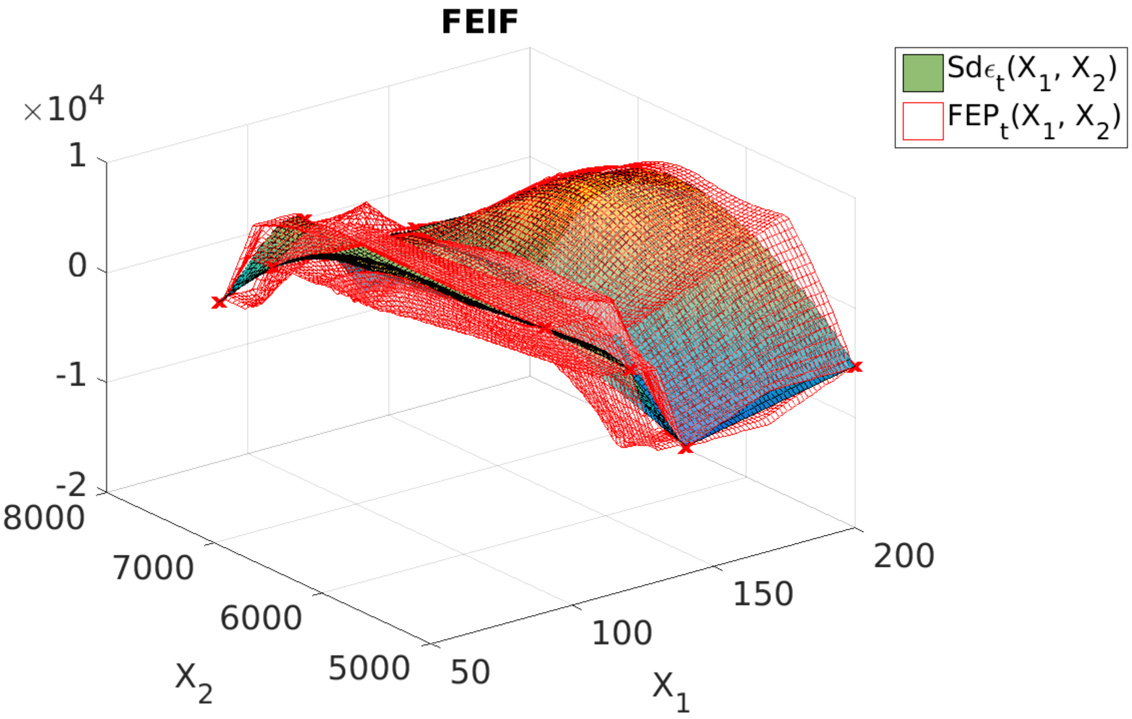


Figure 8.4: Example of 2-D prediction for FEIF

Note: In this 2-D example, 12 samples are considered, where the correction is known. For example, for the position of prediction at:
$x_1 = 160$ and $x_2 = 6000$;
The following samples and their corresponding FEP functions are used:
${}^{p_1}x_1 = 200,\ {}^{p_1}x_2 = 5000;\ {}^{p_2}x_1 = 140,\ {}^{p_2}x_2 = 5000;$
${}^{p_3}x_1 = 140,\ {}^{p_3}x_2 = 7500;\ {}^{p_4}x_1 = 200,\ {}^{p_4}x_2 = 7500;$

One can set a fidelity convergence criteria using above equations, but first it is necessary to establish the relative error for Eq. (8.32) and Eq. (8.33). The relative error for the predicted mean is defined in the following way

$$RE_{\mu_t} = \frac{\mu_{t_{FEP}}}{\mu} \tag{8.34}$$

where μ represents the total mean (see Eq. (3.19) and Eq. (5.20)) for given iteration. The relative error for the predicted variance is defined in the following way

$$RE_{(\sigma_t)^2} = \frac{(\sigma_{t_{FEP}})^2}{(\sigma)^2} \tag{8.35}$$

where $(\sigma)^2$ represents the total variance (see Eq. (3.20) and Eq. (5.21)) for given iteration. The relative error for the statistical properties represents a convergence criteria for the multi fidelity approach. One can compare Eq. (8.34) and

Eq. (8.35) to a pre-described residual and decide, if the correction function is accurate enough, i.e. if the correction function requires more high fidelity samples. However, this approach does not take into account position of samples. In other words, when two samples are close to each other, using the high fidelity model for both samples would require additional computational time with little gain in accuracy. Therefore, an empirical modification is introduced, which takes all this into account. This is topic of next section.

8.4.2 Application of the prediction approach to the fidelity modelling

The adaptive scheme defined in the previous chapters (see Chap. 3 and Chap. 5) sample the stochastic domain according to needs of the interpolation function. It takes into account position of the previous samples, the input distribution and complexity of the underlying function. However, it is optimized for the single fidelity approach. In other words, imagine a situation that the adaptive scheme proposes a sample, where the fidelity error function is very accurate. This would lead to a waste of the computational time and therefore, it is necessary to take into account accuracy of the fidelity model in given area. In other words, it is necessary to assess the accuracy of the correction function in the area of the proposed sample.

To take into account the accuracy of the correction function, one needs to modify Eq. (8.34) and Eq. (8.35) to take into account position of the proposed samples. In this work, an empirical approach is taken, where comparison of volumes is done. One can compute the overall area (volume) which is compared to the area (volume), where the adaptive scheme proposed a sample. The overall area (volume) is computed in the following way

$$S^{MF}_{T:t} = \int\limits_{R} FE_t(\mathbf{x})d\mathbf{x} \tag{8.36}$$

where R represents the entire domain of integration and t represents the selected increment function. The integration domain is defined by the input distributions (see Sec. 3.3). Next step represents the computation of a volume covered by the sample. This is mathematically done in the following way

$$S^{MF}_{S:t} = \int_{p_{x_i}}^{s_{x_i}} ... \int_{p_{x_j}}^{s_{x_j}} FE_t(x_i, ...x_j)dx_i...dx_j \tag{8.37}$$

where ${}^{p}x_i$ represents the sample, where correction is known, ${}^{S}x_i$ represents the proposed sample and t represents the selected increment function. The comparison of two volumes represents the magnitude of influence for given sample. Therefore, one can define the volume ratio, i.e.

$$S_t^{MF} = \frac{S_{S:t}^{MF}}{S_{T:t}^{MF}} \tag{8.38}$$

where t represents the selected increment function. Using the volume ratio (Eq. (8.38) and the relative error for the predicted mean/variance (Eq. (8.34)/Eq. (8.35)), one can establish an empirical equation for error estimation of the error mean/variance, when additional sample is corrected with the high fidelity solution. Mathematical formulation for the error estimation of the mean value reads

$$RE_{\mu_t:sample} = N_s S_t^{MF} RE_{\mu_t} \tag{8.39}$$

and for the variance reads

$$RE_{(\sigma_t)^2:sample} = N_s S_t^{MF} RE_{(\sigma_t)^2} \tag{8.40}$$

where RE_{μ_t} represents the relative error for the predicted mean (Eq. (8.34)), $RE_{(\sigma_t)^2}$ represents the relative error for the predicted variance (Eq. (8.35)), N_s represents the number of the low fidelity samples added in the current iteration and t represents the selected increment function. The number of low fidelity samples added in current iteration N_s represents an empirical approach, because it can happen that the additional samples are added in a way that Eq. (8.39) and Eq. (8.40) are bellow given threshold. However, the relative error for the predicted mean and the predicted variance is higher than given threshold and the correction function needs to be improved with additional high fidelity samples. One can understand Eq. (8.39) and Eq. (8.40) as error of given proposed sample, where in Eq. (8.34) and Eq. (8.35), the error of the whole domain is considered.

Once the relative error for each sample is defined, one can establish the selection criteria. In other words, one needs to select which samples are required for re-interpolation. This is done using the following re-interpolation criteria

$$\begin{gathered} if\ RE_{(\sigma_t)^2:sample} \geq R_{set_t} \vee RE_{\mu_t:sample} \geq R_{set_t} \\ \textit{Sample needs re-interpolation} \rightarrow \textit{Compute given sample with the high fidelity model} \end{gathered} \tag{8.41}$$

where t represents the selected increment function and R_{set_t} represents the residual for the local convergence process (see Eq. (3.25)). Above condition is tested for each proposed sample separately and if the condition is met for given sample, the high fidelity function value is obtained at the position of given sample. When the sample is obtained, one can construct the correction at given sample and improve the accuracy of the correction surrogate model $Sd\epsilon_t(\mathbf{x})$.

8.5 Numerical application of the multi fidelity modelling

The numerical application of the proposed approach is relatively straightforward. It represents a modification of the proposed algorithm in the previous chapters to implement the multi fidelity scheme. However, there are some difficult parts in the process.

The first part represents the central sample. It is necessary to obtain the central sample for the high fidelity model and for all the low fidelity models. Also, it is necessary to obtain the boundary samples for the first order increment functions using the high fidelity model and all the low fidelity models. Once the samples are obtained, the algorithm can estimate the power ratio, P^{MF}, and construct FEIF for each variable. Following this, the algorithm can estimate the stochastic influence of given increment functions, i.e. the partial mean value and the partial variance. Next, the algorithm can start the adaptive scheme for the first order increment functions (see Chap. 3) and propose new samples. Application of the multi fidelity scheme for the proposed sample is the topic of the next section.

8.5.1 Application to the first order increment functions

Once the samples are proposed and surrogate model $Sd\epsilon_t(\mathbf{x})$ is constructed, the sensitivity of the first order increment functions can be established. As mentioned in Sec. 8.4, one can use various surrogate techniques, but it is suggested to use the piecewise interpolation, which is not prone to divergence. Surrogate model $Sd\epsilon_t(\mathbf{x})$ is also, used for the prediction of the fidelity error. For every sample, where the correction is known, the algorithm needs to construct Eq. (8.29). Once the prediction functions are established, the algorithm can use Eq. (8.31) to establish the predicted fidelity error over given stochastic domain.

In the next step, the algorithm needs to establish the partial stochastic values, i.e. the mean value of the predicted error (Eq. (8.32)) and the variance of the predicted error (Eq. (8.33)). These values are established propagating the same matrix as for the MC simulation (see Chap. 3) through the given surrogate model. The mean value of the predicted error and the variance of the predicted error are used to establish the relative errors (Eq. (8.34) and Eq. (8.35)) for given increment function. Note that for the first order increment functions, the total variance and the total mean established at the current iteration are used.

In the next step, the algorithm must establish the overall area (volume) of the prediction error (Eq. (8.36)), where the area (volume) is given by the minimum and the maximum of the input distributions (see Sec. (3.3)). For each proposed sample by the adaptive scheme, the algorithm needs to establish the area (volume) between the proposed sample and the sample, where the correction is known (Eq. (8.37)). Next step represents the estimation of the error of the mean (Eq. (8.39)) and the error of the variance (Eq. (8.40)) for each sample proposed in the current iteration. After obtaining the errors for all samples, it is necessary to test if the accuracy of a given sample is sufficient, which is done using Cond. 8.41. In the case of insufficient accuracy, the algorithm needs to re-interpolate sample with the high fidelity model, i.e. obtain the high fidelity value for given sample and construct a new surrogate model, $Sd\epsilon_t(\mathbf{x})$.

In the next step, it is necessary to select the low fidelity model, which should be used for the proposed sample. The selection of the low fidelity model is based on the power ratio, i.e. application of the relative predicted fidelity error (Eq. (8.34) and Eq. (8.35)) to the power ratio (Eq. (8.1)). This is mathematically done in the following way

$$P^{MF}_{\mu_t} = \frac{e^{-(RE_{\mu_t})}}{T^{MF}} \tag{8.42}$$

$$P^{MF}_{(\sigma_t)^2} = \frac{e^{-(RE_{(\sigma_t)^2})}}{T^{MF}} \tag{8.43}$$

where T^{MF} represents the time ratio for given low fidelity model and t represents the selected increment function. Note that Eq. (8.42) and Eq. (8.43) have to be higher than 1 (see Cond. 8.2) in order to consider the low fidelity model efficient. The low fidelity model with the highest power ratio is used for given sample and when the surrogate model is selected, the algorithm can return to the re-interpolation aspect. In a case that all the low fidelity models have the power ratio lower than 1, only the high fidelity model is used. The proposed multi fidelity modification for the global process to the first order adaptive scheme is showed in Alg. 12 and in Alg. 13 is showed the multi fidelity scheme for the local convergence.

Algorithm 12 Multi fidelity global process

```
Initialize the process:
1. - 5. Same steps as in Alg. 3
for $i^{MF} = 1$ to number of the low fidelity models do
    6. Compute the central point $f({}^{c}\mathbf{x})$ for the $i^{MF}$-th low fidelity model and
    set the iteration to 0, i.e. $k = 0$
end for
7. - 8. Same steps as in Alg. 3
for $i^{MF} = 1$ to number of the low fidelity models do
    9. Sample the boundaries of the stochastic space using the $i^{MF}$-th low
    fidelity model and set the iteration to 1, i.e. $k = 1$
end for
while $GR_{\mu} > GR_{set}^{\mu}$ or $GR_{(\sigma)^2} > GR_{set}^{(\sigma)^2}$ do
    1. Call Alg. 13
    2. Call Alg. 2
end while
Finalize the process:
1. - 2. Same steps as in Alg. 3
End
```

Algorithm 13 Multi fidelity scheme for the local convergence - Part 1

```
Initialize the process:
1. - 4. Same steps as in Alg. 1
while set $ST$ contains non-converged increment functions do
    for $t = 1$ to length of set $ST$ do
        1. - 7. Same steps as in Alg. 1
    end for
    1. - 3. Same steps as in Alg. 1
    if Cond. 8.19 hold then
        for $t = 1$ to length of set $ST$ do
            for $i^{MF} = 1$ to number of the low fidelity models do
                1. Select samples common to the low/high fidelity model and create sample increments (Example Eq. (8.23))
                2. Construct the surrogate model, $Sd\epsilon_k(\mathbf{x})$
                3. Construct the Fidelity Error (Eq. (8.31))
                4. Compute the statistical properties for the Fidelity Error (Eq. (8.32) and Eq. (8.33)) and the relative errors for the predicted statistical values (Eq. (8.34) and Eq. (8.35))
                5. Compute the area of the Fidelity Error (Eq. (8.36))
                6. Compute the area of the proposed sample (Eq. (8.37))
                7. Compute the sample error prediction (Eq. (8.39) and Eq. (8.40))
                8. Compute the power ratios (Eq. (8.42) and Eq. (8.43))
            end for
            1. Select the low fidelity model with the best power ratio and sample given increment functions
            if Cond. 8.41 hold then
                1. Re-interpolate the sample with the high fidelity model
            end if
        end for
    end if
```

8.5.2 Application to the higher order increment functions

The application to the higher order increment functions is slightly different, yet again straightforward. First, it is necessary to establish which increment functions

Algorithm 13 Multi fidelity scheme for the local convergence - part 2

```
        if $k \geq 2$ then
            1. Same step as in Alg. 1
            for $t = 1$ to length of set $ST$ do
                if Cond. 3.25 hold then
                    1. Same step as in Alg. 1
                end if
            end for
            2. $k = k + 1$
        else
            1. $k = k + 1$
        end if
    end while
    Finalize the process:
    1. - 3. Same steps as in Alg. 1
    End
```

can be approximated by the low fidelity model. Therefore, a direct application of the prediction algorithm (Chap. 4) to all the low fidelity models and also, to the high fidelity model is necessary. Note that the prediction algorithm must use the same test samples (see Sec. 4.2.1) for all the fidelity models. Once, the important increment functions are obtained for all the low fidelity models, the algorithm can create a set of increment functions, which can be approximated with the low fidelity model (Eq. (8.17)). It is necessary to apply this approach, when an additional higher order increment functions are considered, e.g. changing from the second order to the third order increment functions etc...

When the algorithm selects the increment functions, which can be approximated with the low fidelity model, the boundaries of the stochastic domains can be sampled. Note that the low fidelity model is assumed to have the same trends (see earlier in Sec. 8.4) as the high fidelity model. The algorithm selects the low fidelity model with the highest power ratio (Eq. (8.18)). In a case that all the low fidelity models have power ratio lower than 1, only the high fidelity model is used. Using the selected low fidelity model, the algorithm can sample the boundaries of the stochastic domain. Once the boundaries are established and the interpolation process of given increment function is done, the algorithm can establish the sensitivities of given stochastic domain. Next step is the application of Cond. 8.19, which select the important increment functions for re-interpolation, i.e. correction with the high fidelity model.

The increment functions selected for re-interpolation needs to be sampled with the high fidelity model. It is necessary to stabilize the interpolation model and therefore, the algorithm needs to sample all the boundary samples with the high fidelity model. Once the boundary samples are obtained, the algorithm can construct the higher order correction surrogate, $Sd\epsilon_k(\mathbf{x})$, using the preselected surrogate technique, e.g. Lagrange interpolation.

Next steps are essentially the same as in the case of the first order increment functions. For every sample, where both functional values (the low fidelity model and the high fidelity model) are known, the algorithm needs to construct Eq. (8.29). Once the prediction functions are established, the algorithm can use Eq. (8.31) to establish the predicted fidelity error over given stochastic domain. The partial stochastic values are obtained using Eq. (8.32) and Eq. (8.33) and with application of MC simulation. The mean value of the predicted error and the variance of the predicted error are used to establish the relative errors (Eq. (8.34) and Eq. (8.35)) for given increment function.

The overall volume is established using Eq. (8.36), where the boundaries of the stochastic domain are given by the input distributions (see Sec. (3.3)). The area between known sample and the proposed sample is obtained using Eq. (8.37). The algorithm needs to estimate the error for the mean (Eq. (8.39)) and for the variance (Eq. (8.40)) for each sample proposed in the current iteration. Using Cond. 8.41, the algorithm select samples, which needs to be re-interpolated. However, there is a difference between the 1-D approach and N-D approach. For the higher order increment functions, it is necessary to fulfil the requirements of the surrogate model used for FEIF. For example, Lagrange interpolation requires a tensor product sampling in order to stabilize the interpolation process. Therefore, if the algorithm selects a sample in the stochastic domain, it automatically selects for re-interpolation also samples to fulfil the conditions required by the interpolation technique. In this work, the tensor product sampling is used in all cases (see Chap. 5), which simplifies the problem.

As in the case of the first order increment functions, the selection process for the low fidelity model is based on its power ratio. The power ratio is established for all considered low fidelity models using Eq. (8.42) and Eq. (8.43). Note that if the power ratio is under 1, the low fidelity model is automatically discarded from the process (see Sec. 8.2). The global process for the multi fidelity higher order increment functions is showed in Alg. 14 and the local convergence process for the higher order multi fidelity algorithm is showed in Alg. 15

Algorithm 14 Global process for the multi fidelity higher order increment functions - part 1

```
Initialize the process:
1. - 2. Same steps as in Alg. 6
3. Predict the important second order increment functions using the predictor scheme (Chap. 4) for the high fidelity model
for $i^{MF} = 1$ to number of the low fidelity models do
    1. Predict the important second order increment functions using the predictor scheme (Chap. 4) for the $i^{MF}$-th low fidelity model
end for
4. Select the low fidelity model with the highest power ratio (Eq. (8.18)) for each considered increment function
5. Assign the selected second order increment functions to set $MD - ST$
if $MD - ST$ is not empty then
    while Stopping criteria not met do
        1. - 3. Same steps as in Alg. 6
        for $t = 1$ to length of set $MD - ST$ do
            1. Same step as in Alg. 6
            2. Construct the surrogate model, $SdF_t^0(\mathbf{x})$, for the increment function, $dF_t$, using the selected low fidelity model
            3. Same step as in Alg. 6
            if Cond. 8.19 holds then
                1. Sample the boundaries of given increment function, $dF_t$, with the high fidelity model
                2. Construct the error increments (Example Eq. (8.23)) and create the surrogate model, $Sd\epsilon_t(\mathbf{x})$
                3. Correct the low fidelity values with the surrogate model, $Sd\epsilon_t(\mathbf{x})$
            end if
            4. - 7. Same steps as in Alg. 6
            if Cond. 5.26 hold then
                1. Same step as in Alg. 6
            end if
        end for
```

Algorithm 14 Global process for the multi fidelity higher order increment functions - part 2

```
        while $GR_\mu > GR^\mu_{set}$ or $GR_{(\sigma)^2} > GR^{(\sigma)^2}_{set}$ do
            1. Call Alg. 15
            2. Call Alg. 5
        end while
        4. Same step as in Alg. 6
        5. Predict the important higher order increment functions using the
        predictor scheme (Chap. 4) for the high fidelity model
        for $i^{MF} = 1$ to number of the low fidelity models do
            1. Predict the important higher order increment functions using the
            predictor scheme (Chap. 4) for the $i^{MF}$-th low fidelity model
        end for
    end while
end if
Finalize the process:
1. - 2. Same steps as in Alg. 6
End
```

Algorithm 15 Scheme for the multi fidelity high order local convergence - part 1

```
Initialize the process:
1. - 4. Same steps as in Alg. 4
while set $MD - ST$ contains non-converged increment functions do
    for $t = 1$ to length of set $MD - ST$ do
        1. - 2. Same steps as in Alg. 4
        if $V_{i...j}$ contains samples then
            1. Same step as in Alg. 4
        else
            1. - 2. Same steps as in Alg. 4
        end if
        3. - 10. Same steps as in Alg. 4
        if Cond. 5.26 hold then
            1. Same step as in Alg. 4
        end if
    end for
```

Algorithm 15 Scheme for the multi fidelity high order local convergence - part 2

```
    1. Same step as in Alg. 4
    for $t = 1$ to the length of the set $MD - ST$ do
      if Cond. 8.19 hold then
        for $i^{MF} = 1$ to number of the low fidelity models do
          1. Update/Construct the surrogate model, $Sd\epsilon_t(\mathbf{x})$
          2. Construct the Fidelity Error (Eq. (8.31))
          3. Compute the statistical properties for the fidelity error (Eq. (8.32) and Eq. (8.33)) and the relative errors for the predicted statistical values (Eq. (8.32) and Eq. (8.33))
          4. Compute the area of the fidelity error (Eq. (8.36))
          5. Compute the area of the proposed samples (Eq. (8.37))
          6. Compute the sample error prediction (Eq. (8.39) and Eq. (8.40))
          7. Compute the power ratios (Eq. (8.42) and Eq. (8.43))
        end for
        for $j = 1$ to the number of proposed samples do
          1. Select the low fidelity model with the best power ratio and sample given increment functions
          if Cond. 8.41 hold then
            1. Re-interpolate the sample with the high fidelity model
            2. Fulfil all necessary interpolation conditions for surrogate model $Sd\epsilon_t(\mathbf{x})$, i.e. obtain other samples if necessary
          end if
        end for
      end if
    end for
    2. Obtain all the proposed samples from the selected models, $f^{HF}(\mathbf{x})$ and $f^{LF}(\mathbf{x})$
    3. $k = k + 1$
  end while
  Finalize the process:
  1. - 2. Same steps as in Alg. 4
  3. Store surrogate model $Sd\epsilon_t(\mathbf{x})$ for all the increment functions and all the low fidelity models
  End
```

8.6 Applied example for the multi fidelity approach

The function selected for the multi fidelity approach is the well known Borehole problem. However, in this case, multiple models defined in the work of Xiong et al. (141) are used. The high fidelity model is the same as used in Sec. 5.5.2 and it is defined as follows

$$F^H(x) = \frac{2\pi T_u(H_u - H_l)}{ln\frac{r}{r_w}(1 + \frac{2LT_u}{ln\frac{r}{r_w}r_w^2 K_w} + \frac{T_u}{T_l})} \tag{8.44}$$

where r_w represents the radius of borehole (m), r represents the radius of influence (m), T_u represents the transmissivity of upper aquifer (m^2/yr), H_u represents the potentiometric head of upper aquifer (m), T_l represents the transmissivity of lower aquifer (m^2/yr), H_l represents the potentiometric head of lower aquifer (m), L represents the length of borehole (m) and K_w represents the hydraulic conductivity of borehole (m/yr). The output is water flow rate in m^3/yr.

The low fidelity model is defined as follows

$$F^L(x) = \frac{5T_u(H_u - H_l)}{ln\frac{r}{r_w}(1.5 + \frac{2LT_u}{ln\frac{r}{r_w}r_w^2 K_w} + \frac{T_u}{T_l})} \tag{8.45}$$

where r_w represents the radius of borehole (m), r represents the radius of influence (m), T_u represents the transmissivity of upper aquifer (m^2/yr), H_u represents the potentiometric head of upper aquifer (m), T_l represents the transmissivity of lower aquifer (m^2/yr), H_l represents the potentiometric head of lower aquifer (m), L represents the length of borehole (m) and K_w represents the hydraulic conductivity of borehole (m/yr). The output is water flow rate in m^3/yr. Distributions associated with each random variable are summarized in Tab. 8.1.

The comparison is done with MC simulation. The selected surrogate model for the correction function is the spline interpolation for the first order increment functions and IPI for the higher order increment functions. MC simulation is applied directly to the high fidelity model (Eq. (8.44)) and the results of MC simulation are summarized in Tab. 8.2. The convergence of the mean value and the standard deviation for the multi fidelity approach is summarized in Tab. 8.3 and the number of functions calls and also, the computational time is summarized in Tab. 8.4. The selected partial mean values and the partial variances for the multi fidelity approach are summarized in Tab. 8.6 and their partial histograms are shown in Fig. 8.7.

ID	Random Variable	Distribution type	Mean	Standard deviation
x_1	r_w	Normal	0.10	0.0161812
x_2	r	Log-Normal	7.71	1.0056
			Min	Max
x_3	T_u	Uniform	63070	115600
x_4	H_u	Uniform	990	1110
x_5	T_l	Uniform	63.1	11.6
x_6	H_l	Uniform	700	820
x_7	L	Uniform	1120	1680
x_8	K_w	Uniform	9855	12045

Table 8.1: Distributions for the Borehole model

Function calls	Mean	Standard deviation
100000	73.84	28.71

Table 8.2: MC simulation for the Borehole model

Case	Desired accuracy	Function calls	Mean	Standard Deviation	Relative error of Mean	Relative error of S. D.
1	0.01	-	73.84	28.71	5.017e-06	1.000e-03
2	0.001	-	73.84	28.71	1.713e-05	6.701e-04

Table 8.3: Results of the high dimensional multi fidelity adaptive UQ-HDMR approach

8.6.1 Discussion about the applied example

The uncertainty quantification is performed on the Borehole problem using the multi fidelity approach. Same as in the single fidelity case, the proposed multi fidelity approach successfully converged for a low number of samples and provided a very accurate representation of PDF, leading to small errors between MC and the proposed method. The right tail of the final distribution is well described for both residuals (case 1: $1e-2$, case 2: $1e-3$). The list of all sampled increment functions is showed in Appendix G.

The number of simulations required for the high fidelity and the low fidelity code is summarized in Tab. 8.4, where the number of the high fidelity calls is in

Case	Model	Function calls	Time of one simulation [s]
1	High Fidelity (Eq. (8.44))	40	0.0112
1	Low Fidelity (Eq. (8.45))	88	3e-05
2	High Fidelity (Eq. (8.44))	59	0.0112
2	Low Fidelity (Eq. (8.45))	126	3e-05

Table 8.4: Number of function calls and computational times for the multi fidelity approach

Note: Considered high fidelity model (Eq. (8.44)) is artificially slowed down in order to test the performance of the method.

Case	Desired accuracy	Function calls	Mean	Standard Deviation	Relative error of Mean	Relative error of S. D.
1	0.01	87	73.85	28.70	1.821e-04	4.736e-4
2	0.001	126	73.84	28.70	2.977e-05	3.198e-4

Table 8.5: Results of the high dimensional single fidelity adaptive UQ-HDMR approach

Increment function	Partial Mean	Partial Variance	Mean Sensitivity	Variance Sensitivity
1	1.8174	531.4032	0.6402	0.6716
6	-0.0258	72.1023	0.0090	0.0911
7	0.9177	71.4914	0.3230	0.0904
8	0.0030	16.7087	0.0010	0.0211
1.8	0.0045	1.7665	0.0015	0.0022
1.7	0.0150	7.6117	0.0053	0.0096
1.6	0.0069	7.5866	0.0024	0.0096
1.4	0.0128	7.5985	0.0045	0.0096

Table 8.6: Results of the selected increment functions for the Borehole problem

both cases lower than in the single fidelity approach (see Tab. 8.5). However, the important part is the overall time. For the single fidelity approach with requested relative accuracy 0.01, the required computational is 0.97 [seconds], where for the multi fidelity approach the total required time is 0.45 [seconds]. There is more than twice gain in the computational time[1]. For the single fidelity approach

[1] Only computational time of the low fidelity model and the high fidelity model is considered. The processing time of the adaptive algorithm is not counted.

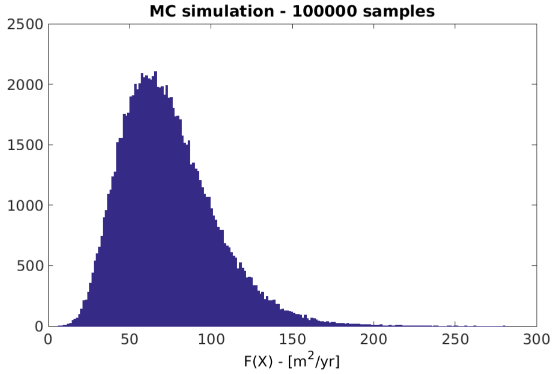


Figure 8.5: PDF obtained by the Monte Carlo simulation

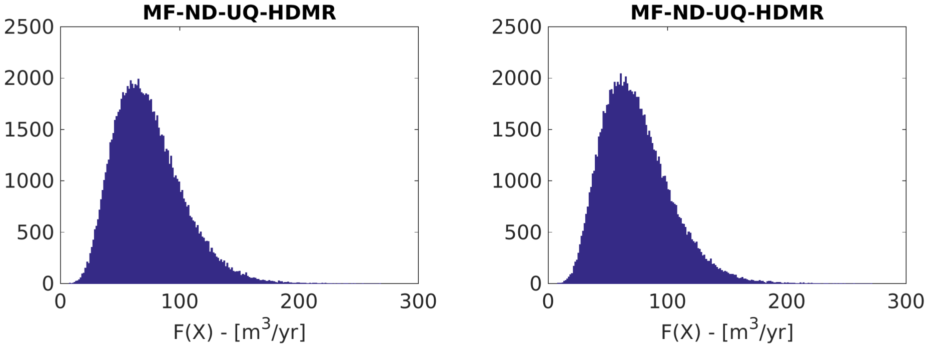
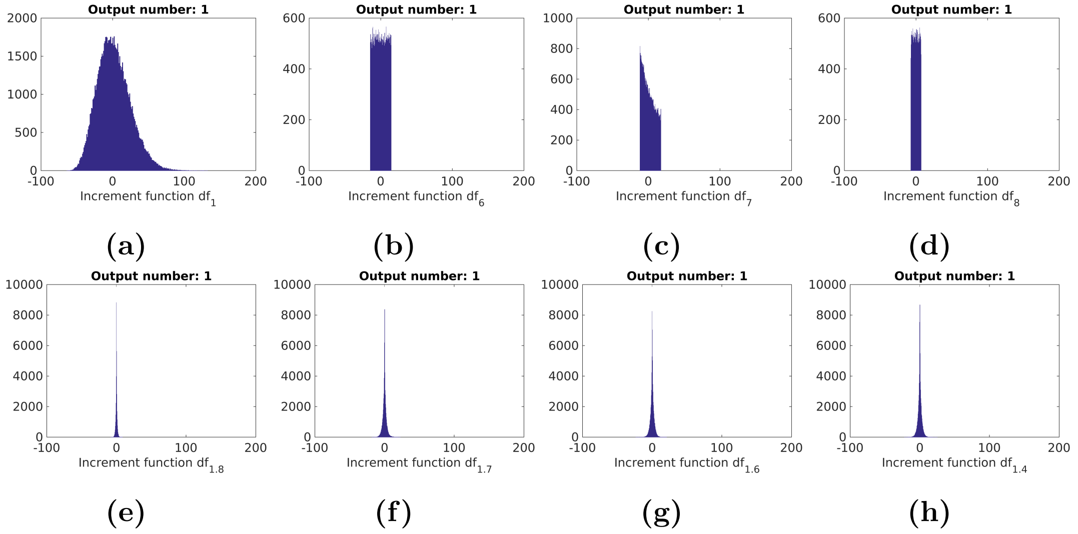


Figure 8.6: PDF obtained by the multi fidelity high dimensional adaptive UQ-HDMR
Left: Relative Accuracy 0.01 Right: Relative Accuracy 0.001

with requested relative accuracy 0.001, the required computational time is 1.41 [seconds], where for the multi fidelity approach the total required time is 0.66 [seconds]. Again the reduction of the computational time is around 2.

The interesting part is the accuracy of the mean, which is higher for the multi fidelity case. However, the difference is negligible for the real application and it can be attributed to the properties of the surrogate models, which are used for the interpolation. Regarding the error of the standard deviation, for both residuals (case 1: $1e-2$, case 2: $1e-3$), the multi surrogate approach is slightly worse than the single fidelity approach. This is the expected result, however, the multi fidelity approach in both cases provided requested accuracy. Nevertheless, this represents a standard trade-off between gained computational time and lost accuracy. Also, a number of the high fidelity samples does not grow so fast as in the case of the single fidelity. This can be attributed to the accuracy of the



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 8.7: Partial histograms for the selected increment functions of the Borehole problem using the multi fidelity approach

correction function. The more samples the increment function requires to fulfil the convergence criteria, the more accurate and useful the correction function is. In other words, when additional samples are added to the stochastic domain, the correction function becomes more accurate. Therefore, the correction function can be used more often and the algorithm does not have to call the high fidelity model.

The partial histograms (Fig. 8.7 for the multi fidelity approach and Fig. 5.8 for the single fidelity approach) are also similar and therefore, the multi fidelity approach can be used to estimate the partial histograms. However, using the partial values without corrections can lead to dramatic errors. Therefore, it is not suggested to use partial histograms obtained only with the low fidelity model.

8.7 Discussion about the multi fidelity scheme

In this chapter, the multi fidelity approach is introduced. It is considered a separate module for the commonly used single fidelity approach. Therefore, the proposed method can handle both approaches - single and multi. The multi fidelity approach takes into account time and accuracy of given low fidelity models, which is not considered in many previous works.

The power ratio defined is Sec. 8.2 represents a selection approach for the low fidelity models. It provides a comparison criterion, which takes into account the accuracy and timing of the low fidelity model. Based on the comparison, the algorithm can select the best model for given increment function. However, the important aspect of the power ratio is that it can show the efficiency of the low fidelity model, i.e. if the low fidelity model can reduce the computational burden and provide accurate results.

In this work, the quantity of interest for the selection process is considered the predicted error of the mean value and variance, respectively. However, the concept is general and the quantity of interest could be any type of accuracy, e.g. $L2$ norm.

The multi fidelity approach is separated into two schemes: the first order increment functions and the higher order increment functions. A priori, there is no sufficient knowledge about the first order increment functions and therefore, for each first order increment function, boundary samples are obtained in order to obtain the basic knowledge about the problem. However, for the higher order increment functions, one can deduce a large portion of information based on the derivative equation (Eq. (2.5)) and it is a waste of the high fidelity samples to correct all the increment functions. Therefore, a selection process is introduced. The selection process includes selection approach (see Chap. 4), which allows selecting the important increment functions. The predicted increment functions can be compared and proper model can be assigned to each increment function. This represents an important aspect as an empirical experience shows us that many low fidelity models do not include interaction effects in the final process.

Once, the selection approach is performed, the important higher order increment functions are corrected. It is necessary to sample the boundaries of the given increment function and stabilize the interpolant. Many interpolants (Lagrange, IPI etc...) have troubles with extrapolation and this can lead to extreme errors. This aspect is crucial in higher dimensions (e.g. > 2), where one cannot track the behaviour of the surrogate model.

The correction surrogate model, $Sd\epsilon_k(\mathbf{x})$, represents an important aspect in the multi fidelity modelling. The surrogate model is created with samples, where the functional values for both models (low and high fidelity) are known. The correction can be additive or multiplicative, however, in this work, only additive correction is used. The additive correction can be easily used in the integration theory, which is part of the statistical analysis, i.e. computation of the mean value and the variance, respectively. Moreover, the modification of the increment function approach is straightforward with the additive correction. Another aspect represents the accuracy of given correction surrogate model. The errors from the correction model propagate to the final surrogate model and influence the final solution. Therefore, using a global polynomial function for the correction surrogate model is not suggested as polynomials are prone to oscillations.

The prediction of errors for the fidelity correction surrogate model is based on the linear prediction, i.e. using the difference between derivatives at given point and linearly approximate the difference in a close neighbourhood. However, the linear approach is modified to use the surrogate model, $Sd\epsilon_k(\mathbf{x})$, which provides more accurate results. This approach is later on extended to take into account other samples, where the correction is known. The fidelity predicted error function (Eq. (8.30)) represents the predicted error between two models, which differs in computational time and accuracy. The fidelity predicted error function is simplified to use only the first order increment functions (see Eq. (8.29)), which introduce a source of errors into prediction. However, if one considers the computational time gained with the simplification, it is a good trade-off between speed and accuracy. Note that the fidelity error prediction does not influence the final result and it is some sort of the expected error (not a real error). Therefore, it can be taken as only informative.

The overall prediction does not take into account position of other known samples. Therefore, the error estimation is modified for each sample and estimation of volumes is used. One can assume that the influenced volume of the error prediction function represents the change of the statistical value, i.e. the partial mean and the partial variance. In other words, how the additional sample influences the statistical properties. This leads to the proposed more efficient approach, where a position of the proposed sample is considered. To illustrate the problem, consider a sample in the region of good fidelity agreement. The low fidelity model is accurate enough and correction would bring only computational waste. Contrary, if the sample lies in a region of large fidelity disagreement, the high fidelity model must correct the low fidelity one. This approach reflects the regional accuracy of the low fidelity model against the high fidelity model.

8.8 Conclusion

In this chapter, the multi fidelity module is introduced. This module represents an additional approach to the high dimensional uncertainty propagation when multiple models with different fidelity are available. However, there is involved a trade-off between the gain of computational speed and loss of accuracy.

In order to create comparison criteria for selection of the low fidelity models, the power ratio theory introduced. The power ratio takes into account the computational time and accuracy of given low fidelity model and such that creates a comparable criterion, which allows selecting the most efficient low fidelity model. In our case, the accuracy is measured as the difference between the mean value and the standard deviation for the low and high fidelity model. However, the nature of the approach allows measuring the accuracy in different ways such as L2 norm. A very important aspect of the power ratio theory represents the efficiency line, which separates the effective low fidelity models from the ineffective ones. The effective low fidelity models can be used in the multi fidelity modelling and some gain can be achieved, but the ineffective models do not provide marginal improvement and such that their usage is not beneficial.

The multi fidelity correction is based on the additive correction, which is implemented into the HDMR approach. The additive correction is selected due to its simplicity and easy modification to the HDMR approach. The correction function is then constructed as a surrogate model using samples, where the correction is known. Observing the partial statistical values of the correction function gives an interesting insight into the difference between fidelity models.

The correction function is used to develop the prediction approach, which helps to estimate the accuracy of the selected low fidelity model. The basic idea is to use the linear approximation to predict the error at the uncorrected locations. Using this approach to predict the error in the partial mean and the partial standard deviation helps to decide if the correction function needs additional samples to improve its accuracy. Moreover, plotting the prediction function allows deeper insight into the regional accuracy of given low fidelity model. This helps the user to better understand the differences between given models.

Important aspect represents the difference between the first order increment functions and the higher order increment functions. For the first order increment functions, nothing can be assumed about the low fidelity model and such that it is necessary to obtain boundary samples for each first order increment function. This helps to establish the power ratio for given low fidelity model and decide if the low fidelity model is efficient or not. Contrary, this restricts the number of low fidelity models to a low number as using a large pool of the low fidelity models only prolong the computational time and beneficial gain is lost. For the high order increment functions, correcting all the increment functions would lead to extreme cost and such that only the important increment functions are corrected. To find, which increment functions are important, the selection scheme is used. The selection scheme is applied to each low fidelity model and this represents another reason to keep the number of the low fidelity models low.

The multi fidelity module is tested on the well known Borehole problem, which is introduced in Chap. 5. The results are compared to MC simulation and the savings in time are compared to the single fidelity approach. In this deterministic case, the advantages of the multi fidelity approach are shown.

Chapter 9

Applied example: Probabilistic modelling of debris re-entry

In this chapter, the developed single fidelity uncertainty quantification method is applied to the re-entry problem. We consider a spherical object, which is falling from the low earth orbit (120 km) down to the Earth ground. The observed criterion is the location of impact on the ground, i.e. the longitudinal and lateral position. Three test cases are considered, which differs in the flight path angle. Interesting conclusions are obtained for the re-entry cases and results are discussed in detail. A discussion is mainly aimed at the efficiency and accuracy of the proposed method.

The chapter is structured as follows: In the first section, the re-entry simulation tool is described. The second section is given to a description of the uncertain input parameters and its necessities. The results are discussed in the third section and the last section is given to the conclusion.

9.1 Debris re-entry propagation

Space Situational Awareness (SSA) is quickly becoming imperative for nations around the world, especially those with space capabilities. As the low Earth orbit debris and spacecraft population that has exceeded their operational lifetime rise each year, the rate at which objects re-enter the Earth atmosphere will also steadily rise. Most of these objects will probably not reach the ground for impact; however, parts of large objects like rocket bodies and satellites or resident space objects with a mass greater than a ton have a high probability of surviving the harsh re-entry environment. The surviving parts can be hazardous (e.g. fuel tanks with unused hydrazine or radioactive components) and can cause damage and casualties within a populated area. The impact location of an object re-entering the atmosphere can be altered by uncertainties in initial conditions, atmospheric characteristics, and object properties, as well as break-up/fragmentation events.

For the re-entry simulation, the Free Open Source Tool for Re-entry of Asteroids and Space Debris (FOSTRAD) is used. The goal of FOSTRAD is to have a high fidelity code, which will avoid the complexity of a spacecraft-oriented approach. The framework for FOSTRAD is modular in nature, where work on modules is done in the work of Mehta et al. (2, 161, 162).

9.1.1 Trajectory Dynamics

The atmospheric entry simulation is set to begin at an altitude of 120 km. A simple spherical object re-entering the Earth's atmosphere is modelled as a point mass and tracked through the atmosphere down to ground. The dynamics of the object is governed by the following system of differential equations:

$$\dot{V} = V_\infty \sin(\gamma) \tag{9.1}$$

$$\dot{V_\infty} = -\frac{D}{m} - g\sin(\gamma) + \omega_E^2 (R_E + h)\cos(\phi)(\sin(\gamma)\cos(\phi) - \cos(\phi)\cos(\chi)\sin(\phi)) \tag{9.2}$$

$$\begin{aligned}\dot{\gamma} = &\left(\frac{V_\infty}{R_E + h} - \frac{g}{V_\infty}\right)\cos(\gamma) + 2\omega_E \sin(\chi)\cos(\phi) + \\ &+ \omega_E^2 \frac{R_E + h}{V_\infty}\cos(\phi)(\cos(\chi)\sin(\gamma)\sin(\phi) + \cos(\gamma)\cos(\phi))\end{aligned} \tag{9.3}$$

$$\dot{\chi} = -\left(\frac{V_\infty}{R_E + h}\right)\cos(\phi)\sin(\chi)tan(\phi) \tag{9.4}$$

$$\dot{\phi} = \left(\frac{V_\infty}{R_E + h}\right)\cos(\gamma)\cos(\chi) \tag{9.5}$$

$$\dot{\lambda} = \left(\frac{V_\infty}{R_E + h}\right)\frac{\cos(\gamma)\cos(\chi)}{\cos(\phi)} \tag{9.6}$$

where h represents the altitude, V_∞ represents the speed of the object, γ represents the flight path angle, D represents the drag force, g represents the gravitational acceleration, ω_E represents the Earth's rotational speed, R_E represents the radius of the Earth, γ represents the path direction angle, χ represents the latitude and λ represents the longitude. The gravitational acceleration is modelled as a function of the altitude given as:

$$g = g_0 \left(\frac{h}{R_E + h} \right)^2 \tag{9.7}$$

where g_0 represents the gravitational constant set to $9.81ms^2$. The integration scheme used is the Runge Kutta method (163).

9.1.2 Aerodynamics

The object of interest is modelled as a sphere with triangular facets. The pressure and shear contribution are modelled independently as a function of local flow with respect to inclination angle. The axial and normal force coefficients are computed as integrals of the pressure and shear distributions over the surface.

9.1.2.1 Continuum flow regime

The aerodynamic contribution in the continuum flow regime is computed using the modified Newtonian theory, which reads:

$$C_p = C_{p,max} \sin^2(\theta) \tag{9.8}$$

where C_p represents the local pressure coefficient and θ represents the maximum pressure point coefficient. The shear contribution in the continuum regime is assumed to be 0.

9.1.2.2 Free molecular flow regime

The aerodynamic contribution of each facet in the free molecular regime is computed using Schaaf and Chambres (164) closed-form analytic model, which reads

$$\begin{aligned} C_p = \frac{1}{s^2} \Bigg[& \left(\frac{2 - \sigma_N}{\sqrt{\pi}} s \sin(\theta) + \frac{\sigma_N}{2} \sqrt{\frac{T_w}{T_\infty}} \right) e^{-(s \sin(\theta))^2} + \\ & \left\{ (2 - \sigma_N) \left((s \sin(\theta))^2 + \frac{1}{2} \right) + \frac{\sigma_N}{2} \sqrt{\frac{\pi T_w}{T_\infty}} s \sin(\theta) \right\} (1 + erf(s \sin(\theta))) \Bigg] \end{aligned} \tag{9.9}$$

$$C_\tau = -\frac{\sigma_N \cos(\theta)}{s\sqrt{\pi}} \left[e^{-(s \sin(\theta))^2} + \sqrt{\pi} s \sin(\theta) (1 + erf(s \sin(\theta))) \right] \tag{9.10}$$

where C_p represents the pressure coefficient, C_τ represents the shear coefficient, σ_N represents the normal momentum accommodation coefficient, σ_T represents the tangential momentum accommodation coefficient, T_w represents the surface or body wall temperature, T_∞ represents the free stream temperature, V_∞ represents the speed of the object or free stream velocity, $erf()$ represents the error function and s represents the speed ratio, which reads:

$$s = \frac{V_\infty}{\sqrt{2R_U T_\infty}} \tag{9.11}$$

where R_U represents the universal gas constant. The above equations can be found in the work of Schaaf and Chambre (164).

9.1.2.3 Transition flow regime

Aerodynamic computations in the transition regime are performed using the sigmoid bridging functions, which are introduced in the work of Mehta et al. (161). The developed function uses the sigmoid as the basis function and to obtain optimized accuracy, two sigmoid functions are used. The sigmoid function reads,

$$\begin{aligned} C_{X_{trans}} &= C_{X_c} + \\ &+ (C_{X_{fm}} - C_{X_c})[a_{si} sig_{10}(b_{si} log_{10}(Kn) + c_{si}) + d_{si} sig_{10}(e_{si} log_{10}(Kn) + f_{si}) + g_{si}] \end{aligned} \tag{9.12}$$

where a_{si} - g_{si} represents the fitting constants and $sig_{10}(x)$ represents the sigmoid function, which reads

$$sig_{10}(x) = \frac{1}{1 + 10^{(-/+)x}} \tag{9.13}$$

where the sign of the exponent depends on the trend of the coefficient across the transition region.

9.2 Uncertain parameters of the re-entry case

For the debris re-entry case, a 16-D case is considered. This represents a high dimensional problem, which requires a large number of simulations - expensive function calls. In this work, we focus on the longitudinal and lateral distributions of the impact location, which are computed using the following relation:

$$y_d = F(\mathbf{X})Re \tag{9.14}$$

where y_d represents two separate distributions in the longitudinal and lateral directions and Re represents the radius of the Earth. These distributions correspond to two separate outputs of $F(\mathbf{X})$, which are the longitude and latitude angle difference between the entry point and the impact location. One should keep in mind that this problem represents two separate uncertainty propagation problems.

Three cases are investigated. All cases consider the same uncertain inputs and differ only in the flight path angle distribution. The first case represents the controlled normal re-entry, where the flight path angle range from 0 to -2.5 degrees. This case represents common re-entry problem and results are applicable to re-entry of rocket parts or satellites. The second and third case represents the re-entry, where the first case considers the shallow angle (0 - -0.1 degrees) and the second case considers the steep angle (-4.5 - -5.5 degrees).

The first 10 random variables represent the atmospheric conditions and are considered to be uncertain. However, the code for debris propagation requires a continuous function for the atmospheric properties and to overcome this problem, the interpolation routines were developed. Distribution data for the atmospheric properties were taken from the US standard atmosphere (165). The mean value for the altitude 120 km is read off the US standard atmosphere and the standard deviation was assumed to be the same as for the altitude 90 km. The example of the interpolation is showed in Fig. 9.1. Each function call uses one sample. The atmosphere is assumed to be composed of N2 and O2 and the mole fraction of N2 is considered to be uncertain. The heat capacity (C_p) of the free-stream air that has an effect on the computation of drag coefficient through the pressure behind the shock is also assumed to be uncertain. The input distributions are summarized in Tab. 9.1.

9.3 Results and discussion

The uncertainty propagation is done on the debris re-entry. Three cases are considered, where the first represents the normally controlled re-entry and the other two cases represent the uncontrolled re-entry with a shallow angle and controlled re-entry with a steep angle, respectively. First, let us discuss the controlled re-entry.

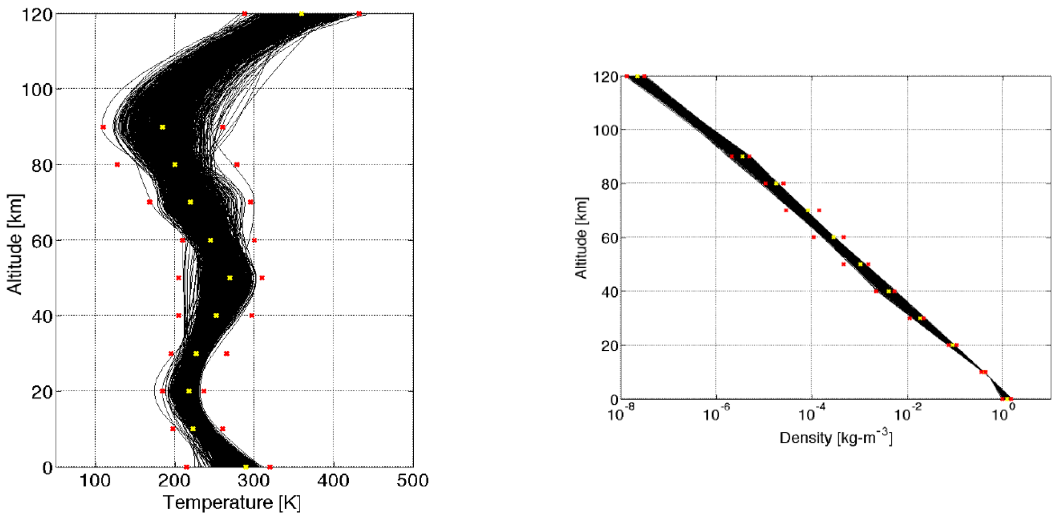


Figure 9.1: 1000 samples from the interpolation routine of:
Left: Temperature Right: Density

9.3.1 Controlled normal re-entry

The results of MC simulations are summarized in Tab. 9.2 and results obtained by the adaptive UQ-HDMR method are summarized in Tab. 9.3. Sensitivities of all the selected increment functions are given in Tab. 9.4 for the lateral distribution and in Tab. 9.5 for the longitudinal distribution.

The flight path angle (x_{13}) has the strongest effect on the longitudinal impact location and it is responsible for more than 75% of the total variation in the final distribution and offset from the central sample is more than 130 km. The direction angle (x_{16}) has the strongest effect on the lateral distribution with all other independent variables providing a negligible contribution. In the longitudinal direction, an important contribution is given by the increment function $dF_{13.14}$, which represents the interaction of the flight path angle and the re-entry speed. In the lateral direction, important contributions come from the interaction effects of the direction angle, the re-entry speed, and the flight path angle. These interaction effects are responsible for the tails of the final distribution, if one considers only the direction angle, the final distribution would collapse to the uniform distribution.

In Fig. 9.2 is shown the final histogram for the longitudinal distribution, which is obtained with our method and also, with MC simulation. The lateral distribution is showed in Fig. 9.3, where histograms obtained by our method and MC

ID	Random Var.	Description	Distrib. type	Mean	Standard dev.
x_1	T_0	Temperature at 0 km $[K]$	Gumbell	280	16.667
x_2	T_{20}	Temperature at 20 km $[K]$	Gumbell	218	7.333
x_3	T_{50}	Temperature at 50 km $[K]$	Landau	252	16.667
x_4	T_{70}	Temperature at 70 km $[K]$	Landau	187	24.0
x_5	T_{90}	Temperature at 90 km $[K]$	Normal	185	25.0
x_6	T_{120}	Temperature at 120 km $[K]$	Normal	360	24.0
x_7	ρ_0	Density at 0 km $[kg/m^3]$	Normal	1.225	8.167e-2
x_8	ρ_{40}	Density at 40 km $[kg/m^3]$	Normal	4.0e-3	5.330e-4
x_9	ρ_{90}	Density at 90 km $[kg/m^3]$	Normal	3.416e-6	5.693e-7
x_{10}	ρ_{120}	Density at 120 km $[kg/m^3]$	Normal	2.222e-8	3.703e-9
				Min	Max
x_{11}	X_{N2}	Percentage of $N2$ [%]	Uniform	0.784	0.816
x_{12}	C_p	Heat Capacity $[J/K]$	Uniform	1304.35	1441.65
x_{13}	γ	Flight path angle $[deg]$ - Shallow	Uniform	0	-0.1
	γ	Flight path angle $[deg]$ - Normal	Uniform	0	-2.5
	γ	Flight path angle $[deg]$ - Steep	Uniform	-4.5	-5.5
x_{14}	V_∞	Re-entry speed $[m/s]$	Uniform	7410	7790
x_{15}	m	Mass of debris $[kg]$	Uniform	243.75	256.25
x_{16}	χ	Direction Angle $[deg]$	Uniform	87.5	92.5

Table 9.1: Input distributions for the debris re-entry

simulation are displayed. One important aspect should be noted. Very high relative error for the final mean for the lateral case is caused by the ill-conditioned relative error. In other words, one can see from Fig. 9.3 that the final distribution is symmetrical around 0 and therefore, improving absolute accuracy will lead to increase of the relative error, i.e. $0/0 \to \infty$. The error between our method and MC simulation is around $7m$ with a spread over $600km$ and therefore, one can consider an obtained result as an accurate one. From all given figures, one can understand that our method provided an accurate result with negligible errors and therefore, one can state that the proposed method works well.

The partial histograms are showed in Fig. 9.4 for the longitudinal distribution and in Fig. 9.5 for the lateral distribution. For the longitudinal distribution, the partial histogram for the flight path angle (x_{13}) suggests that the underlying function has a parabolic shape (i.e. x^2). This suggests that the same error in measuring higher velocities will lead to a larger uncertainty. The same conclusion can be applied to the re-entry speed (x_{14}). Also, the size of the contribution suggests that these two variables are responsible for the overall shape of the final distribution. The final distribution has a sharp peak that smoothly transitions into a long tail. The peak and the long tail is a result of the second order increment functions, where the tail is caused by a steep ascent in the underlying function in one of the corners of the given stochastic domain, i.e. to avoid such a strong tail, the input distributions need to be shortened from one side. The smooth decrease on the left side of the interaction increment function is responsible for the smooth decrease on the left side in the final distribution.

For the lateral case, the main shape of the final distribution is from the direction angle (x_{16}). The distribution exhibits a smooth Gaussian-like transition from the tails to the peak where a plateau feature is quite clearly visible. The smooth transition is contributed to the interaction effects of the increment functions $dF_{13.16}$ and $dF_{14.16}$, whereas the plateau is derived from the uniform distribution of the increment function dF_{16}. The very long tails of the final distribution are obtained with the increment function $dF_{13.14.16}$, i.e. the third order interaction of the flight path angle, the direction angle, and the re-entry speed.

Function calls	Mean longitudinal	Standard dev. longitudinal
100000	2.328e+03	7.733e+02
	Mean lateral	Standard dev. lateral
100000	5.085e-02	6.418e+01

Table 9.2: MC simulation for normal re-entry problem

Case	Desired accuracy	Function calls	Mean	Standard Deviation	Relative error of Mean	Relative error of S. D.
Longitudinal	0.01	91	2.330e+03	7.785e+02	8.591e-04	6.724e-03
Lateral	0.01	124	5.664e-02	6.495e+01	1.139e-01	1.200e-02

Table 9.3: Results of the high dimensional adaptive UQ-HDMR approach for debris re-entry: controlled-normal

Increment function	Partial Mean	Partial Variance	Mean Sensitivity	Variance Sensitivity
1	1.1036e-02	2.1632e-03	5.4489e-08	4.5898e-15
2	-6.7442e-05	1.3578e+00	3.3299e-10	2.8809e-12
3	-5.6970e-02	3.6448e+02	2.8128e-07	7.7333e-10
4	-2.6796e-01	3.1826e+01	1.3230e-06	6.7528e-11
5	-7.1419e-02	3.7174e+00	3.5262e-07	7.8874e-12
6	1.8877e-03	5.2654e-01	9.3202e-09	1.1172e-12
7	1.0087e-05	6.0361e-08	4.9802e-11	1.2807e-19
8	9.9100e+02	1.4726e+08	4.8929e-03	3.1245e-04
9	1.0291e+03	1.7914e+08	5.0812e-03	3.8009e-04
10	5.0657e+01	1.2840e+06	2.5011e-04	2.7243e-06
11	-1.2810e-03	7.6214e-01	6.3248e-09	1.6171e-12
12	-4.2154e+00	4.8746e+04	2.0813e-05	1.0343e-07
13	1.3197e+05	3.6725e+11	**6.5161e-01**	**7.7922e-01**
14	2.8101e+04	3.8826e+10	**1.3874e-01**	**8.2380e-02**
15	-2.4426e+01	8.2723e+06	1.2060e-04	1.7552e-05
16	-6.7447e+02	3.6563e+05	3.3301e-03	7.7578e-07
13.14	3.9540e+04	6.4843e+10	**1.9523e-01**	**1.3758e-01**
9.14	1.2458e+02	3.2518e+07	6.1510e-04	6.8995e-05
9.13	2.1854e+01	1.8683e+07	1.0790e-04	3.9641e-05

Table 9.4: Results of the partial increment functions for the longitudinal distribution considering the controlled re-entry

9.3.2 Uncontrolled 'Shallow' re-entry

The results of MC simulations for the shallow re-entry case are summarized in Tab. 9.6 and results obtained with the adaptive UQ-HDMR method for the shallow re-entry case are summarized in Tab. 9.7. Sensitivities of all the selected increment functions are given in Tab. 9.8 for the longitudinal distribution and in Tab. 9.9 for the lateral distribution. In Fig. 9.6 is showing the final histogram for the longitudinal distribution, which is obtained with our method and with MC simulation. Comparison of both methods is also provided in given figure.

The lateral distribution is showed in Fig. 9.7, where histograms obtained with our method and MC simulation are displayed. In both cases, the distribution looks almost identical. However, there is a small peak at the end of the longitudinal

Increment function	Partial Mean	Partial Variance	Mean Sensitivity	Variance Sensitivity
1	6.7575e-19	8.1108e-36	9.2454e-21	2.3574e-45
2	-4.1296e-21	5.0909e-33	5.6500e-23	1.4797e-42
3	-3.4884e-18	1.3666e-30	4.7727e-20	3.9720e-40
4	-1.6408e-17	1.1933e-31	2.2449e-19	3.4683e-41
5	-4.3731e-18	1.3938e-32	5.9832e-20	4.0511e-42
6	1.1559e-19	1.9742e-33	1.5814e-21	5.7381e-43
7	6.1764e-22	2.2632e-40	8.4503e-24	6.5780e-50
8	6.0681e-14	5.5214e-25	8.3021e-16	1.6048e-34
9	6.3016e-14	6.7166e-25	8.6215e-16	1.9522e-34
10	3.1019e-15	4.8142e-27	4.2438e-17	1.3993e-36
11	-7.8439e-20	2.8576e-33	1.0732e-21	8.3056e-43
12	-2.5812e-16	1.8277e-28	3.5315e-18	5.3122e-38
13	8.2633e-12	1.3910e-21	1.1306e-13	4.0431e-31
14	1.7207e-12	1.4558e-22	2.3542e-14	4.2312e-32
15	-1.4957e-15	3.1016e-26	2.0463e-17	9.0149e-36
16	-6.0825e+00	2.9858e+09	**8.3218e-02**	**8.6782e-01**
14.16	1.0191e+01	2.8120e+07	**1.3943e-01**	**8.1732e-03**
13.16	4.6801e+01	3.1536e+08	**6.4031e-01**	**9.1659e-02**
9.16	-1.2461e+00	1.4266e+05	1.7049e-02	4.1465e-05
8.16	4.0492e-01	1.0467e+05	5.5400e-03	3.0422e-05
13.14	2.2512e-12	3.7074e-22	3.0800e-14	1.0776e-31
13.14.16	8.3658e+00	1.1104e+08	**1.1446e-01**	**3.2275e-02**

Table 9.5: Results of the partial increment functions for the lateral distribution considering the controlled re-entry

distribution (Fig. 9.6), which is not caught by our method. We run deeper analysis to understand this peak and it was found that the peak is a result of small oscillation in the MC distribution that is not captured with the interpolation technique for the increment function dF_{14} (see Fig. 9.8). Unfortunately, such problems (capturing small deviations from the interpolation function) cannot be solved in general and are inherent to numerical integration and interpolation methods and form as such a limitation of the HDMR method.

The same problem as in the previous case represents the high relative error for the final mean of the lateral case. Again, the final distribution is symmetrical around

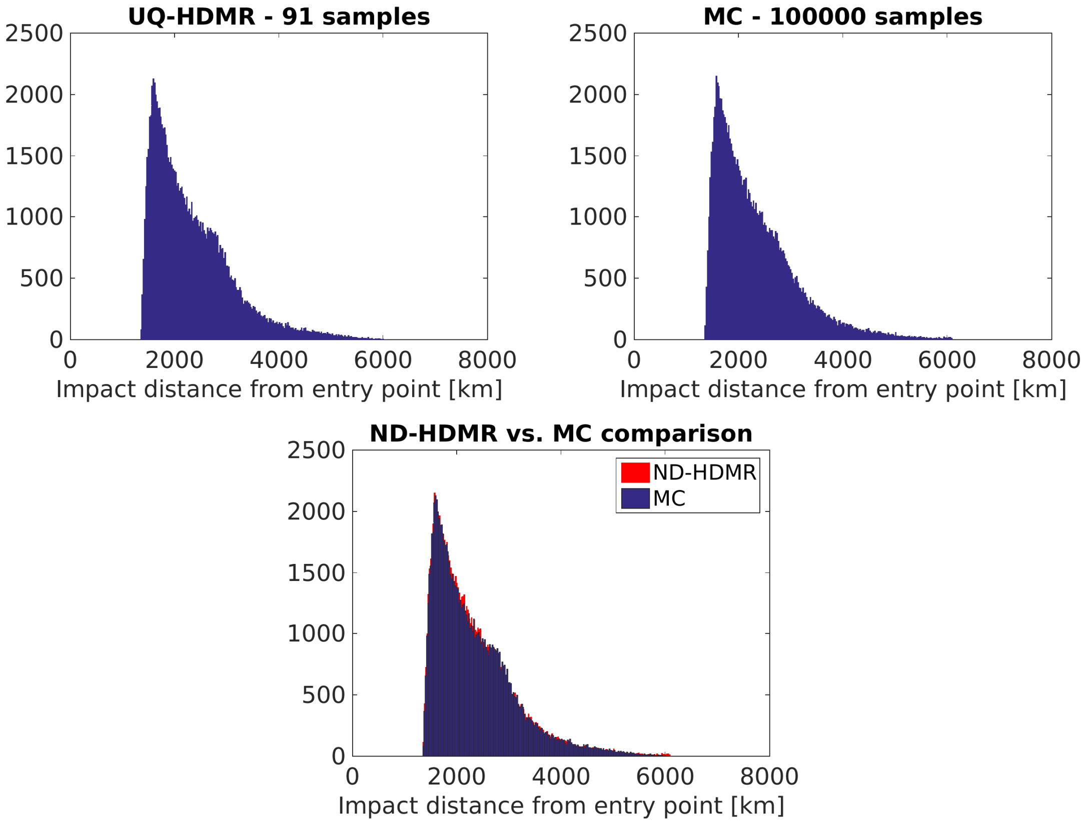


Figure 9.2: Final histograms for the longitudinal distribution of the controlled normal case

0 and therefore, a more accurate result will lead to a higher relative error, i.e. $0/0 \rightarrow \infty$. Nevertheless, the difference between MC and our method is around 20 meters, where the spreadiness is around 600 kilometres. Therefore, one can conclude that our method provides an accurate result for the lateral distribution as can be seen in Fig. 9.7.

The partial histograms are shown in Fig. 9.9 for the longitudinal distribution and in Fig. 9.10 for the lateral distribution. For the longitudinal case, the important variable is the re-entry speed (x_{14}), which also gives the overall shape of the output distribution. The shape of the partial histogram suggests that the increment function has a parabolic shape (x^2) and therefore, higher velocities will lead to a higher uncertainty. Next variable influencing the output is the flight path angle (x_{13}), where the partial histogram suggests the linear shape of the increment function. The important interaction effect is the combination of the re-entry speed and the flight path angle. This interaction effect is responsible for the smooth transition on the left and right side of the output distribution.

For the lateral distribution, the most influential variable is the direction angle (x_{16}), which is responsible for the plateau in the final distribution. The smooth transition is again given by the interaction effects, namely the increment function $dF_{14.16}$, $dF_{13.16}$, $dF_{8.16}$ and $dF_{9.16}$, however, this time the tails are no so strong as in the normal re-entry case and therefore, higher order interactions are not included.

Function calls	Mean Longitudinal	Standard dev. Longitudinal
100000	3.807e+03	9.197e+02
	Mean Lateral	Standard dev. Lateral
100000	4.146e-02	1.066e+01

Table 9.6: MC simulation for shallow re-entry problem

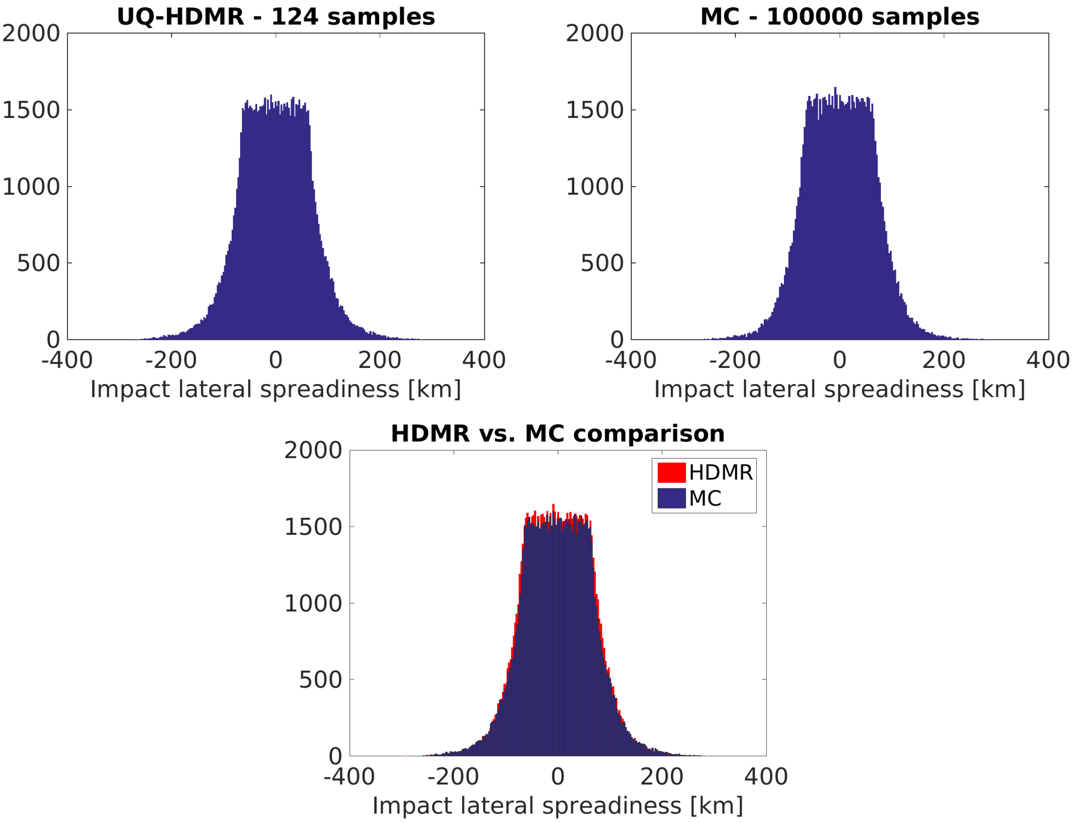
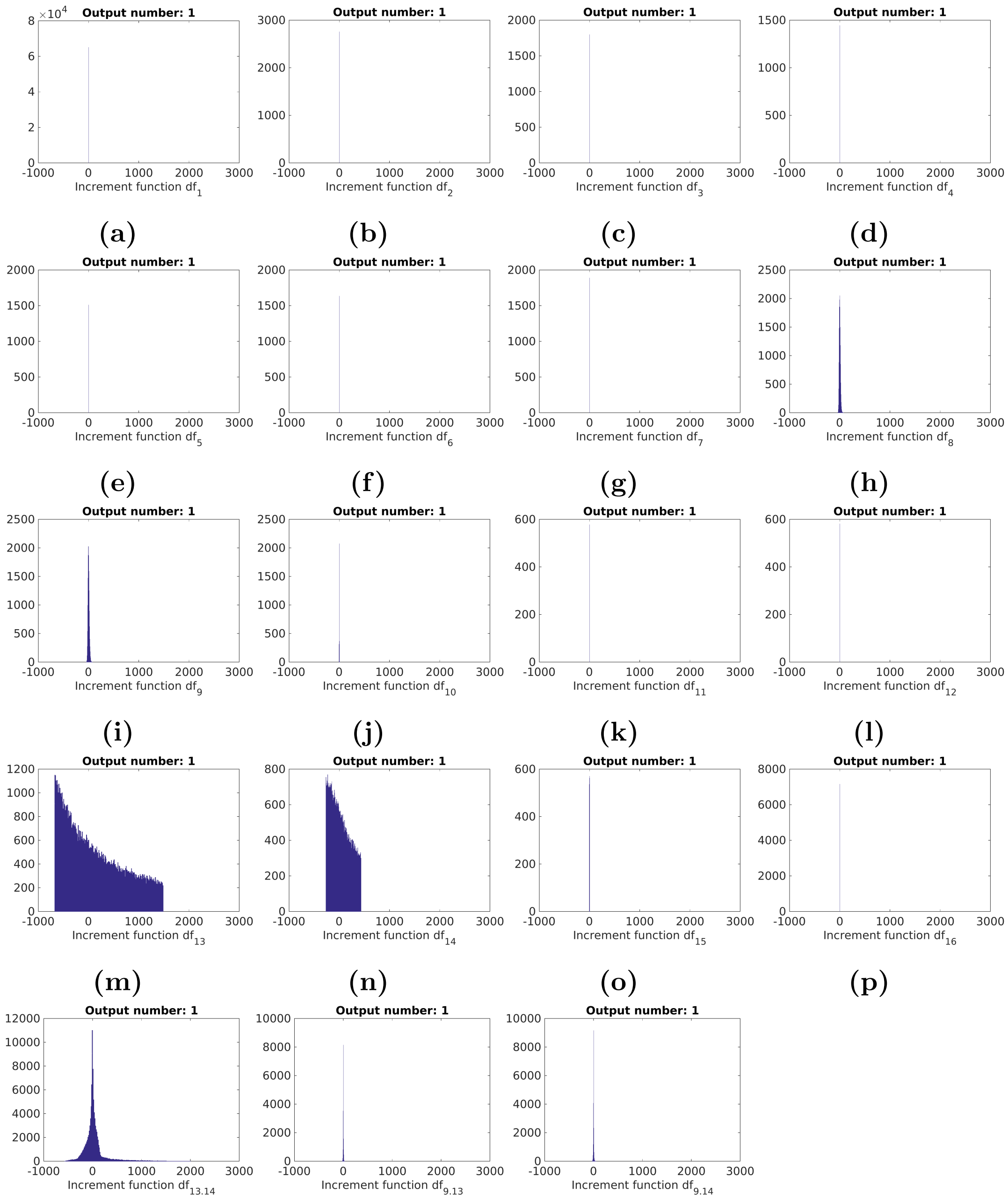


Figure 9.3: Final histograms for the lateral distribution of the controlled normal case



(a) **(b)** **(c)** **(d)**

(e) **(f)** **(g)** **(h)**

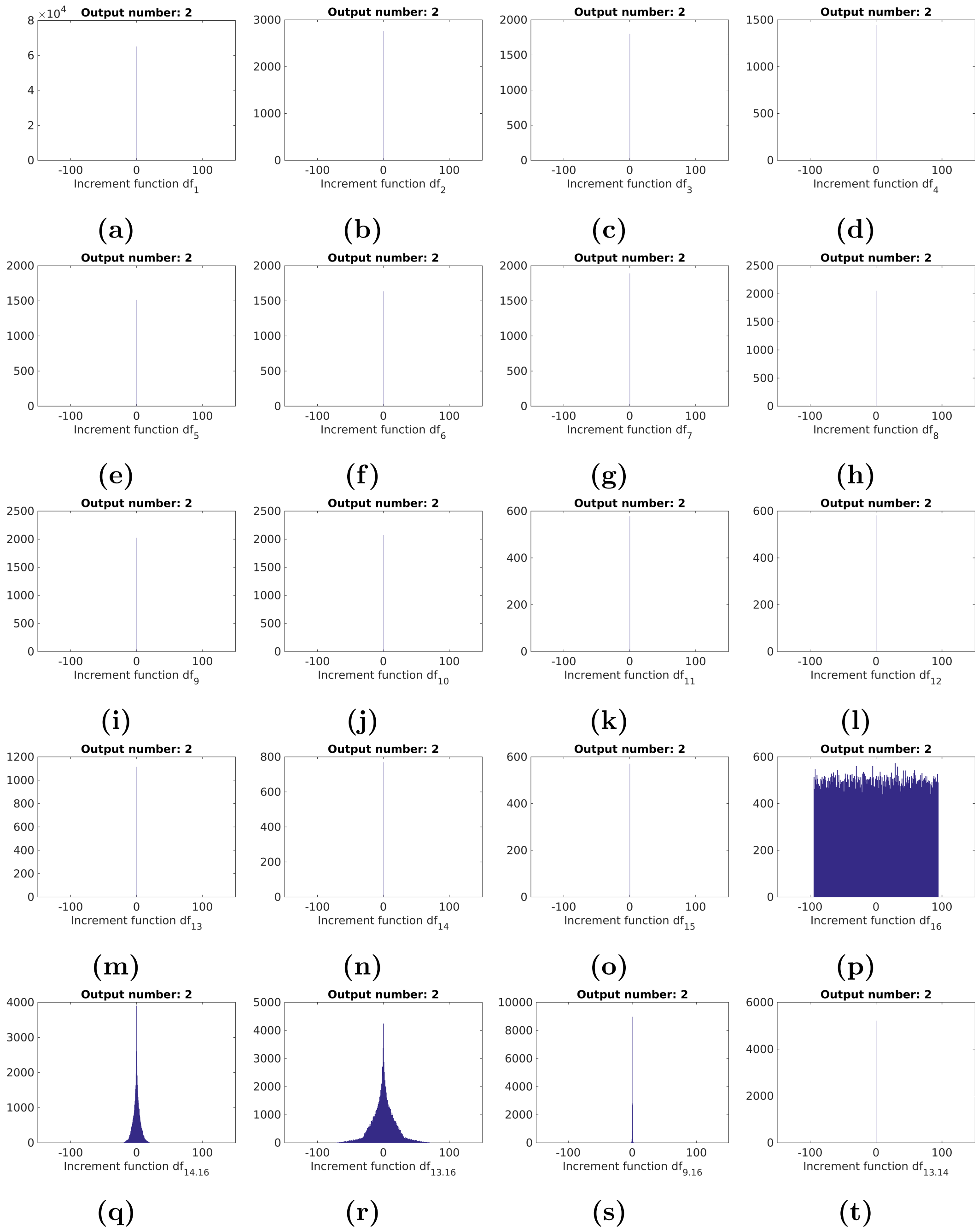
(i) **(j)** **(k)** **(l)**

(m) **(n)** **(o)** **(p)**

(q) **(r)** **(s)**

Figure 9.4: Partial histograms for the increment functions of the controlled reentry problem for the longitudinal impact distance

Note: Output number: 1 - Longitudinal impact distance
Output number: 2 - Lateral spreadiness



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

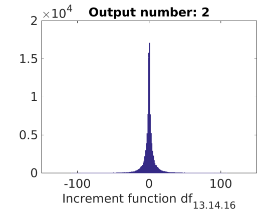
(m) (n) (o) (p)

(q) (r) (s) (t)

Figure 9.5: Partial histograms for the increment functions of the controlled re-entry problem for the lateral spreadiness

Note: Output number: 1 - Longitudinal impact distance

Output number: 2 - Lateral spreadiness



(u)

Figure 9.5: Partial histograms for the increment functions of the controlled re-entry problem for the lateral spreadiness

Note: Output number: 1 - Longitudinal impact distance
Output number: 2 - Lateral spreadiness

Case	Desired accuracy	Function calls	Mean	Standard Deviation	Relative error of Mean	Relative error of S. D.
Longitudinal	0.01	80	3.798e+03	7.785e+02	2.354e-03	2.665e-02
Lateral	0.01	108	5.783e-02	1.084e+01	3.948e-01	1.713e-02

Table 9.7: Results of the high dimensional adaptive UQ-HDMR approach for debris re-entry: shallow angle

Increment function	Partial Mean	Partial Variance	Mean Sensitivity	Variance Sensitivity
1	1.2867e-02	2.9407e-03	4.7516e-08	3.6354e-15
2	-9.6873e-05	1.8752e+00	3.5774e-10	2.3182e-12
3	-5.7214e-02	4.1188e+02	2.1128e-07	5.0918e-10
4	-3.3922e-01	5.1683e+01	1.2527e-06	6.3891e-11
5	-1.3942e-01	3.0501e+01	5.1487e-07	3.7706e-11
6	1.0830e-02	1.6553e+01	3.9995e-08	2.0463e-11
7	1.0319e-05	6.2593e-08	3.8105e-11	7.7380e-20
8	1.1156e+03	1.6708e+08	4.1199e-03	2.0655e-04
9	1.6633e+03	4.0741e+08	6.1424e-03	5.0365e-04
10	2.4320e+02	2.1903e+07	8.9810e-04	2.7077e-05
11	-3.4832e-03	5.6268e+00	1.2863e-08	6.9560e-12
12	-7.5151e+00	1.7393e+05	2.7752e-05	2.1502e-07
13	7.1023e+02	2.3184e+09	**2.6228e-03**	**2.8661e-03**
14	2.6347e+05	8.0587e+11	**9.7294e-01**	**9.9623e-01**
15	-3.5105e+01	1.5726e+07	1.2964e-04	1.9441e-05
16	-1.1176e+03	1.0039e+06	4.1271e-03	1.2410e-06
13.14	-1.9597e+03	8.8256e+07	7.2368e-03	1.0910e-04
10.14	-7.3394e+01	3.2508e+06	2.7103e-04	4.0187e-06
9.14	-4.0132e+02	2.4705e+07	1.4820e-03	3.0540e-05

Table 9.8: Results of the partial increment functions for the longitudinal distribution considering the shallow re-entry case

Increment function	Partial Mean	Partial Variance	Mean Sensitivity	Variance Sensitivity
1	7.8789e-19	1.1026e-35	9.0871e-21	1.1182e-45
2	-5.9317e-21	7.0309e-33	6.8413e-23	7.1303e-43
3	-3.5034e-18	1.5443e-30	4.0406e-20	1.5661e-40
4	-2.0771e-17	1.9378e-31	2.3957e-19	1.9652e-41
5	-8.5372e-18	1.1436e-31	9.8464e-20	1.1598e-41
6	6.6316e-19	6.2062e-32	7.6486e-21	6.2939e-42
7	6.3184e-22	2.3469e-40	7.2873e-24	2.3801e-50
8	6.8313e-14	6.2644e-25	7.8788e-16	6.3530e-35
9	1.0185e-13	1.5275e-24	1.1747e-15	1.5491e-34
10	1.4892e-14	8.2124e-26	1.7175e-16	8.3284e-36
11	-2.1328e-19	2.1097e-32	2.4599e-21	2.1395e-42
12	-4.6017e-16	6.5214e-28	5.3074e-18	6.6136e-38
13	4.3489e-14	8.6926e-24	5.0158e-16	8.8155e-34
14	1.7573e-11	3.1813e-21	2.0268e-13	3.2262e-31
15	-2.1496e-15	5.8963e-26	2.4792e-17	5.9797e-36
16	-1.0407e+01	8.7407e+09	**1.2003e-01**	**8.8643e-01**
14.16	7.1791e+01	1.1174e+09	**8.2800e-01**	**1.1332e-01**
13.16	-1.9397e+00	1.9680e+06	2.2372e-02	1.9959e-04
9.16	-2.0868e+00	3.9527e+05	2.4068e-02	4.0085e-05
8.16	4.7933e-01	1.4266e+05	5.5283e-03	1.4468e-05

Table 9.9: Results of the partial increment functions for the lateral distribution considering the shallow re-entry case

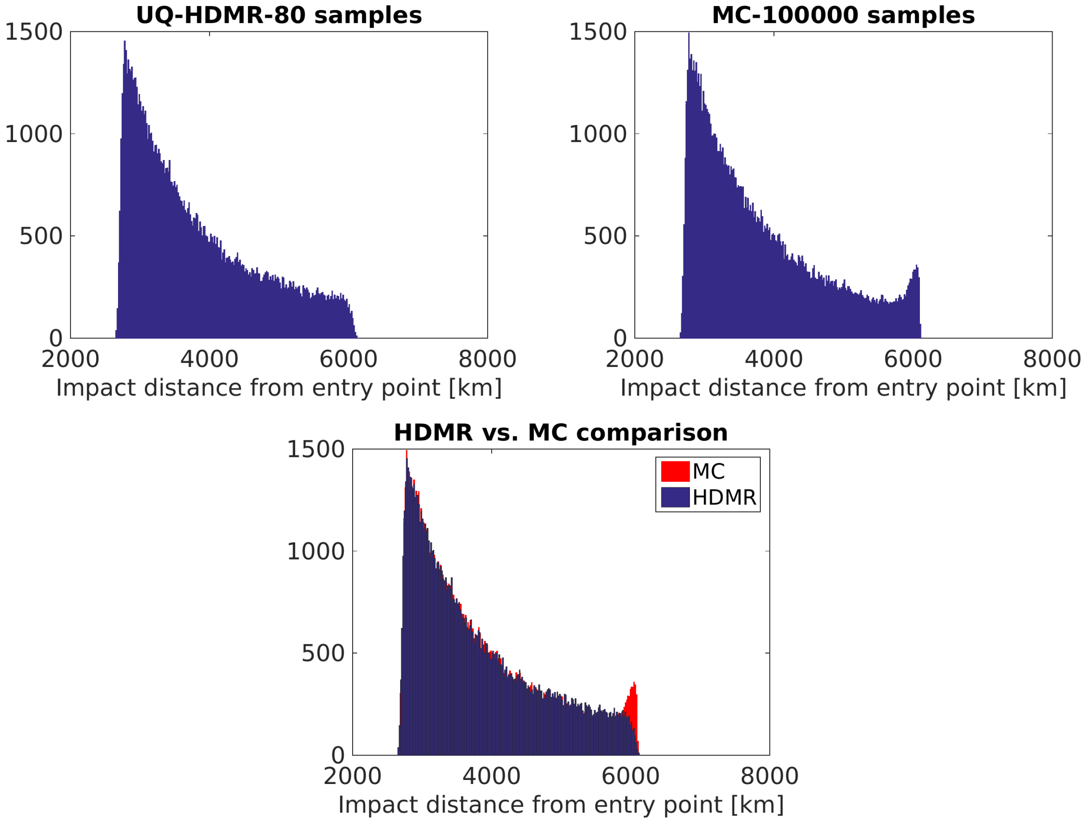


Figure 9.6: Final histograms for the longitudinal distribution of the shallow re-entry case

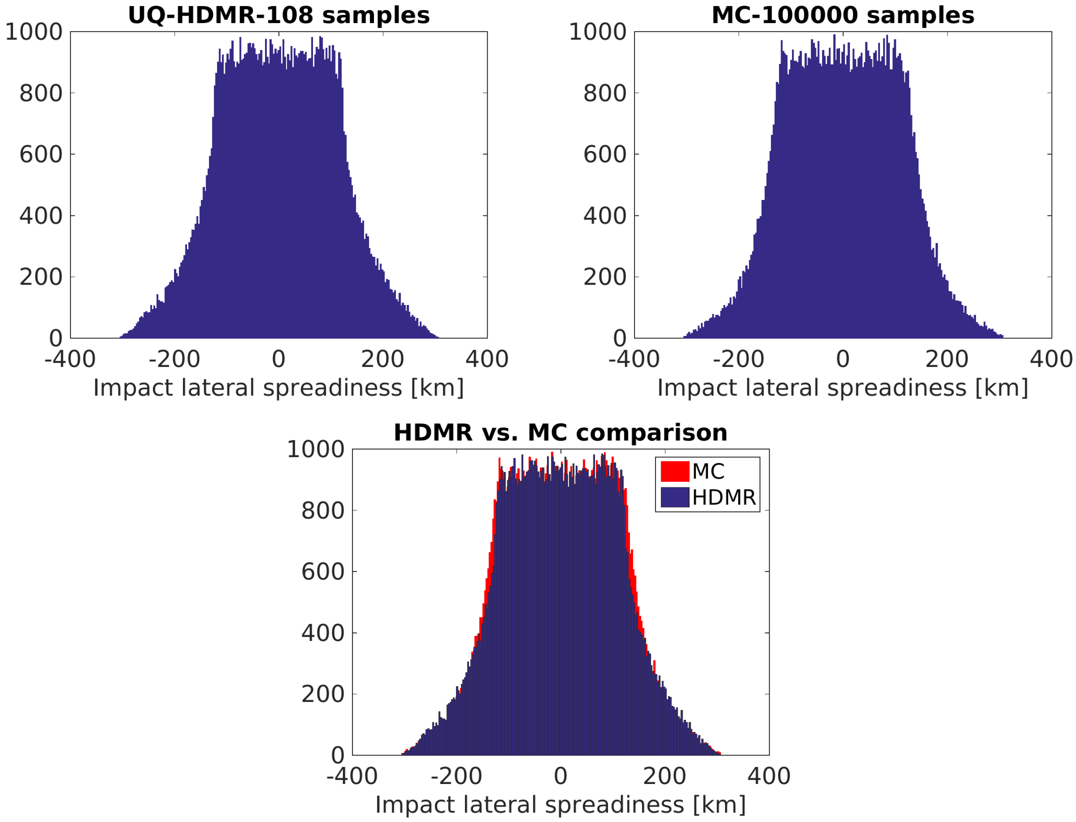


Figure 9.7: Final histograms for the lateral distribution of the shallow re-entry case

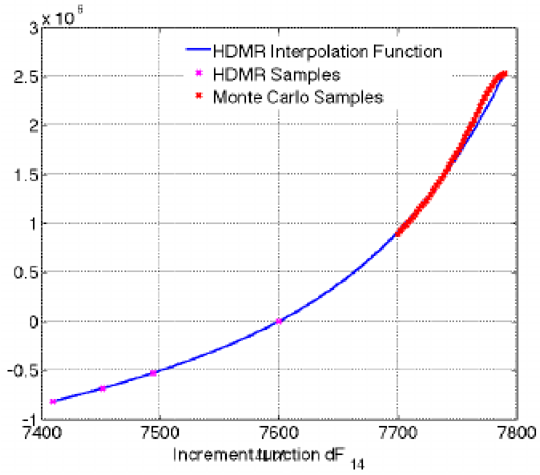
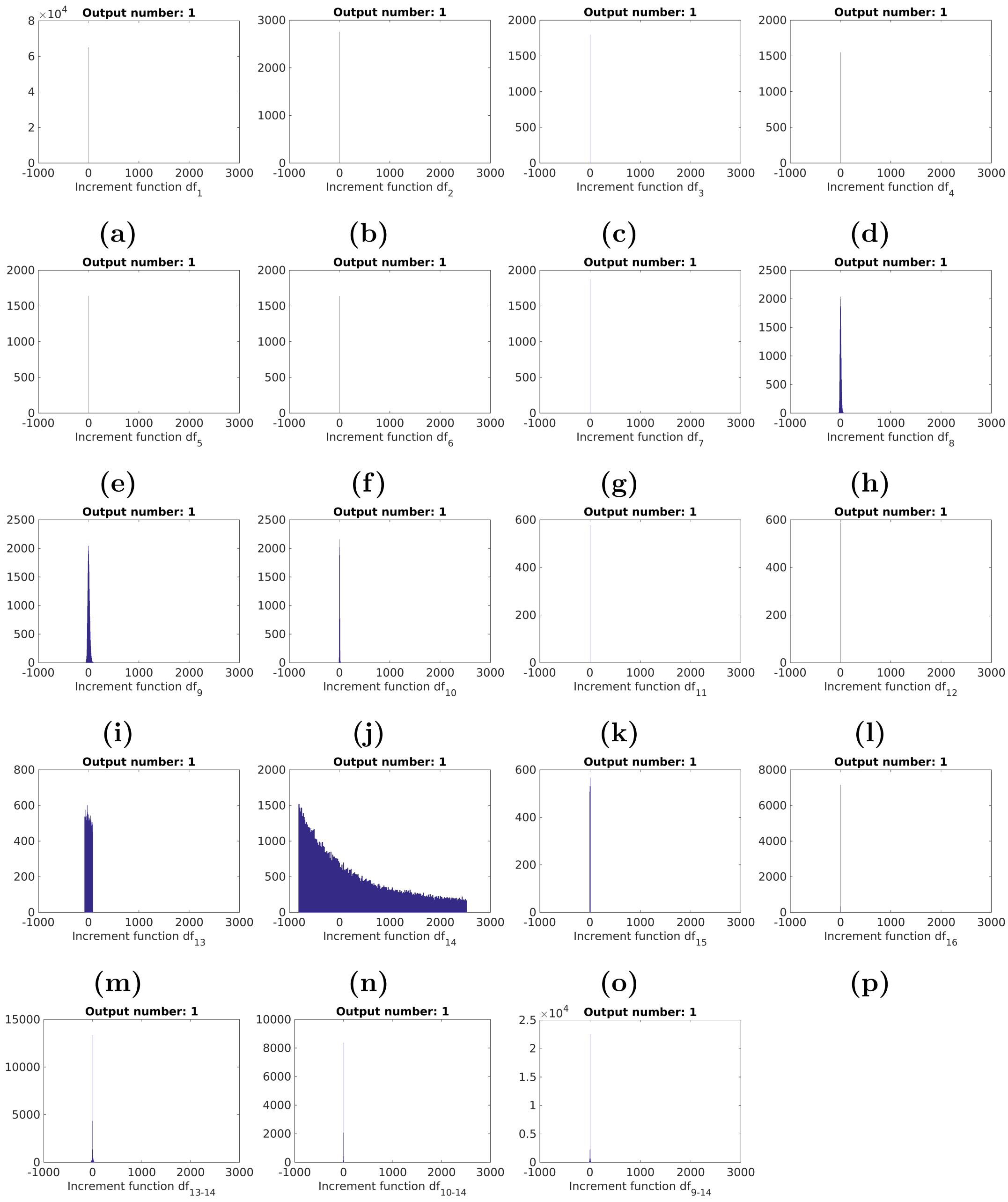


Figure 9.8: The cause of the peak at the tail of the longitudinal HDMR PDF

Note: The graph represents the increment to the longitudinal impact distribution (Y abscissa) on the change of speed during the re-entry process (X abscissa). For the re-entry speed between 7700 and 7800, the interpolation fails to accurately describe the stochastic domain as can be seen on the difference between the MC simulation and the interpolation model. Hence, the small peak on the right side of the longitudinal distribution (Fig. 9.6).



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

(q) (r) (s)

Figure 9.9: Partial histograms for the increment functions of the shallow re-entry problem for the longitudinal impact distance

Note: Output number: 1 - Longitudinal impact distance

Output number: 2 - Lateral spreadiness

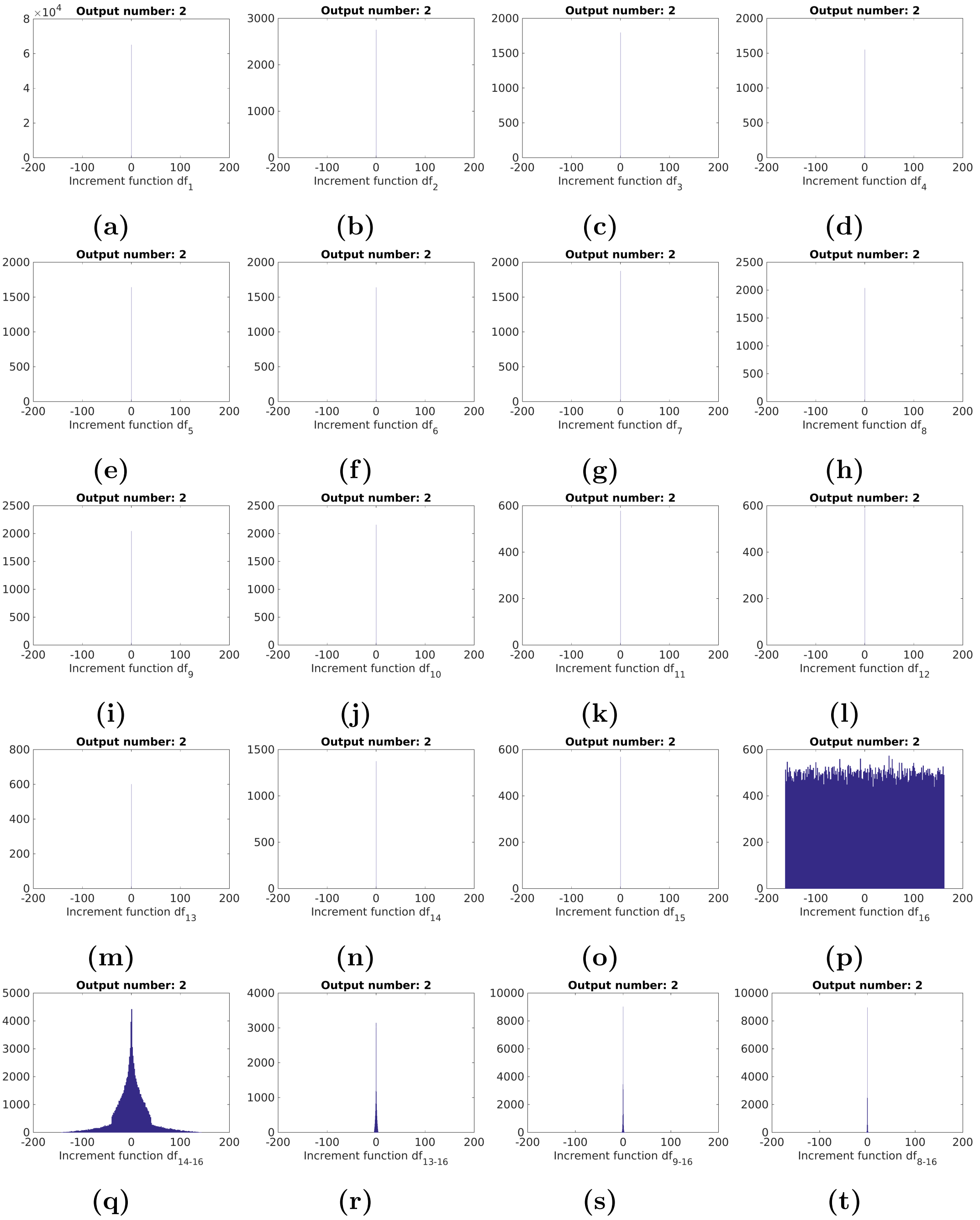


Figure 9.10: Partial histograms for the increment functions of the shallow re-entry case for the lateral spreadiness

Note: Output number: 1 - Longitudinal impact distance

Output number: 2 - Lateral spreadiness

9.3.3 Controlled 'Steep' re-entry

The results of MC simulations are summarized in Tab. 9.10 and results obtained with the adaptive UQ-HDMR method are summarized in Tab. 9.11. Sensitivities of all the selected increment functions are given in Tab. 9.12 for the longitudinal distribution and in Tab. 9.13 for the lateral distribution. In Fig. 9.11 is showing the final histogram for the longitudinal distribution, which is obtained by our method and with MC simulation. The lateral distribution is showed in Fig. 9.12, where histogram obtained with our method and MC simulation are displayed. In both cases, the distribution looks almost identical and both residuals are under prescribed tolerance.

Several interesting observations can be made on comparing the longitudinal and lateral distributions for the different cases (un-controlled shallow and controlled normal and steep cases):

1. The longitudinal distribution seems to approach a normal distribution with steeper flight path angle, which is expected because as the re-entry occurs at steeper angles, the amount of time the object spends traversing the atmosphere is drastically reduced and so is the chance for the other uncertainties (especially atmospheric) to have an effect.
2. The longitudinal impact distribution for the shallow re-entry case is not as wide as that for the normal case because of a smaller input distribution of the flight path angle. Both distributions have the same larger boundary because the input flight path angle distributions for both cases have a common boundary at zero degrees that corresponds to the largest impact distance from the entry point. The longitudinal distribution for the controlled steep case has a very small spread with large impact probabilities close to the mean value of the distribution and has no overlap with the other two cases because the object falls well short of the impact locations for the other cases due to the high flight path angle values.
3. The lateral impact distributions for the shallow and normal cases have the same spread (boundary values) corresponding to the common boundary value of zero degrees in the input distributions for the flight path angle while the distribution is much narrower for the steep case. Increasing the flight path angle results in large impact probabilities close to the mean value of the distribution.

Function calls	Mean Longitudinal	Standard dev. Longitudinal
100000	8.900e+02	4.412e+01
	Mean Lateral	Standard dev. Lateral
100000	-2.940e-00	2.248e+01

Table 9.10: MC simulation for the controlled steep re-entry problem

Case	Desired accuracy	Function calls	Mean	Standard Deviation	Relative error of Mean	Relative error of S. D.
Longitudinal	0.01	84	8.902e+05	4.425e+04	2.573e-04	3.100e-03
Lateral	0.01	127	-2.964e-00	2.247e+04	8.200e-03	6.678e-04

Table 9.11: Results of the high dimensional adaptive UQ-HDMR approach for debris re-entry: controlled steep

The partial histograms are showed in Fig. 9.13 for the longitudinal distribution and in Fig. 9.14 for the lateral distribution. For the longitudinal distribution, the most influential variables are the flight path angle and the re-entry speed (measured with partial variance). From the partial histograms, one can understand that increment function for the re-entry speed (x_{14}) is linear and the increment function for the flight path angle (x_{13}) is slightly curved yet very close to linear. The interaction effects have negligible influence on the final distribution and only brings uncertainty to the final solution. For the lateral case, one can see that the influence of the direction angle is even higher, in this case, i.e. it is responsible for the plateau in the lateral distribution. Again, the interaction effects are responsible for the smooth transition of the final distribution, however, they do not have any special effect on the final distribution. Moreover, comparing the previous cases, one can see that the influence of interaction effects is diminishing.

9.4 Conclusion

In this chapter, the single fidelity approach is applied to the re-entry propagation problem. In all the considered cases, the developed method provided very good results for an excellent number of expensive function calls. Moreover, it provided deeper insight into debris re-entry and showed the importance of the probabilistic modelling.

Increment function	Partial Mean	Partial Variance	Mean Sensitivity	Variance Sensitivity
1	1.5969e-02	4.5295e-03	4.9937e-06	2.3103e-12
2	-1.6104e-04	3.1030e+00	5.0360e-08	1.5827e-09
3	-4.7919e-02	1.7323e+02	1.4984e-05	8.8354e-08
4	-2.2300e-01	5.0003e-01	6.9732e-05	2.5504e-10
5	-7.1047e-03	1.7789e-02	2.2217e-06	9.0732e-12
6	5.0908e-05	3.4684e-04	1.5919e-08	1.7690e-13
7	1.0234e-05	6.1582e-08	3.2002e-09	3.1409e-17
8	5.5167e+02	4.5043e+07	1.7251e-01	2.2974e-02
9	1.5987e+02	6.1107e+06	4.9992e-02	3.1167e-03
10	2.4520e+00	1.1313e+03	7.6675e-04	5.7702e-07
11	-6.3865e-05	1.8939e-03	1.9971e-08	9.6598e-13
12	-2.9540e+00	5.0781e+02	9.2373e-04	2.5901e-07
13	1.9207e+03	1.6530e+09	**6.0062e-01**	**8.4311e-01**
14	2.1576e+02	2.5203e+08	**6.7470e-02**	**1.2854e-01**
15	-9.2191e+00	1.2242e+06	2.8829e-03	6.2438e-04
16	-2.8065e+02	6.3305e+04	8.7759e-02	3.2288e-05
13.14	4.4114e+01	2.8550e+06	1.3795e-02	1.4562e-03
9.13	8.5510e+00	1.5342e+05	2.6739e-03	7.8250e-05
8.13	1.6323e+00	1.3301e+05	5.1043e-04	6.7841e-05

Table 9.12: Results of the partial increment functions for the longitudinal distribution considering the controlled steep re-entry case

The trajectory propagation is governed by a system of differential equations and the considered object is a simple sphere to simplify the propagation, where three aero-dynamical regimes are considered. The aero-dynamical regimes are the continuum flow, the free molecular flow, and the transition regime. This allows propagating the object from the high altitude to the ground. The observed criterion is the impact location in the longitudinal and lateral location.

In given test cases, we consider 16 uncertain parameters. First 10 uncertain parameters represent the atmospheric properties. We developed an interpolation routines[1] for these 10 parameters in order to simulate the uncertain atmosphere

[1] For each input parameter, a random sample is considered and the interpolation process is constructed between these samples. Lagrange interpolation is considered for the temperature and for the density. Also, isopycnic point for density is considered at the altitude of 10km.

Increment function	Partial Mean	Partial Variance	Mean Sensitivity	Variance Sensitivity
1	9.7783e-19	1.6983e-35	1.8278e-19	3.3790e-44
2	-9.8612e-21	1.1634e-32	1.8433e-21	2.3148e-41
3	-2.9342e-18	6.4950e-31	5.4848e-19	1.2923e-39
4	-1.3655e-17	1.8748e-33	2.5524e-18	3.7302e-42
5	-4.3504e-19	6.6699e-35	8.1321e-20	1.3271e-43
6	3.1172e-21	1.3004e-36	5.8269e-22	2.5874e-45
7	6.2665e-22	2.3089e-40	1.1714e-22	4.5940e-49
8	3.3780e-14	1.6888e-25	6.3144e-15	3.3602e-34
9	9.7891e-15	2.2911e-26	1.8298e-15	4.5585e-35
10	1.5014e-16	4.2418e-30	2.8065e-17	8.4396e-39
11	-3.9106e-21	7.1011e-36	7.3099e-22	1.4129e-44
12	-1.8088e-16	1.9040e-30	3.3812e-17	3.7883e-39
13	1.0435e-13	6.1329e-24	1.9506e-14	1.2202e-32
14	1.3212e-14	9.4494e-25	2.4696e-15	1.8801e-33
15	-5.6451e-16	4.5899e-27	1.0552e-16	9.1322e-36
16	-2.4924e+00	5.0134e+08	**4.6590e-01**	**9.9749e-01**
15.16	2.5140e-01	7.8919e+02	4.6993e-02	1.5702e-06
14.16	7.6188e-01	1.6212e+05	1.4242e-01	3.2257e-04
13.16	-1.4480e+00	1.0583e+06	2.7067e-01	2.1057e-03
9.16	-2.1655e-01	4.1401e+03	4.0479e-02	8.2374e-06
8.16	1.7945e-01	3.7133e+04	3.3545e-02	7.3881e-05

Table 9.13: Results of the partial increment functions for the lateral distribution considering the controlled steep re-entry case

along the whole propagation. Other uncertain variables are connected to the properties of the considered object.

All results obtained with our method show very good agreement with the MC simulation. However, the relative error in the mean value for some cases is very large. This is caused by badly defined relative error as the final distribution has the mean value around 0 and therefore, dividing by a number close to 0 will lead to an extreme value. Nevertheless, the final histogram shows an excellent similarity.

The partial histograms allowed deeper insight into the re-entry propagation problem. The interesting conclusion is about the tails of given distribution, which is

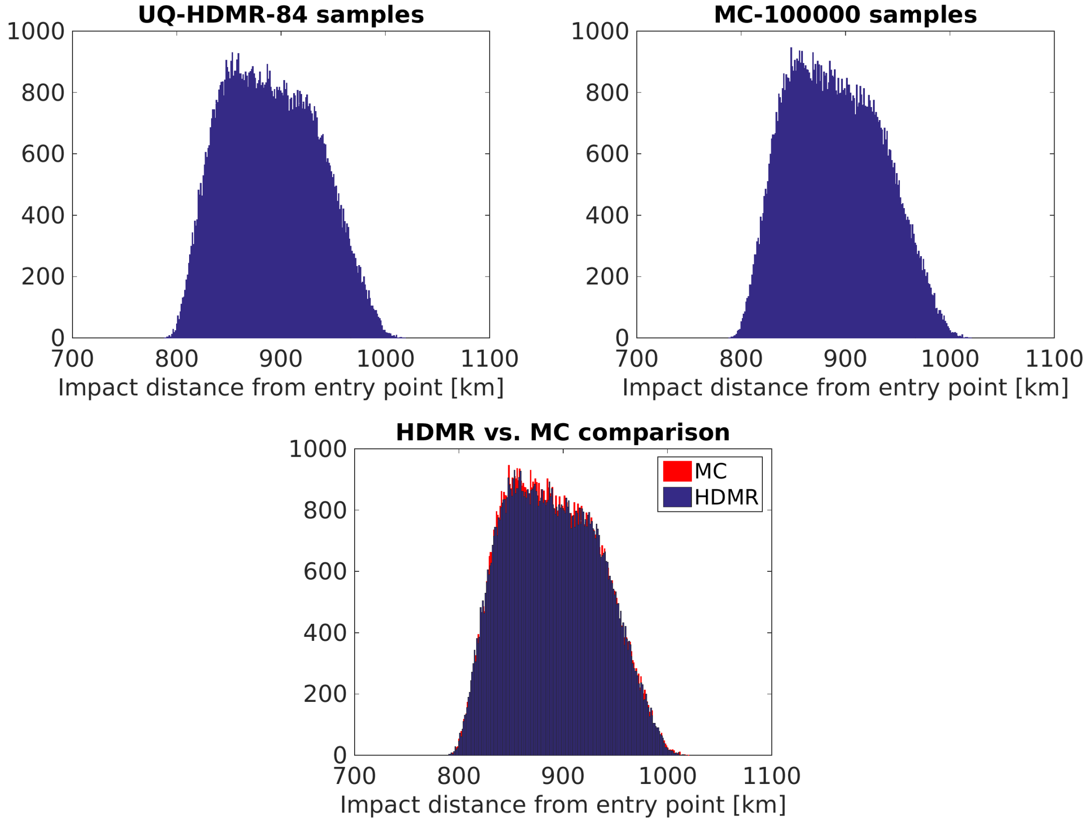


Figure 9.11: Final histograms for the longitudinal distribution of the controlled steep re-entry case

mainly caused by the interaction between flight path angle and the re-entry speed. This interaction is valid for both lateral and longitudinal distribution and such that decrease in this interaction lead to decrease in the spreadiness of the final result. Another interesting result represents the uncertainty at various flight path angles. If the flight path angle is large enough, the tail of the final distribution is starting to shrink. Therefore, increase accuracy in the flight path angle could improve the re-entry estimation in the future.

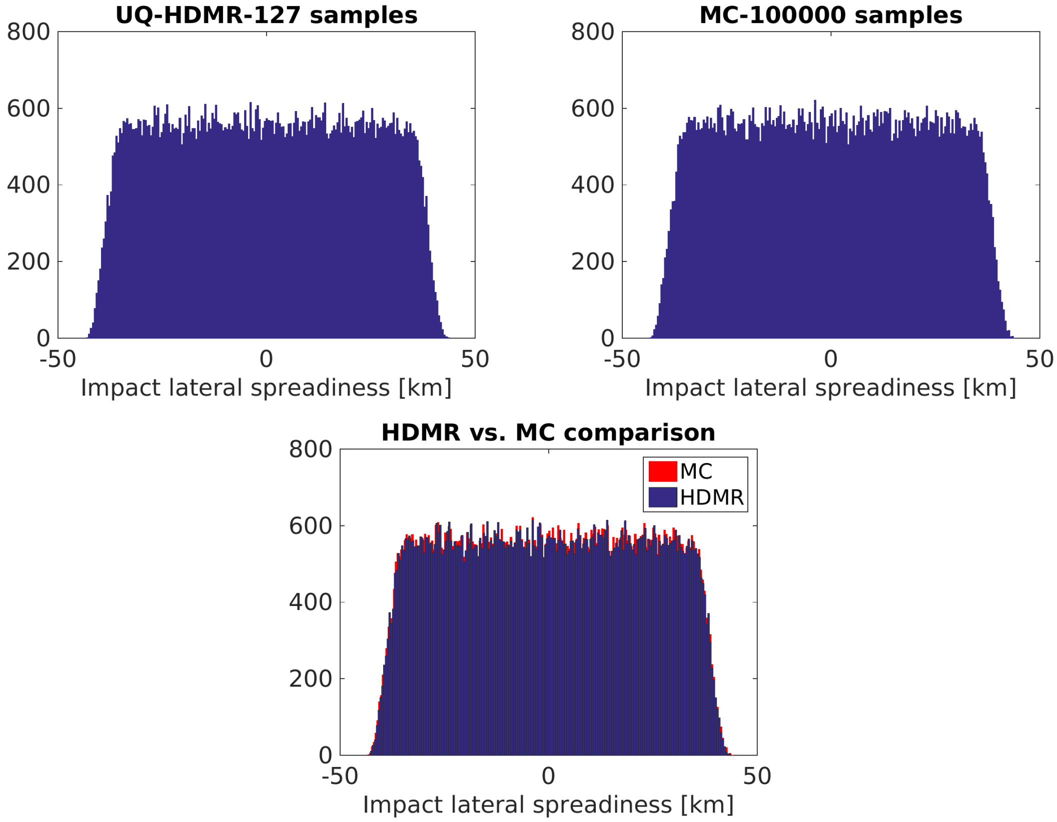
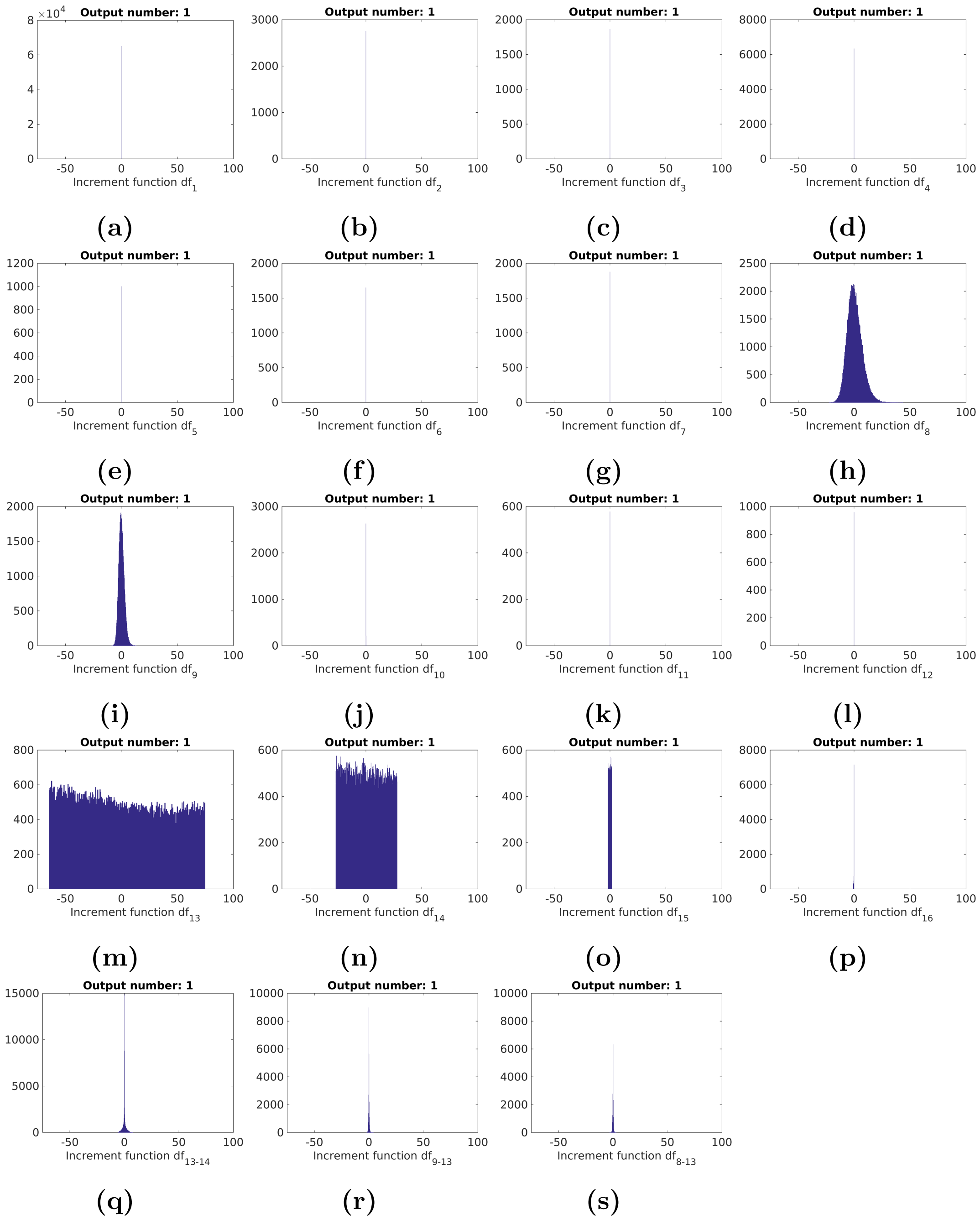


Figure 9.12: Final histograms for the lateral distribution of the controlled steep re-entry case



(a) **(b)** **(c)** **(d)**

(e) **(f)** **(g)** **(h)**

(i) **(j)** **(k)** **(l)**

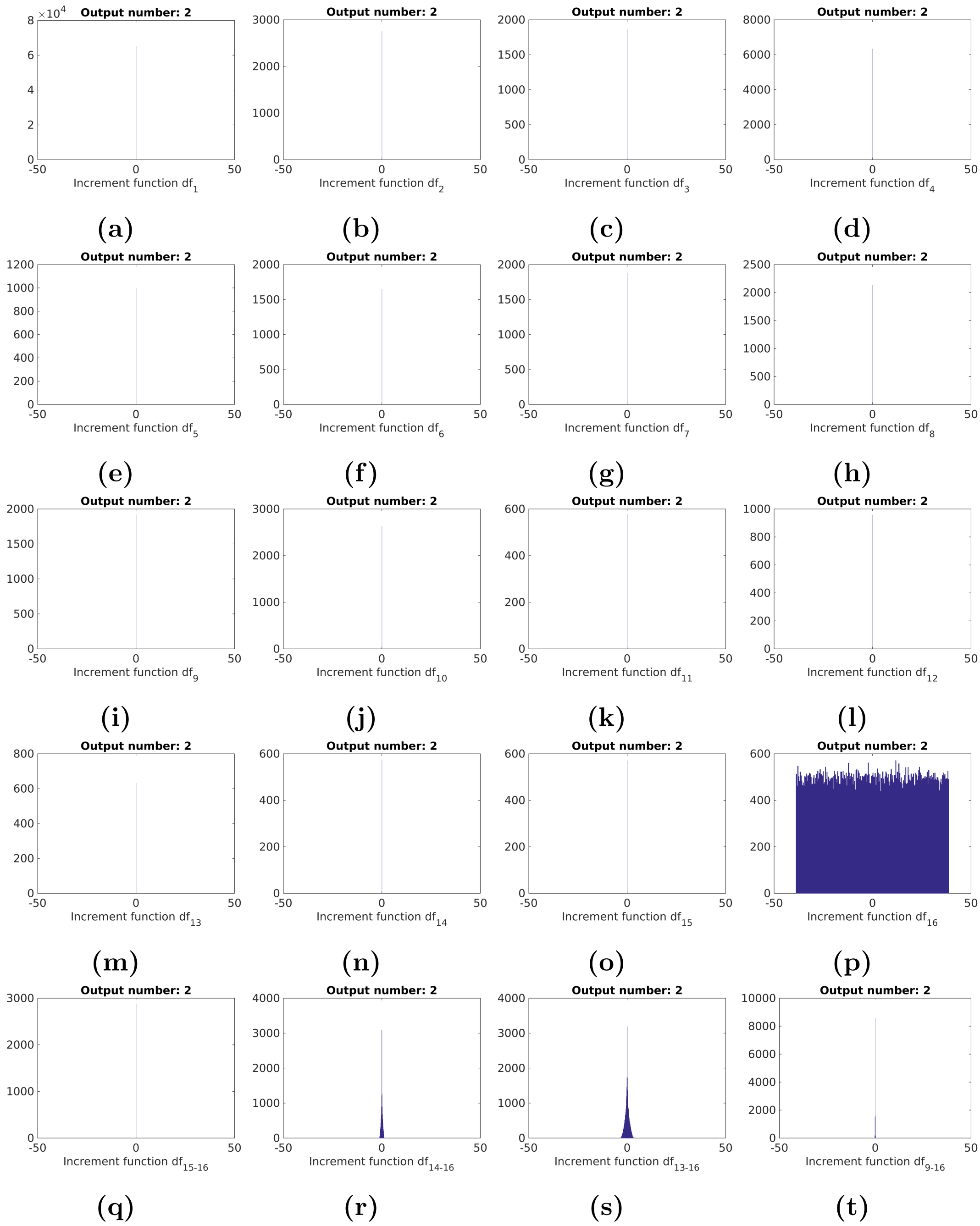
(m) **(n)** **(o)** **(p)**

(q) **(r)** **(s)**

Figure 9.13: Partial histograms for the increment functions of the steep re-entry case for the longitudinal impact distance

Note: Output number: 1 - Longitudinal impact distance

Output number: 2 - Lateral spreadiness



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

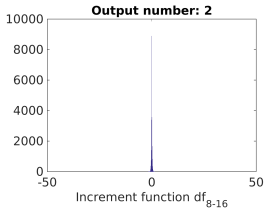
(m) (n) (o) (p)

(q) (r) (s) (t)

Figure 9.14: Partial histograms for the increment functions of the steep re-entry case for the lateral spreadiness

Note: Output number: 1 - Longitudinal impact distance

Output number: 2 - Lateral spreadiness



(u)

Figure 9.14: Partial histograms for the increment functions of the steep re-entry case for the lateral spreadiness

Note: Output number: 1 - Longitudinal impact distance
Output number: 2 - Lateral spreadiness

Chapter 10

Applied example: Gravity Field and Steady-State Ocean Circulation Explorer

In this chapter, the single fidelity and the multi fidelity uncertainty quantification is applied and tested on a hypersonic example. However, the considered test case is computationally expensive and the full Monte Carlo simulation cannot be used for validation purposes. Therefore, the single fidelity approach is used.

The test case is the re-entry of GOCE satellite when it started to fall down from the orbit. Due to the nature of the problem (rarefied flow), Direct Simulation Monte Carlo (DSMC) is used to calculate the properties of the flow and the observed uncertain criterion is the drag coefficient. The model used for the simulation is simplified CAD model, which was provided by ESA.

The chapter is structured in the following way: In the first section, the GOCE satellite is introduced, its main purpose and a short history are given. In the second section, the basic set-up for the DSMC simulation is introduced. This includes a description of the simplified CAD model and mesh. The third section is devoted to the definition of the inputs and uncertain parameters. Also, in this section, the starting conditions for the uncertainty propagation are introduced. In the fourth section, the results and discussion about results are given. The final section presents the conclusions.

10.1 Gravity Field and Steady-State Ocean Circulation Explorer

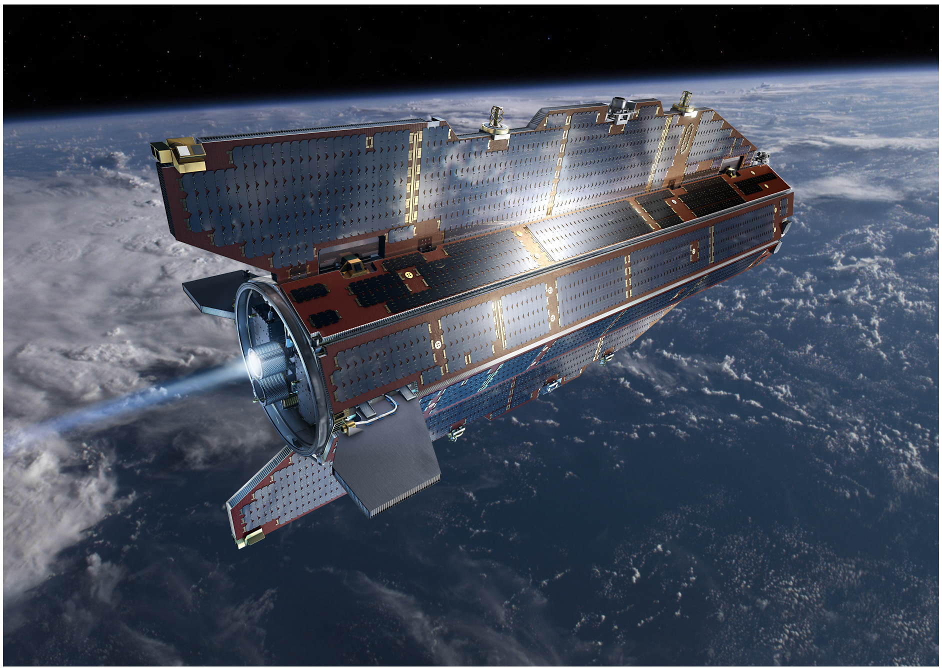


Figure 10.1: GOCE in orbit. Credits: ESA-AOES-Medialab

The Gravity Field and Steady-State Ocean Circulation Explorer (GOCE) was intended to map in unprecedented detail the Earth's gravity field. The spacecraft was mainly equipped with highly sensitive gravity gradiometer consisting of three pairs of accelerometers. This equipment was able to measure the gravitational gradients in all three orthogonal axes.

It brought new insight into oceanic behaviour, which was the main driver for this mission. Using data from other satellites and data from GOCE, scientists were able to track the direction and speed of geostrophic ocean currents. Moreover, the accuracy of the instrument on board of GOCE allowed improving the resolution of the geoid.

The satellite's excellent stability kept GOCE stable as it flew through the upper thermosphere. An ion propulsion system continuously compensated the aerodynamic forces and continuously restoring the path of the spacecraft to the inertial trajectory. Moreover, an ion propulsion was without vibration as it is common for chemical propulsions and therefore, it limits the errors in the gravity gradient measurements due to non-gravitational forces.

However, once the GOCE run out of fuel and the satellite started re-entry to the atmosphere, another part of the mission started. For three weeks, GOCE was flying around earth and measuring all aspects of the space flight, which include the EGG gradiometer data, the orbital data from the SSTI GPS receiver, the attitude information from the star trackers, magnetometer measurements of the Earth's Magnetic Field, currents actuating the magnetic torquers and the linear accelerations sensed by the six ultra-sensitive accelerometers. The uncontrolled re-entry took part on 01:16 CET on 11 November 2013 near the Falkland Islands[1].

10.2 Main study of the re-entry case

During its mission, GOCE gathered a large amount of data, which are extremely useful for validation. However, confronting pure measurements with a single point calculation would mean that all the uncertainty in the measurement is not considered. Therefore, it is necessary to include the uncertainty in the mathematical model and create a robust validation approach. The solution of the flow properties around GOCE represents an expensive simulation and therefore, it is an ideal test case for the proposed single and multi fidelity approach.

The solution technique selected for the hypersonic flows is the DSMC simulation, which solves Boltzmann's equation by using a statistical representation of the particles in the rarefied flow. DSMC approach can provide a high fidelity representation of the molecular gas dynamics, where it simulates the behaviour of each individual molecule. However, this means a serious computational cost. A DSMC simulation can be described in three main phases:

- Mesh generation and particle initialization
- Particle movements and collision simulations
- Average field properties computation

[1] Information obtained at: `http://www.esa.int/Our_Activities/Observing_the_Earth/GOCE` [Date: 20/11/2016]

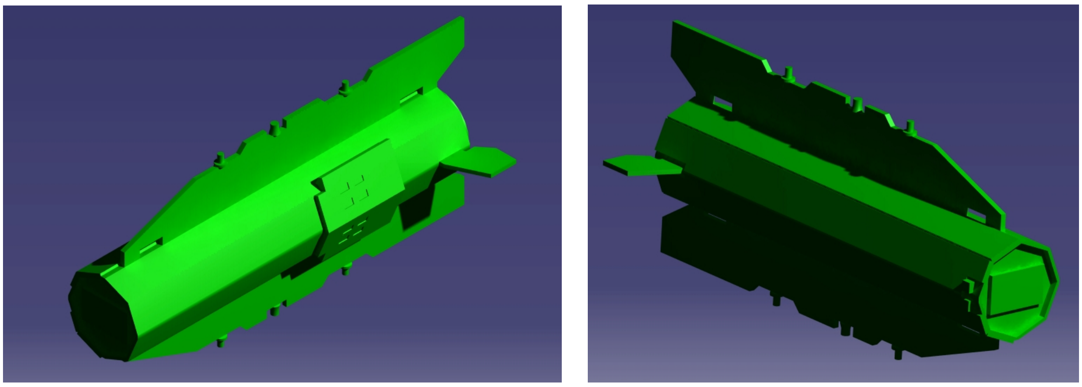


Figure 10.2: Simplified CAD model of GOCE

The DSMC solver selected for the work is the dsmcFoamStrath code (166), developed at the Strathclyde University. For the nature of the problem, a mixed specular-diffusive model along with a variable hard sphere model for the inter-particles collisions and viscosity model is selected. The aerodynamics of the model is solved as a Free Molecular Flow (FMF) problem.

The DSMC computation of the rarefied gas flow is mesh dependent and mesh has to be refined in accordance with an increase in density. Therefore, leading to raising in the number of particles and computational cost. Using free molecular flow allowed to constrain the number of particles inside a cell, which is constrained by the field size and geometry's surface modelled quality. Moreover, the cell size has to be a fraction of the mean free path.

The mesh is generated from the simplified CAD model shown in Fig. 10.2. The CAD model was provided by ESA for the re-entry study. The meshing tool is selected the SnappyHexMesh tool and for visualization is used ParaView. The mesh is generated using only half symmetry model to reduce the computational time, which is possible because only the side slip angle is considered uncertain during the simulation, while other geometries influencing parameters are considered constant. The field size is 8 x 2.5 x 6 $[m]$ and the cell size ratio to the mean free path is of 2 $\sim$ 3. There are constructed two approaches, which differs in mesh properties and mainly in computational time. The properties of the mesh for the high/low fidelity model are summarized in Tab. 10.1 and an example of the mesh for the high fidelity model is showed in Fig. 10.3.

-	Low fidelity model	High fidelity model
Total volume	120	120
Total number of cells	24675	24675
Particles per cell	5	20
Avogadro's number	6.0221409e+23	6.0221409e+23
Reference area for C_D	0.565	0.565
Number of steps to average forces and C_D after convergence	10	6
Wait time for the convergence check	1	0.1
Maximum time to provide the latest result	3000	3000
Percentage max. error	1	0.5

Table 10.1: Mesh properties for the Low/High fidelity DSMC solver

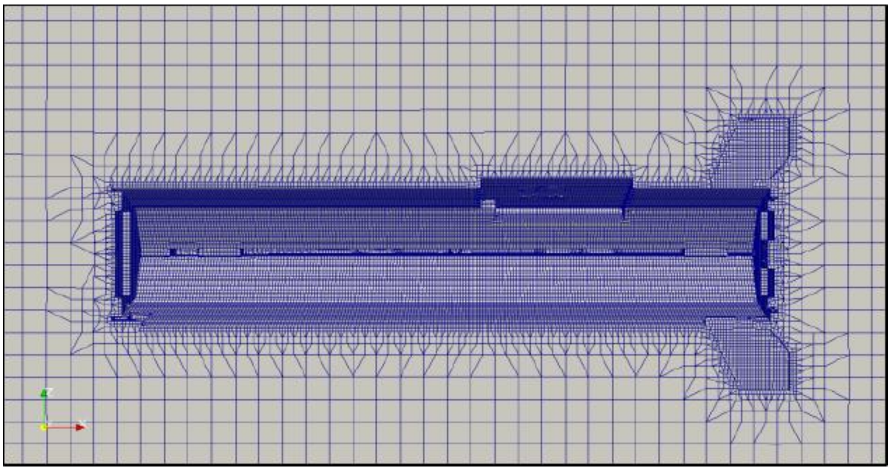


Figure 10.3: Mesh used in the high fidelity DSMC solver

10.3 Input and uncertain parameters of the re-entry case

The orbital data, which are required by the DSMC solver are characterized by NRLMSISE-00 model (167). This model is dependent on altitude, latitude, longitude, a day of the year and magnetic index. The following GOCE's data and initial conditions are used:

- Detailed geometric model (Fig. 10.2 and Fig. 10.3)
- Orbit parameters
- Atomic composition: See Tab. 10.2
- Mass density
- Longitude: 0 deg
- Latitude: 0 deg
- Altitude: 260 km
- Angle of attack: 0 deg

For DSMC simulations, which use the inter-particle collision model and the gas-surface interaction, a set of other parameters is used:

- Wall Temperature: 350 K
- Atomic species diameter: See Tab. 10.2

For the uncertainty propagation, 9 uncertain parameters are selected as the proposed method is suited for the high dimensional problems. The selection of the uncertain parameters is based on an expert knowledge of the problem and it is believed that these parameters are the most influential. The uncertain parameters are summarized in Tab. 10.3. For random variables 1, 2, 7, 8 and 9, the maximum and minimum values of the input distributions are based on expert knowledge. However, the minimum and maximum values for the distributions of variables 3, 4, 5 and 6 are obtained as a yearly average from NRLMSISE-00.

-	Ar	N2	O2	O	NO	N
Mass	66.3e-27	46.5e-27	53.12e-27	26.56e-27	49.88e-27	23.25e-27
Diameter	4.17e-10	4.17e-10	4.07e-10	3.00e-10	4.2e-10	3.00e-10
Rotational degrees of freedom	0	2	2	0	2	0
Vibrational modes	0	1	1	0	1	0
Omega	0.81	0.74	0.77	0.8	0.79	0.8
Alpha	1	1	1	1	1	1
Characteristic vibrational temperature	0	3371	2256	0	2719	0
Dissociation temperature	0	113500	59500	0	75500	0
Char. diss. quantum level	0	33	26	0	27	0

Table 10.2: Gas properties for DSMC solver

ID	Random Variable	Distribution type	Min	Mean	Max
x_1	Speed ratio	Uniform	7.841	8.090	8.356
x_2	Side-slip	Uniform	-5	0	5
x_3	O	Uniform	8.16e+14	1.52e+15	2.45e+15
x_4	N2	Uniform	2.85e+14	4.88e+14	7.95e+14
x_5	O2	Uniform	9.01e+12	1.40e+13	2.15e+13
x_6	Ar	Uniform	8.32e+10	1.58e+11	2.87e+11
x_7	Relax. Coef.	Uniform	5	250	5000
x_8	$\frac{T_{wall}}{T_{flow}}$	Uniform	0.3209	0.3325	0.3440
x_9	Ac. Coef.	Gamma	0.8	0.93	1

Table 10.3: Input distributions for GOCE

10.3.1 Starting conditions for the uncertainty propagation

First, it is necessary to set-up the UQ process. The selected output of interest is considered the drag coefficient, C_D. The random nature of the DCMS code does not allow very high accuracy as the randomness in results influence the convergence criteria. In other words, the randomness of outputs does not allow the convergence residual to be lower than given randomness, i.e. the code would never converge. Therefore, the relative residual is set to 0.03. This allows using the developed code for the stochastic processes because the randomness is suppressed with the high tolerance.

The first UQ propagation is done with the single fidelity approach, where only the high fidelity code is used. The second UQ propagation considers two fidelity codes with the same number of random inputs. Only two fidelity levels are considered as more fidelity levels would only increase the computational burden (see Sec. 8.3.2).

The surrogate models selected for the uncertainty propagation are based on results obtained in Sec. 6.5. For the first order increment functions, the selected models are Lagrange interpolation and PChip interpolation (PChip is selected in a case that the function of interest is discontinuous). The Lagrange interpolation is used from the start and the PChip is considered active after 5 iterations (see Sec. 6.2). For the higher order increment functions, only IPI (Chap. 7) is used. For the multi fidelity error correction function, the spline is used for the first order increment functions and IPI is considered for the higher order increment functions. The spline and IPI are selected due to the fact they are robust and stable interpolants.

10.4 Results and discussion of the uncertainty propagation for GOCE re-entry case

The uncertainty propagation is applied twice. In the first case, the single fidelity approach is used and in the second case, the multi fidelity case is considered. For both cases, results are obtained and compared.

The sensitivity results for the single fidelity approach are summarized in Tab. 10.4 and the graphical representation is given in Fig. 10.4. The multi fidelity sensitivity results are summarized in Tab. 10.5 and a graphical representation is given in Fig. 10.5. The statistical properties (the expected value and the standard deviation) are summarized in Tab. 10.6 and timing of the high/low fidelity model is given in Tab. 10.7. The final histogram for the single fidelity approach and for the multi fidelity is shown in Fig. 10.6. The comparison of the final histograms is given in Fig. 10.7.

Increment function	Partial Mean	Partial Variance	Mean Sensitivity	Variance Sensitivity
1	0.0038	0.0009	0.0179	0.0387
2	0.1780	0.0235	**0.8236**	**0.9432**
3	0.0037	1.3979e-05	0.0175	0.0005
4	0.0035	1.4004e-05	0.0164	0.0005
5	0.0037	1.5135e-05	0.0173	0.0006
6	0.0017	1.1556e-05	0.0080	0.0004
7	0.0014	2.7220e-06	0.0064	0.0001
8	0.0030	1.0143e-05	0.0139	0.0004
9	0.0136	0.0003	0.0629	0.0126
2.9	-0.0016	4.9516e-05	0.0076	0.0019
2.4	-0.0009	4.5219e-06	0.0045	0.0001
2.3	-0.0013	8.3482e-07	0.0015	3.3378e-05
1.2	-0.0004	1.2581e-05	0.0019	0.0005

Table 10.4: Results of the increment functions for GOCE using the single fidelity approach

First, let us discuss results for the single fidelity approach. The C_D is distributed between $C_D = 3.43$ and $C_D = 4.11$ (see Fig. 10.6 on the left). Therefore, we can conclude that DSMC simulation and UQ propagation provided a reliable result. Unfortunately, there is no possibility to validate the result with the full MC simulation as running 100000 would be too huge computational effort (Computational times for the high fidelity model and the low fidelity model are given in Tab. 10.7). Observing the results (Tab. 10.4) and partial histograms (Fig. 10.4), one can understand that the side-slip angle (x_2) is the most influential variable and gives the basic shape of the final distribution, i.e. its elongation around the x abscissa and peak around $C_D = 3.52$. Moreover, the influence of the side-slip angle on the final distribution is very large and positive. Therefore, the distribution is on the right side of the central point ($C_D = 3.47$). One can conclude that

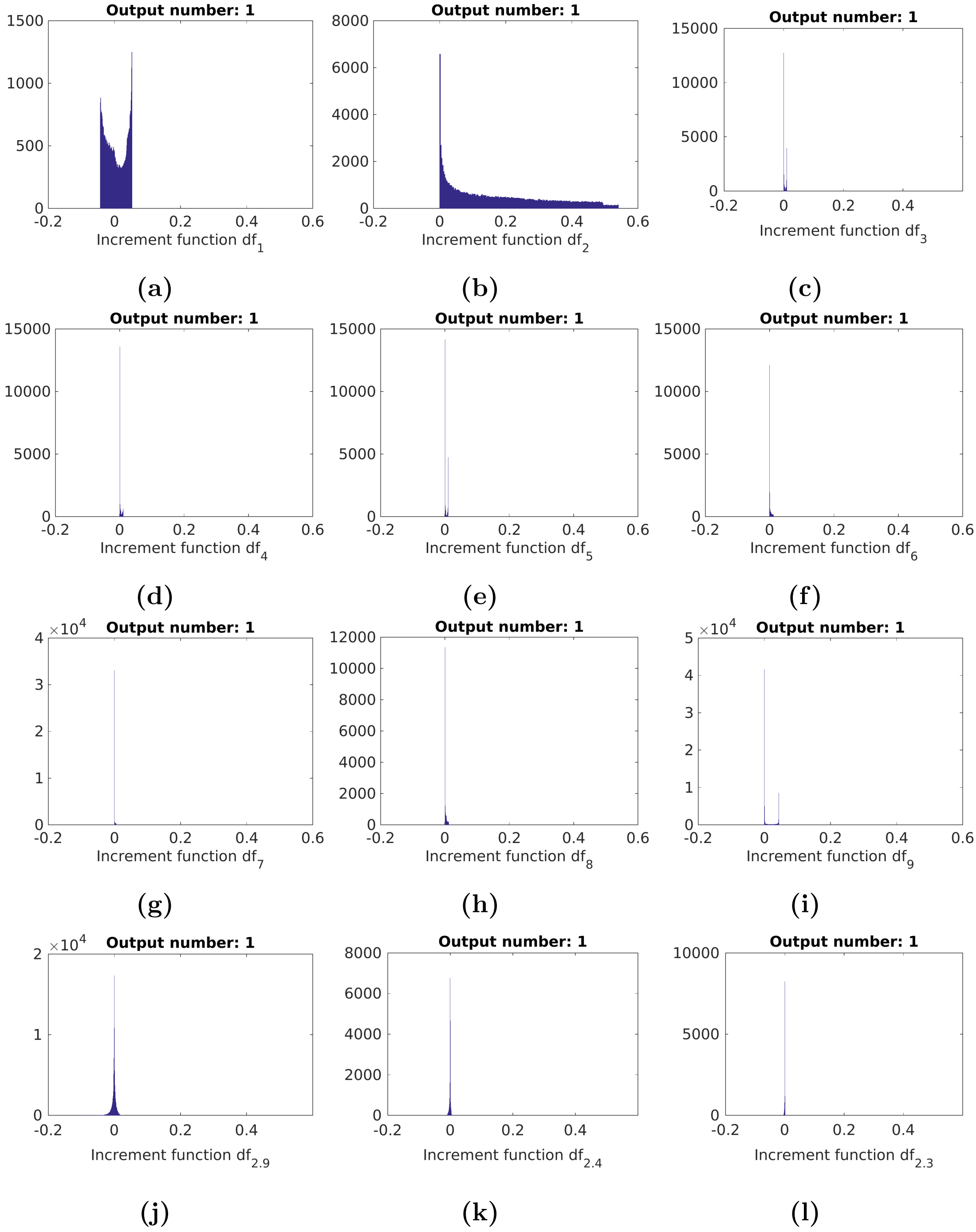
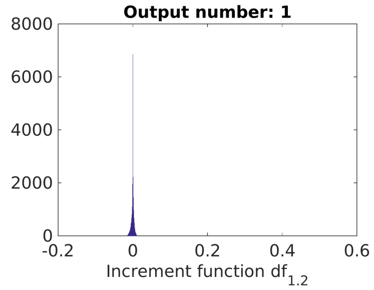


Figure 10.4: Histograms of the partial increment functions for GOCE using the single fidelity approach - part 1

assuming Gaussian distribution and using deterministic approach can lead to a significant error in the final result.

Interesting influence has the interaction effect (Increment function $dF_{2.9}(x_2, x_9)$)



(m)

Figure 10.4: Histograms of the partial increment functions for GOCE using the single fidelity approach - part 2

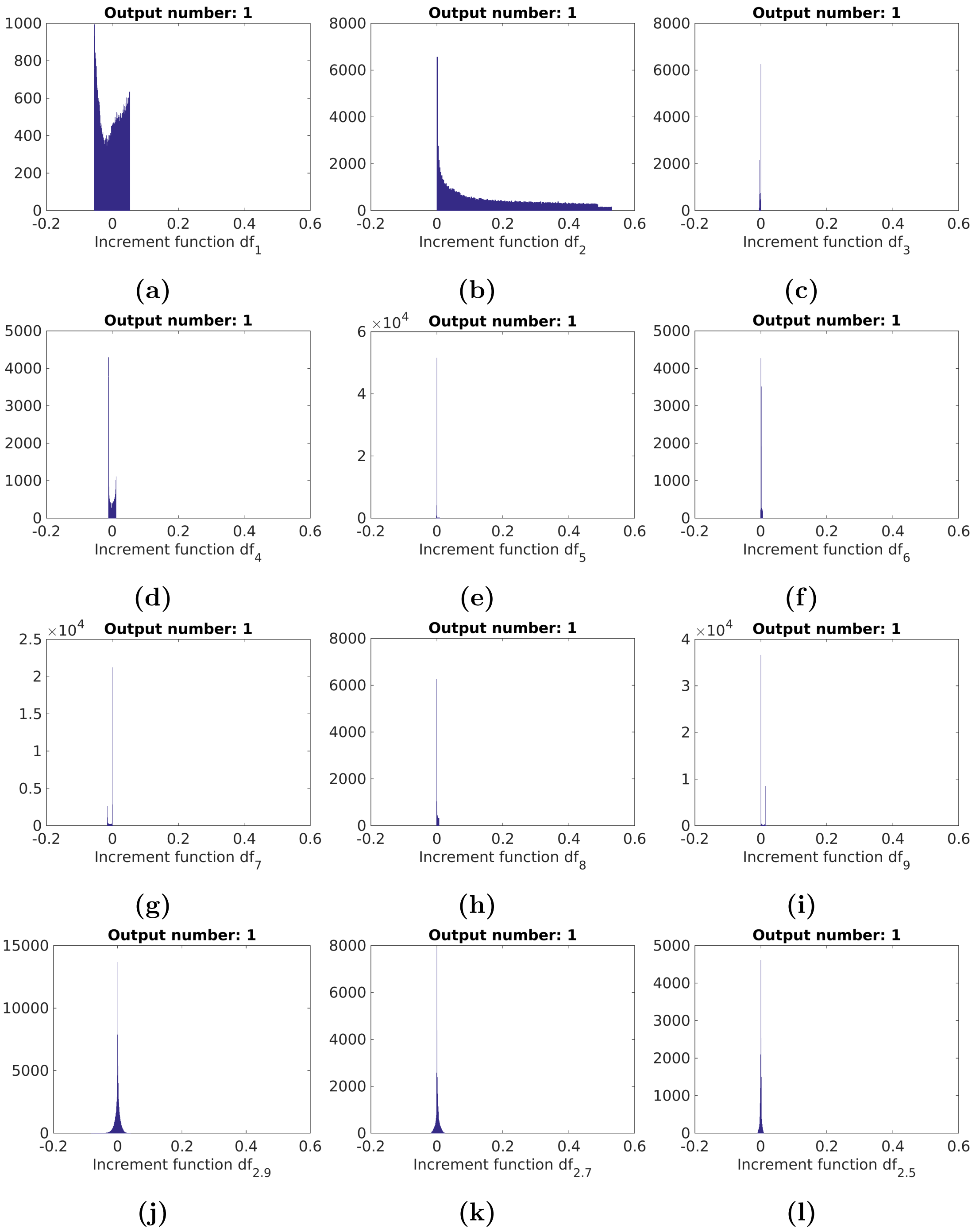
Increment function	Partial Mean	Partial Variance	Mean Sensitivity	Variance Sensitivity
1	-0.0018	0.0011	0.0095	0.0451
2	0.1759	0.0238	**0.8994**	**0.9448**
3	-0.0018	2.1214e-06	0.0094	8.3872e-05
4	-0.0006	5.9122e-05	0.0031	0.0023
5	0.0002	5.0040e-06	0.0014	0.0001
6	0.0013	2.3284e-06	0.0068	9.2057e-05
7	-0.0042	2.9940e-05	0.0216	0.0011
8	0.0016	5.7459e-06	0.0085	0.0002
9	0.0043	3.1284e-05	0.0223	0.0012
2.9	-0.0006	6.6591e-05	0.0033	0.0026
2.7	0.0015	2.9333e-05	0.0078	0.0011
2.5	-0.0001	5.3121e-06	0.0007	0.0002
1.2	0.0010	1.8474e-05	0.0056	0.0007

Table 10.5: Results of the increment functions for GOCE using the multi fidelity approach

Case	Residual set	Function calls	Mean	Standard Deviation
Single F.	0.03	57	3.6794	0.1555
Multi F.	0.03	HF:38 / LF: 48	3.6529	0.1597

Table 10.6: Final results of the UQ propagation for GOCE

of the side-slip angle (x_2) and the accommodation coefficient (x_9), which is slightly tailed to the negative values and therefore, moving C_D value to the left of the cen-



(a) (b) (c)

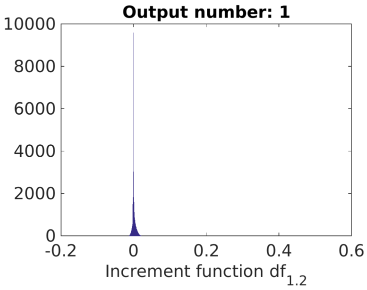
(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 10.5: Histograms of the partial increment functions for GOCE using the multi fidelity approach - part 1

tral sample ($C_D = 3.47$). The interaction effect (increment function $dF_{1.2}(x_1, x_2)$) of the side-slip angle (x_2) and the speed ratio (x_1) brings only uncertainty to the problem due to its symmetricity. However, this interaction is responsible for the



(m)

Figure 10.5: Histograms of the partial increment functions for GOCE using the multi fidelity approach - part 2

Model fidelity	Computational time [Sec.]	Computer center	Cores
High F.	1923	ARCHIE-WeSt	12
Low F.	519	ARCHIE-WeSt	12

Table 10.7: Computational times of the Low/High fidelity model for GOCE
Note: Properties of ARCHIE-WeSt can be found at: www.archie-west.ac.uk

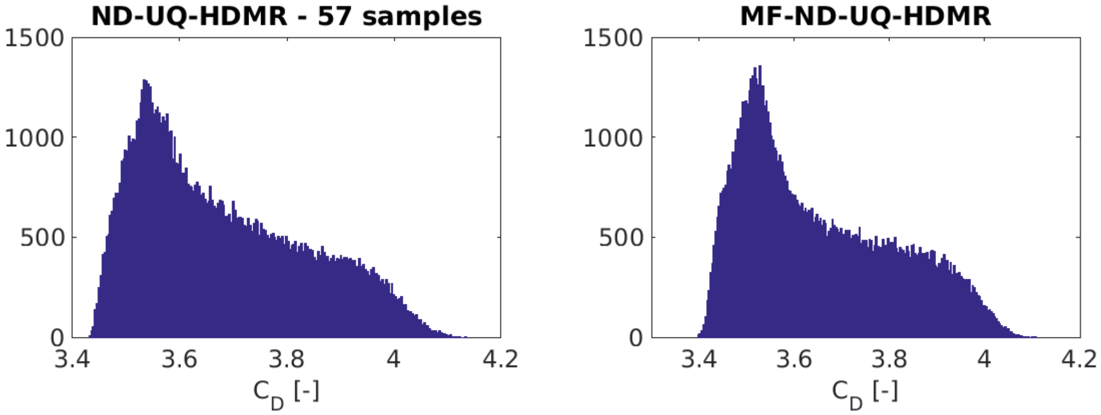


Figure 10.6: Final histogram for the single fidelity approach (left) and the multi fidelity approach (right)

steep slope on the left and tail on the right of the final distribution. The speed ratio (x_1) is not very influential and creates a symmetrical contribution to the final distribution. However, interesting part represents the partial histogram, which shows that the contribution is mainly negative or positive and with low probability around 0. This suggests that the increment function has large steep around the central point and flat regions on its borders. The chemical composition (x_3,

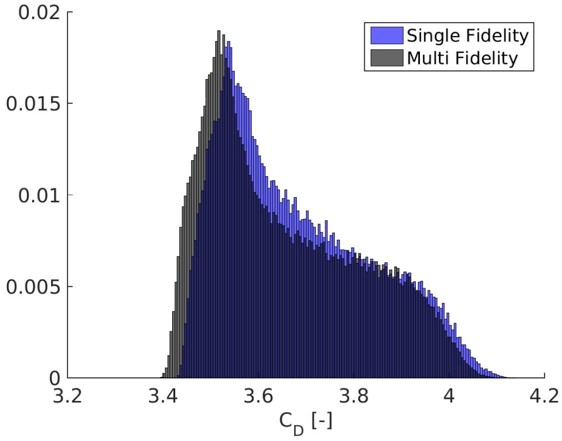


Figure 10.7: Comparison of the final histogram obtained with the single fidelity approach and with the multi fidelity approach

x_4, x_5, x_6), relaxation coefficient (x_7) and temperature ratio (x_8) excluding the argon (x_6) are positive, however, not very influential. The uncertain variable Argon (x_6) creates elongation around the x abscissa. However, its large peak in slight negative values creates a negative contribution to the final PDF, i.e. values on the left of the central point. Nevertheless, the contribution is very small, i.e. negligible. The accommodation coefficient (x_9) represents a small, yet influential contribution to the final distribution. Interesting part represents the large peak on the left side and also, on the right side. The shape of the partial histogram suggests a flat region on the left and right and steep ascent in the center.

Let us now focus on the multi fidelity approach. The relative difference in the mean value is around 0.72% and the relative difference in the standard deviation is around 2.74%. Both values are within the prescribed tolerance of 3% and it can be concluded that the multi fidelity approach works. However, there is a constant bias in the final distribution (see Fig. 10.7), yet the distributions look similar. To explain the bias, we compared the boundary samples for the first order increment functions obtained with the single fidelity approach and with the multi fidelity approach. These samples are in all cases corrected with the high fidelity model (the same model used for the single fidelity approach) and therefore, they must have the same value. Obtained results are summarized in Tab. 10.8, where one can see that for each boundary sample, the final value of the increment function differs. Moreover, some of the values are completely negative, i.e. the increment function gives only negative increment and therefore, move the final distribution to the left of the central point. In other words, the randomness of DSMC created in a single fidelity case positive increment function, while in the multi fidelity case completely negative increment function. Sum of these negative increment functions (see Fig. 10.5) creates the observed bias.

In order to show the randomness, we performed the uncertainty propagation multiple times using the single fidelity approach and the multi fidelity approach. In Fig. 10.8 are showed 4 runs of the multi fidelity code and two runs of the single fidelity code. It can be seen that the variance of both codes is relatively high and the randomness of the expensive code influences the results. Nevertheless, the proposed method converged below the desired accuracy. Moreover, randomness of DSMC is also responsible for the selection of different higher order increment functions (see Tab. 10.4 and Tab. 10.5). The selection process of higher order increment function is based on the influence of samples in the higher order domains (see Sec. 4.2.1) and random nature of these samples influences the importance of given higher order increment functions. However, the multi fidelity approach selected the same important increment functions and therefore, the final result does not differ.

From the partial histogram of the side-slip angle (x_2) for the single fidelity approach (Fig. 10.4) and the multi fidelity approach (Fig. 10.5), one can see that the partial histograms look exactly the same. In this case, the randomness is small and therefore, one can compare given partial histograms. Also, the interaction effect (increment function 2.9) of the side-slip angle (x_2) and the accommodation coefficient (x_9) have the same behaviour. The interaction effect (increment function 1.2) for the side-slip angle (x_2) and the speed ratio (x_1) is slightly different. This can be attributed to the interpolation error of the fidelity error function and also, to the random nature of both fidelity codes. Regarding the other increment functions with a small influence on the final distribution, one cannot compare these partial histograms as these histograms are too much influenced by the random error.

The number of low fidelity samples is smaller than the number of samples in the single fidelity approach (see Tab. 10.6). One would expect that the number of samples should be same or higher than in the single fidelity approach as the function of interest stays the same, i.e. the number of samples required to approximate the function of interest should be the same. This problem can be explained by the randomness of DSMC code because the total number of the function calls will be different with each run. Computing the total computational time for the

Random Variable	Left boundary	Single F. inc.	Multi F. inc.	Right boundary	Single F. inc.	Multi F. inc.
x_1	7.841	0.0529	0.0529	8.356	-0.0430	-0.0550
x_2	-5	0.4940	0.4880	5	0.5410	0.5310
x_3	8.16e+14	0.0040	0	2.4540e+15	0.0059	-0.0049
x_4	2.848e+14	0.0109	0.0110	7.9490e+14	0.0040	-0.0030
x_5	9.012e+12	0.0069	0	2.1530e+13	0.0029	0.0100
x_6	8.317e+10	-0.0010	-0.0009	2.8660e+11	0.0120	0.0060
x_7	5	0.0059	0.0020	5000	0.0049	-0.0009
x_8	0.3209	0.0049	0.0070	0.3440	0.0120	0.0070
x_9	0.7491	0.0219	0.0169	0.9957	0.0069	0.0020

Table 10.8: Difference for the single/multi fidelity approach in boundary samples

single fidelity approach (30.45 computational hours) and for the multi fidelity approach (27.22 computational hours), one can estimate that the multi fidelity approach improved the speed of solution around 10.6%. The small gain in the computational speed (compared to the deterministic case provided in Chap. 8) is mainly due to the small requested accuracy. It is necessary to correct all boundary samples for the first order increment functions and the algorithm does not have enough time to use the low fidelity model as all the non-important increment functions are already converged under prescribed tolerance.

10.5 Conclusion

In this chapter, application of the developed method to the GOCE re-entry case is done. The uncertainty propagation is done in two ways. The first way considers only the single fidelity approach and the second way considers the multi fidelity approach.

In the proposed test case, GOCE is at high altitude and at the hypersonic speed. This represents a fully rarefied flow and therefore, DSMC approach is selected to solve the flow properties. The solver used is the in-house developed dsmcFoam-Strath code applied on the simplified CAD model and the uncertain criterion considered is the drag coefficient. We consider 9 uncertain parameters, where 3 represents the flight properties, 4 are given to the chemical composition of the

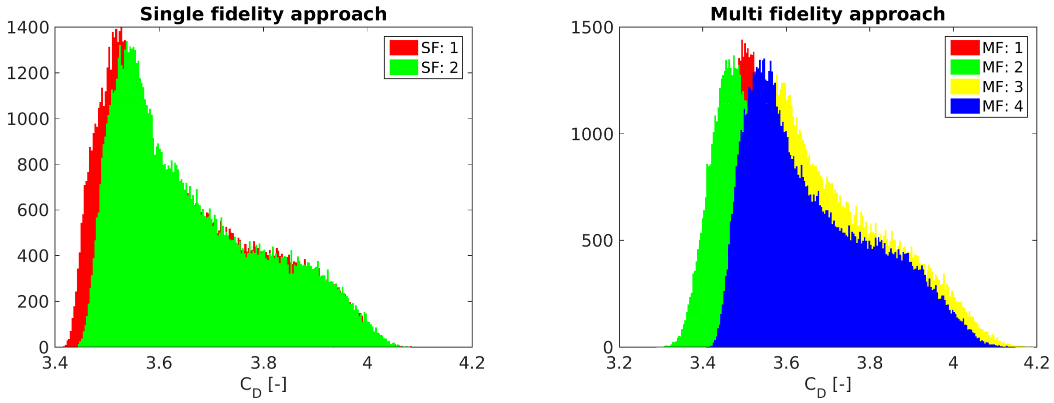


Figure 10.8: Multiple runs of the single fidelity approach (left) and the multi fidelity approach (right)

atmosphere and remaining 2 are parts of the DSMC solver.

A difference between the high fidelity code and the low fidelity code is mainly due to the number of particles in the solution and the convergence residual for the DSMC solver. This allowed obtaining a much faster code, i.e. the low fidelity code. However, this code needs to be corrected with the high fidelity code. Using advantages from both codes, the UQ propagation is performed under shorter time.

The results obtained with the developed method suggest that using simple deterministic approach is not enough and it can lead to serious errors in the final solution. The DSMC code is random in nature and such that it is hard to use approximation techniques for this code. However, even in this case, the developed method provided an accurate result. It was not possible to validate the developed method against the MC simulation due to time constraints.

The validation of the multi fidelity approach is done against the single fidelity approach. As mentioned above, the DSMC code is random to a certain degree and such that the final results can vary. We run additional simulations to confirm the variance of the final results. The multi fidelity approach provided a faster solution, yet the gain is not large. This is due to the low requested accuracy and a small number of total function calls (57). Also, the main influence on the drag coefficient is done by the side-slip angle, more than 90%, and this means that the other variables converge very fast, i.e. the low fidelity model lose its advantage. Nevertheless, the multi fidelity approach provided a reliable result below the desired accuracy.

The final distribution is mainly given by the side-slip angle and its interactions with other variables, which creates small tails on the boundaries of the final distribution. An interesting fact is that most of the first order increment functions are positive, i.e. moving the final distribution to the higher values and therefore, using only deterministic approach will lead to a certain bias. This again proves the necessity of probabilistic modelling.

Chapter 11

Final discussion and conclusion

In this chapter, the overall summary of the developed single fidelity and the multi fidelity method is discussed. The chapter is divided into four sections, where the first section is devoted to the single fidelity method. The second section is given to the discussion about the multi fidelity module. The third section is focused on the future aims and possible plans. The last section is given to the final conclusion.

11.1 Discussion about the uncertainty propagation using the single fidelity model

Many non-intrusive techniques create a simplified model, which mimics the original model. These simplified models are computationally more affordable and sampling techniques, such as the Monte Carlo simulation, can be applied directly to them in order to perform UQ. These simplified models are called surrogate models, and they are constructed by using the expensive code as a black-box tool. However, the construction of these surrogate models is very complex and delicate.

We developed a new non-intrusive uncertainty propagation method. First, the stochastic domain is decomposed into smaller domains. This is possible due to the derivative equation (see Chap. 2), which decomposes the stochastic domain with an application of the partial derivatives. Using these partial derivatives, the increment function is constructed. The increment function is an independent contribution to the final model and it is handled as a separate problem. Each one of these increment functions is approximated separately and the final model is constructed as a sum of these increment functions. This represents a large simplification to the approximation methodology as it is easier to approximate series of small (low dimensional) problems rather than one large problem. Each one of these increment functions follows a certain set of rules. These rules allow a large simplification of the problem of interest and therefore, mitigate the computational burden.

The important aspect represents the statistical properties of the increment function. The partial mean value obtained from the increment function represents the movement of the final mean value and the partial variance represents the importance of the increment function, i.e. how big the influence of the increment function on the final distribution is. These values are obtained with the MC simulation, which is applied directly to the increment function (or its surrogate model). This step allows visualizing the partial histogram, representing the statistical contribution to the final statistics. The main advantage of the partial histogram is that it provides deeper insight into the final statistics and behaviour of the function. Moreover, using the conclusions obtained from the derivative equation, the behaviour of the underlying increment function can be established. This is extremely useful in higher domains, where the increment function cannot be plotted, allowing the user to focus on the problematic parts of the stochastic domain.

Let us now discuss the conclusions obtained with the Derivative Equation (DE). The first important conclusion obtained with the DE is related to the shape of PDF, which depends only on the partial derivatives and input distributions. The second important conclusion is about the tails of the final distribution: if the function has strong interaction effects, i.e. the higher order increment functions have a large influence, the final probability distribution is heavy-tailed. This is an important aspect of the statistical measurement as when the measurements are heavy-tailed, models that consider this aspect must be taken into account.

To provide an efficient method for an uncertainty quantification, usually, the Design of Experiments (DOE) plays an important part. However, this work focuses on an adaptive sampling, where the basic idea of the adaptive scheme is based on the comparison of interpolants between different iterations. It is divided into several steps. The first step represents the error comparison function, which compares the change in the function. The second step represents the position function, which considers the position of previous samples. The combination of these two steps creates a natural exploitation vs. exploration approach, where simple functions are uniformly sampled and functions with complicated regions are mainly sampled in these regions. The last point represents the input probability distribution, which behaves as a weight function and such that emphasizes regions with large probability. These three points let to the implementation of a function, whose maximum indicates the new sampling location. A very important property is that this proposed adaptive scheme naturally takes into account the behaviour of the interpolation technique. In other words, in the case of Lagrange interpolation, the samples are added to the boundaries to stabilize the interpolant, where on the other hand, in the case of piecewise interpolation, samples are more evenly distributed.

The previously mentioned error comparison function could be used as the convergence measurement, but that is impractical. Therefore, the total mean (mean of the final model) and the total variance (variance of the final model) are used as convergence criteria. with a two steps approach.

The first step represents the local convergence, where only the convergence of the increment function is checked. Based on the sensitivity analysis provided in each iteration step, the algorithm can easily decide which increment functions need more samples. This provides an optimal sampling scheme for the problem of interest. To further improve the convergence, the logical convergence is introduced. This convergence takes into account only the behaviour of the increment function and helps to avoid unnecessary sampling if one of the increment functions is diverging.

The second step represents the global convergence, which protects the algorithm from the premature convergence, i.e. if the final mean and the final variance is not converged under prescribed tolerance, the problem still needs additional samples. Both steps ensure that the final model is converged under pre-scribed tolerance.

The main growth of samples is due to high order interactions and such that it is necessary to restrict the sampling of higher order domains as much as possible. The commonly used approach using HDMR modelling is to stop adding the higher order increment functions in the prescribed order. However, this approach can lead to serious problems and it can lead to completely wrong results. Moreover, it is not very efficient in a case of a large number of stochastic variables. Therefore, the prediction scheme is developed. The prediction scheme consists of three main aspects.

The first aspect is the deduction approach, which deduces what increment functions should be used. It is based on an inverse logic and with very few properly placed samples, it can estimate the influence of the higher order increment functions. The second aspect represents the prediction algorithm, which is an empirical approach to the selection process. The deduction algorithm selects a large portion of the important higher order increment functions yet some of them do not need to be very important. This is due to a small number of samples used for the deduction algorithm and therefore, the prediction is done to further reduce the number of increment functions. If the increment function has a small predicted effect, it is neglected from the sampling process. The prediction algorithm is based on conclusions obtained from DE and reflects the behaviour of the stochastic domain. However, it should be kept in mind that it is still a prediction and therefore, results from the prediction should not be used in any other way. The last part of the prediction scheme is the neglection algorithm, which is again based on conclusions obtained from DE. It samples the selected increment function with a small number of samples, e.g. 1, and estimates the value of the increment function. In the case of small influence, the increment function is neglected.

An important part of the prediction scheme is the estimation of the neglection residual, i.e., how small influence the increment function must have to be neglected. The residuals are selected on a basis of a statistical influence of the neglected increment functions. In other words, we build a linear model which represents the influence of the neglected increment functions and propagating the statistics over this model defines the required residual for the selection process. An interesting fact is that the model has to be linear, which is a result of the conclusions obtained in Sec. 4.3.

The prediction scheme is also responsible for the convergence of the whole methodology, i.e. the final model. The final model is convergent when there are no more important increment functions to add. This aspect represents the main difference between commonly used convergence approaches and our method.

Once the portion of the important higher order increment functions is known, the higher order adaptive scheme can be applied. The higher adaptive scheme represents an extension of the first order adaptive scheme and it is based on the same logic. The error comparison function and the position function are slightly modified to take into account the higher dimensions. However, the largest difference represents the requirement of the interpolation technique. In higher dimensions, some interpolation techniques have special requirements, which needs to be fulfilled. This step makes the position of new samples not optimal yet it is necessary in order to have stable and accurate interpolation technique. Therefore, one more step is added to the sampling process in order to fulfil the condition required by an interpolant. For example, in the case of Lagrange interpolation, the tensor grid is constructed.

The convergence process is also, almost identical to the first order convergence process. The small difference is an estimation of the total variance and the total mean considered, which are considered in the convergence process. This step allows a slightly more accurate solution to the influence of the increment function.

Altogether, the whole procedure creates an efficient interpolation approach. However, this approach is not fully robust. There can be found functions, where the whole algorithm diverges due to the interpolation technique. To overcome this problem, the multi surrogate interpolation technique is developed. The proposed interpolation technique is based on a simple idea of project management, where the main algorithm behaves as a project manager and selects the most accurate and fast solution to the problem. The mathematical model resembles the Kriging surrogate model, yet it has some several distinctions. The process of weight selection is iterative in nature and requires an iterative (adaptive) scheme. Therefore, a combination of the adaptive scheme and the multi surrogate model is done. However, the focus of the developed technique is not on efficiency but mainly to provide a more robust interpolant. Nevertheless, the proposed technique is still very efficient. The robustness of the code is shown on many examples and important conclusions are obtained. These conclusions are later on used in the multi fidelity approach.

The last part of the single fidelity approach is given to the new interpolation technique - Independent Polynomial Interpolation (IPI). This technique is developed in order to have a technique, which would suit our problems. In other words, which would be capable to handle adaptive scheme, the proposed HDMR approach, and high dimensional discontinuities. IPI resemble the spline technique in the higher dimensions yet in discontinues problems it is more efficient. On the other hand, for continues cases, it is slightly worse. IPI represents a new way for interpolation processes and with further research, it can be much more efficient than the previously mentioned spline.

11.2 Discussion about the multi fidelity approach

In this work, the multi fidelity approach is considered as a separate module for the single fidelity UQ propagation tool. In other words, the whole approach can be used in the single fidelity way or in the multi fidelity way. This makes the developed method very practical.

One of the main problems of the multi fidelity modelling represents the selection process of the low fidelity models. The power ratio theory tries to answer this question. The power of the code of interest is represented by a simple equation, which considers two main aspects. The first aspect represents the accuracy of the lower fidelity model and the second aspect represents the time of the lower fidelity model. This theory provides simple comparison criteria for the low fidelity model selection. The nature of the method allows using various accuracy criteria for comparison such as $L2$ error. However, in this work, only the mean value and the variance is used as an accuracy comparison criteria. The model with the highest power ratio represents the best low fidelity model for a given problem and such that it is selected for replacement of the high fidelity model. However, it is necessary to correct the low fidelity model to reduce the possible errors.

Another aspect of the power ratio represents the line of efficiency, which separates the efficient low fidelity models from the inefficient ones. The inefficient models are automatically neglected from the process as they do not provide any benefit to the problem. Unfortunately, this concept does not take into account the complexity of correction of the low fidelity model. In other words, if the low fidelity model needs too many corrections, i.e. high fidelity model function calls, the low fidelity models lose its advantage and the actual computation time is higher. Nevertheless, on provided examples (Borehole model and GOCE), it is showed that the proposed concept works.

The power ratio helps to select the best low fidelity model. However, the important question is: which increment functions need to be corrected. For the first order increment functions, one cannot assume anything about the low fidelity model and to get the first insight into accuracy and efficiency of the low fidelity model, it is necessary to correct the boundary samples. This allows to create a simple correction function and establish the first prediction for the low fidelity model. This step is done for all the lower fidelity models and the best model is selected. However, one can quickly realize that in the case of a large number of the low fidelity models, the whole process loses its advantage. Therefore, it is suggested to keep the number of the low fidelity models low, e.g. 1 or 2.

In order to correct the low fidelity model, the additive correction is used. We restrict ourselves to the additive correction due to its easy application to the HDMR model and proven functionality. Moreover, the additive approach can be easily used in combination with integration. On the other hand, using the multiplicative approach would lead to problems with integration parts in the higher dimensions and such that it is neglected from this work. The additive correction is implemented via the Fidelity Error Increment Function (FEIF), which represents the additive correction for the HDMR approach. An important aspect is that it follows all the rules defined for the HDMR model. For the interpolation purposes, it is suggested to use a piecewise interpolation technique as these techniques are not prone to oscillations.

The Fidelity Error Increment Function is used in the error prediction process. This approach tries to predict the error in the mean value and in the variance if the low fidelity model is used. It is based on the linear prediction of the error between the fidelity models from the position, where the correction is known. In other words, we assume that the error of the low fidelity model can be predicted with the linear extrapolation around known correction. Using this prediction, the error on the mean value and the variance can be established. This is used as a criterion for the decision process if the correction function needs more high fidelity samples. For our purposes, the predicted error is transformed to the statistical error, however, the predicted error can be used to estimate other properties such as estimation of the error at one particular position or visualization of the fidelity error (for example Fig. 8.3).

Another important aspect is the selection of the higher order increment functions. Correcting all the higher order increment functions would lead to a large waste of samples and therefore, the prediction algorithm is applied to all the low fidelity models. The prediction algorithm compares the selected increment functions and if the increment function is presented in both fidelity models, then the low fidelity model is used for the prediction. Note that in the case of the single fidelity approach, the prediction algorithm (see Chap. 4) is not a necessity, however, in the case of the multi fidelity approach, it represents a compulsory thing.

Once the higher order increment functions are selected, the algorithm uses only the low fidelity model to estimate the influence of the higher order increment functions. One cannot correct all the higher order increment functions as the whole process of the multi fidelity modelling would lose its advantage. Therefore, only the important increment functions are corrected. This approach is based on an assumption that the low fidelity model has the same trends. It should be kept in mind that errors of uncorrected increment functions are propagated to the final solution, yet it is assumed that the influence of the uncorrected functions is extremely low and the final solution is not influenced much. However, if the uncorrected increment function does not behave in a similar manner such as the high fidelity model, the error can influence the final solution. Nevertheless, this problem is caught with the prediction scheme.

11.3 Future work and aims

In this work, a new scheme for the uncertainty modelling is presented. This scheme is brand new and requires a rigorous testing, which will take part in near future. Considered future testing examples are going to range from simple analytic models to complex stochastic methods such as DSMC.

The adaptive scheme is currently independent of the interpolation technique. Therefore, next focus will be given to the adaptive connection of the interpolating method. Also, convergence criteria will be coupled with the interpolation technique. For example, piecewise interpolation needs more samples to prove convergent than the global ones.

The prediction approach represents a new thing, yet it is proven to be working efficiently. However, it still offers large possibilities for extensions. Therefore, the further potential improvements will be investigated. Another step will be focused on usage of the results from the prediction approach to the interpolation process. In other words, construct an interpolation scheme from the samples used for the prediction. This should lead to improvement of the overall efficiency of the proposed method.

The multi surrogate modelling technique is developed to make a robust approach to the uncertainty propagation. However, it is still dependent on the database of provided models. This database will be extended in the future steps to other interpolants and approximation techniques. Also, an ideal combination of surrogate models will be established.

Another aspect of the multi surrogate approach represents the starting iteration for each surrogate model, i.e. when the surrogate model becomes active in the approximation process. In this work, the user sets the starting iteration based on his/her personal experience and this is not an optimal approach. Therefore, an automatic selection approach will be developed in the next steps.

IPI is a new approach to the piecewise interpolation and it opens new possibilities. One of the possible improvement represents the selection of the derivative scheme, which can greatly improve its efficiency in the case of discontinues functions. Also, selection and distance of the weight function can improve accuracy in a case of continues function. Therefore, a combination of the adaptive scheme and IPI using new derivative schemes and weight functions will be investigated.

The last part represents the multi fidelity module. The efficiency line separating the efficient model from the inefficient ones in the power ratio theory will be closely investigated as it provides interesting insight into multi fidelity modelling. Another aspect represents the selection criteria for the higher order increment functions. A more complex selection process is going to be investigated, which will take into account differences between estimated influences.

The surrogate model considered for correction is going to be investigated more deeply. It is necessary to develop an adaptive scheme, which will adaptively select the best model. This should improve the correction of samples from the low fidelity models.

11.4 Final Conclusion

In this work, a new way of uncertainty propagation and multi fidelity modelling is introduced. The proposed method is tested on two examples and it proved to be very efficient. Moreover, the proposed method can be applied to other uncertainty propagation problems.

Important conclusions about statistical propagation are obtained with the new approach. These conclusions are proved on applied examples and they can be used in various other scientific disciplines.

The derivative equation derived in this work represents an important piece of information for the high dimensional modelling. It shows in clear and understandable way, how the information propagates from the lower domains to the higher ones. However, further research into aspects of the derivative equation will be necessary.

A new way of sensitive analysis is introduced. The user can understand how each variable influences the final result and partial histograms show the influence of given variable over the stochastic domain. This aspect represents an important step in the statistical modelling.

Many of the proposed techniques can be used independently, however, the main strength of the proposed method lies in the combination of all the approaches altogether. The non-intrusive uncertainty propagation is suited for various kind of problems and such that it is not restricted to only space applications. However, this needs to be proved with additional examples. Nevertheless, the main application still remains the uncertainty quantification for re-entry propagations and hypersonic flows.

Bibliography

[1] Fabio Nobile, Raul Tempone, and Clayton Webster. **The Analysis of a Sparse Grid Stochastic Collocation Method for Partial Differential Equations with High-Dimensional Random Input Data**. Technical report, Sandia National Laboratories, 2007. xvii, 114

[2] P. M. Mehta, E. Minisci, and M. Vasile. **Break-up Modeling and Trajectory Simulation under Uncertainty for Asteroids**. In *4th IAA Planetary Defense Conference*, Frascati, Roma, 13-17 April 2015. 3, 222

[3] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford university, December 1997. 4

[4] L. W. T. Ng and M. S. Eldred. **Multifidelity Uncertainty Quantification Using Non-Intrusive Polynomial Chaos and Stochastic Collocation**. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, April 2012. 4, 32, 185

[5] Jerome Morio. **Global and local sensitivity analysis methods for a physical system**. *European Journal of Physics*, **32**(6):1577–1586, 2012. 10

[6] A. Saltelli, Marco Ratto, Terry Andres, Francesca Campolongo, Jessica Cariboni, Debora Gatelli, Michaela Saisana, and Stefano Tarantola. *Global Sensitivity Analysis: The Primer*. John Wiley and sons, January 2008. 10, 11

[7] A. Saltelli, Stefano Tarantola, Francesca Campolongo, and Marco Ratto. *Sensitivity Analysis in Practice: A Guide to Assessing Scientific Models*. John Wiley and sons, 2004. 10, 11

[8] Andrea Saltell and Ricardo Bolado. **An alternative way to compute Fourier amplitude sensitivity test**. *Computational Statistics and Data Analysis*, 1998. 10

[9] K. T. Fang, Runze Li, and Agus Sudjianto. *Design and Modeling for Computer Experiments*. Chapman and Hall/CRC press, New York, 2006. 10, 11, 32, 33, 97

[10] J. P. C. Kleijnen and J. C. Helton. **Statistical analyses of scatterplots to identify important factors in large-scale simulations, 1: Review and comparison of techniques**. *Reliability Engineering and System Safet*, **65**:147–185, 1999. 10

[11] J. P. C. Kleijnen and J. C. Helton. **Statistical analyses of scatterplots to identify important factors in large-scale simulations, 2: robustness of techniques**. *Reliability Engineering and System Safet*, **65**:187197, 1999. 10

[12] Max D. Morris. **Factorial Sampling Plans for Preliminary Computational Experiments**. *Technometrics*, **33**(2):161–174, 1991. 10

[13] Eric W. Weisstein. **Web Diagram**, 2014. http://mathworld.wolfram.com/WebDiagram.html. 11

[14] Giles Hooker. **Discovering additive structure in black box functions**. In *KDD 04 Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 575–580. ACM New York, NY, USA, 2004. 11, 12

[15] Michael Eldred and John Burkardt. **Comparison of Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Quantification**. In *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, Orlando, Florida, January 2009. American Institute of Aeronautics and Astronautics. 11, 32, 113

[16] Michael Eldred. **Recent Advances in Non-Intrusive Polynomial Chaos and Stochastic Collocation Methods for Uncertainty Analysis and Design**. In *50th AIAA /ASME /ASCE /AHS /ASC Structures, Structural Dynamics, and Materials Conference*, Palm Springs, California, May 2009. 11, 32, 56, 113

[17] Norbert Wiener. **The homogeneous chaos**. *America Journal of Mathematics*, **60**(4):897 – 936, October 1938. 11, 32

[18] Kanali Togawa, Andrea Benigni, and Antonello Monti. **Advantages and Challenges of Non intrusive Polynomial Chaos Theory**. In Roy Crosbie, editor, *Proceedings of the 2011 Grand Challenges on Modeling and Simulation Conference*, pages 30–35, Vista, CA, June 2011. Society for Modeling and Simulation International. 11, 32, 56

[19] Haiyan Cheng and Adrian Sandu. **Collocation Least-squares Polynomial Chaos Method**. In *SpringSim '10, Proceedings of the 2010 Spring Simulation Multiconference*, San Diego, CA, 2010. Society for Computer Simulation International. 11, 32, 33, 56

[20] Serhat Hosder, Robert Walters, and Rafael Perez. **A Non-Intrusive Polynomial Chaos Method For Uncertainty Propagation in CFD Simulations**. In *44th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 2006. 11, 32

[21] Michal Branicki and A.J. Majda. **Fundamental Limitations of Polynomial Chaos for Uncertainty Quantification in Systems with Intermittent Instabilities**. *Communications in Mathematical Sciences*, **11**, 2013. 11, 32

[22] M. S. Eldred, C. G. Webster, and P. G. Constantine. **Evaluation of Non-Intrusive Approaches for Wiener-Askey Generalized Polynomial Chaos**. In *The 49th AIAA /ASME /ASCE /AHS /ASC Structures, Structural Dynamics, and Materials Conference*, 2008. 11, 32, 49, 54

[23] Jaehun Lee and Jang Hyuk Kwon. **On the use of Kriging in the interpolation of fluid-structure interaction analysis**. *Japan society of computational fluid dynamics*, **16**, 2008. 11, 32

[24] Alexander I. J. Forrester, Andras Sobester, and Andy J. Keane. *Engineering Design via Surrogate Modelling.* A John Wiley and Sons Ltd., 2008. 11, 32, 86

[25] Bruno Sudret. **Global sensitivity analysis using polynomial chaos expansions**. *Reliability Engineering and system safety*, 2008. 11

[26] R. E. Bellman. *Adaptive Control Processes: A Guided Tour.* Princeton University Press, New Jersey, NJ, 1961. 11, 32

[27] Ray-Bing Chen, Dai-Ni Hsieh, Ying Hung, and Weichung Wang. **Optimizing Latin hypercube designs by particle swarm**. *Statistics and Computing*, **23**:663–676, September 2013. 11, 33

[28] Serhat Hosder, Robert W. Walters, and Michael Balch. **Efficient Sampling for Non-Intrusive Polynomial Chaos Applications with Multiple Uncertain Input Variables**. In *48th AIAA /ASME /ASCE /AHS /ASC Structures, Structural Dynamics, and Materials Conference*, Honolulu, HI, April 2007. 11, 33

[29] Dirk Pfluger. *Spatially Adaptive Sparse Grids for High-Dimensional problem.* Verlag Dr. Hut, Munchen, 2010. 11

[30] Thomas Gerstner and Michael Griebel. **Numerical integration using sparse grids**. *Numerical Algorithms*, **18**:209–232, 1998. 11, 33

[31] Volker Barthelmann, Erich Novak, and Klaus Ritter. **High dimensional polynomial interpolation on sparse grids**. *Advances in Computational Mathematics*, **12**:273–288, 2000. 11, 21, 33

[32] Herschel Rabitz, Omer F. Alis, Jeffrey Shorter, and Kyurhee Shim. **Efficient input-output model representations**. *Journal of Physical Chemistry*, **117**:11–20, 1999. 11, 12, 18, 24

[33] Jeffrey A. Shorter, Precila C. Ip, and Herschel A. Rabitz. **An Efficient Chemical Kinetics Solver Using High Dimensional Model Representation**. *J. Phys. Chem.*, **103**:7192–7198, 1999. 11

[34] Genyuan Lia, Sheng-Wei Wanga, Herschel Rabitza, Sookyun Wangb, and Peter Jaffe. **Global uncertainty assessments by high dimensional model representations (HDMR)**. *Chemical Engineering Science*, **57**, 2002. 11

[35] Xiang Ma. *An Efficient Computational Framework for Uncertainty Quantification in Multiscale Systems.* PhD thesis, Cornell University, 2011. 11, 24, 33, 161, 175

[36] Kunkun Tang, Pietro M. Congedo, and Remi Abgral. **Sensitivity analysis using anchored ANOVA expansion and high order moments computation**, May 2014. 11, 13, 20

[37] Genyuan Li, Maxim Artamonov, Herschel Rabitz, Sheng wei Wang, Panos G. Georgopoulos, and Metin Demiralp. **High-Dimensional Model Representations Generated from Low Order Termslp-RS-HDMR**. *Journal of Computational Chemistry*, **24**(5), 2003. 11, 13

[38] M. Alper Tunga. **An approximation method to model multivariate interpolation problems: Indexing HDMR**. *Mathematical and Computer Modelling*, **53**:1970–1982, 2011. 11

[39] M. Alper Tunga and Metin Demiralp. **Hybrid high dimensional model representation (HHDMR) on the partitioned data**. *Journal of Computational and Applied Mathematics*, **185**, 2006. 11

[40] Zhongqiang Zhang, Minseok CHoi, and George Em Karniadakis. **Error Estimates for The ANOVA Method With Polynomial Chaos Interpolation: Tensor Product Functions**. *SIAM journal o scientific computation*, **34**(2):1165–1186, 2012. 11, 13, 22, 55

[41] I. M. Sobol. **Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates**. *Mathematics and Computers in Simulation*, **55**:271–280, 2001. 12, 16, 17

[42] G. Geraci, P.M. Congedo, and G. Iaccarino. **Decomposing high-order statistics for sensitivity analysis**. *Center for Turbulence Research Annual Research Briefs - Nutno dodelat*, 2014. 17

[43] Jens Prager, Habib N. Najm, Khachik Sargsyan, Cosmin Safta, and William J. Pitz. **Uncertainty quantification of reaction mechanisms accounting for correlations introduced by rate rules and fitted Arrhenius parameters**. *Combustion and Flame*, **160**:1583–1593, 2013. 21

[44] Eric W. Weisstein. **Fundamental Theorems of Calculus**, 1999. http://mathworld.wolfram.com/FundamentalTheoremsofCalculus.html. 22

[45] Genyuan Li, Sheng-Wei Wang, and Herschel Rabitz. **High Dimensional Model Representation (HDMR): concepts and applications**. *Journal of Physical Chemistry*, **117**:11–20, 1999. 24

[46] Ajit Desai and Sunetra Sarkar. **Analysis of a Nonlinear Aeroelastic System with Parametric Uncertainties Using Polynomial Chaos Expansion**. *Mathematical Problems in Engineering*, 2010. 32

[47] Raymond H. Myers, Douglas C. Montgomery, and Christine M. Anderson-Cook. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. Wiley, 3 edition, 2009. 32, 72

[48] Tonkid Chantrasmi. **Pade-Legendre Method for Uncertainty Quantification with Fluid Dynamics Applications**, June 2011. 32, 161

[49] Felipe A. C. Viana, Christian Gogu, and Raphael T. Haftka. **Making the Most Out of Surrogate Models: Tricks of the Trade**. In *ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference IDETC/CIE*, August 15-18 2010. 32, 124, 132

[50] Alexander I. J. Forrester and Andy J. Keane. **Recent advances in surrogate-based optimization**. *Progress in Aerospace Sciences*, **45**:50–79, 2009. 32, 124

[51] Nestor V. Queipo, Raphael T. Haftka, Wei Shyya, Tushar Goel, Rajkumar Vaidyanathan, and P. Kevin Tucker. **Surrogate-based analysis and optimization**. *Progress in Aerospace Sciences*, **41**, 2005. 32, 33

[52] Alexander I. J. Forrester, Andras Sobester, and Andy J. Keane. *Engineering Design via Surrogate Modelling: A Practical Guide*. A John Wiley and Sons, Ltd., Publication, 2008. 32, 124

[53] Xiang Ma and Nicholas Zabaras. **An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations**. *Journal of Computational Physics*, **228**(8):3084–3113, 2009. 33

[54] Xiang Ma and Nicholas Zabaras. **An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations**. *Journal of Computational Physics*, **229**:3884–3915, 2010. 33

[55] Gael Poette and Didier Lucor. **Non intrusive iterative stochastic spectral representation with application to compressible gas dynamics**. *Journal of Computational Physics*, **231**:3587–3609, 2012. 33

[56] Geraud Blatman and Bruno Sudret. **Adaptive sparse polynomial chaos expansion based on least angle regression**. *Journal of Computational Physics*, pages 2345–2367, 2011. 33, 35, 92

[57] Ruichen Jin, Wei Chen, and Timothy W. Simpson. **Comparative Studies Of Metamodeling Techniques Under Multiple Modeling Criteria**. *Structural and Multidisciplinary Optimization*, **23**:1–13, 2000. 33, 124, 132

[58] M. Isabel and Reis Santos. **SEQUENTIAL DESIGNS FOR SIMULATION EXPERIMENTS: NONLINEAR REGRESSION METAMODELING**. In *MIC '07 Proceedings of the 26th IASTED International Conference on Modelling, Identification, and Control*, pages 88–93, 2007. 33

[59] T. J. Mackman and C. B. Allen. **Investigation of an adaptive sampling method for data interpolation using radial basis functions**. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING*, **83**:915–938, 2010. 33, 34

[60] Liang Zhao1, K.K. Choi, and Ikjin Lee. **A Metamodeling Method Using Dynamic Kriging and Sequential Sampling**. In *13th AIAA/ISSMO Multidisciplinary Analysis Optimization Conference*, 2010. 33, 34

[61] Prasanna Balaprakash, Robert B. Gramacy, and Stefan M. Wild. **Active-Learning-Based Surrogate Models for Empirical Performance Tuning**. In *IEEE Cluster 2013*, 2013. 33

[62] Karel Crombecq, Luciano De Tommasi, Dirk Gorissen, and Tom Dhaene. **A novel sequential design strategy for global surrogate Modeling**. In *Winter Simulation Conference*, 2009. 33

[63] Julien Villemonteix, Emmanuel Vazquez, and Eric Walter. **An informational approach to the global optimization of expensive-to-evaluate functions**. *Journal of Global Optimization*, 2008. 33

[64] L. Le Gratiet and C. Cannamela. **Kriging-based sequential design strategies using fast cross-validation techniques with extensions to multi-fidelity computer codes**. *ArXiv e-prints*, October 2012. 33

[65] C. W. Clenshaw and A. R. Curtis. **A method for numerical integration on an automatic computer**. *Numerische Mathematik*, **2**:197–205, 1960. 33

[66] Xiang Ma and Nicholas Zabaras. **An adaptive hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations**. *Journal of Computational Physics*, **228**:3084–3113, 2009. 34

[67] Luc Pronzato and Werner Muller. **Design of computer experiments: space filling and beyond**. *Statistics and Computing*, **22**(3), 2012. 34

[68] Bengt Fornberg and Julia Zuev. **The Runge phenomenon and spatially variable shape parameters in RBF interpolation**. *Computers and Mathematics with Applications*, **54**(3), August 2007. 36

[69] Zhen Gao and Jan S. Hesthaven. **On ANOVA expansions and strategies for choosing the anchor point**. *Applied Mathematics and Computation*, **217**(7), August 2010. 55

[70] G. A. Bird. *Bird, Molecular Gas Dynamics and the Direct Simulation of Gas Flows*. Oxford Science Publication, New York, 1994. 59

[71] K. Pearson. **On Lines and Planes of Closest Fit to Systems of Points in Space**. *Philosophical Magazine*, **2**(11):559–572, 1901. 59

[72] Michael E. Tipping and Christopher M. Bishop. **Probabilistic Principal Component Analysis**. *Journal of the Royal Statistical Society*, **61**(3):611–622, 1999. 59

[73] Robert Tibshirani. **Principal curves revisited**. *Statistics and Computation*, **2**(4):183–190, 1992. 59

[74] G. Kerschen, J. Golinval, A. F. Vakakis, and L. A. Bergman. **The Method of Proper Orthogonal Decomposition for Dynamical Characterization and Order Reduction of Mechanical Systems: An Overview**. *Nonlinear Dynamics*, **41**:147–169, 2005. 59

[75] Mara C. Alonso, Jos A. Malpica, and Alex Martnez De Agirre. **Consequences of the Hughes Phenomenon on Some Classification Techniques**. In *ASPRS 2011 Annual Conference Milwaukee*, Wisconsin, 2011. 60

[76] Charu C. Aggarwal, Alexander Hinneburg, and Daniel A. Keim. **On the surprising behavior of distance metrics in high dimensional space**. In *The 8th International Conference on Database Theory*, January 4-6 2001. 60

[77] Dominik Schnitzer and Arthur Flexer. **Choosing the Metric in High-Dimensional Spaces Based on Hub Analysis**. In *ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Bruges, Belgium, 23-25 April 2014. 60

[78] Milos Radovanovic, Alexandros Nanopoulos, and Mirjana Ivanovic. **Hubs in Space: Popular Nearest Neighbors in High-Dimensional Data**. *Journal of Machine Learning Research*, **11**:2487–2531, 2010. 60

[79] Arthur Zimek, Erich Schubert, and Hans-Peter Kriegel. **A Survey on Unsupervised Outlier Detection in High-Dimensional Numerical Data**. *Journal Statistical Analysis and Data Mining*, **5**(5):363–387, October 2012. 60

[80] Sameer A. Nene and Shree K. Nayar. **A Simple Algorithm for Nearest Neighbor Search in High Dimensions**. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(9), September 1997. 60

[81] Oliver Kramer. **Unsupervised K-Nearest Neighbor Regression**. In *IEEE 10th International Conference on Machine Learning and Applications*, Honolulu, Hawaii, 2011. 60

[82] P.M. Mehta, M. Kubicek, E. Minisci, and M. Vasile. **Sensitivity Analysis and Probabilistic Re-entry Modeling for Debris using High Dimensional Model Representation based Uncertainty Treatment**. *Advances in Space Research*, 2016. 66

[83] Burgess Davis and David R. McDonald. **An elementary proof of the local central limit theorem**. *Journal of Theoretical Probability*, **8**(3):693–701, July 1995. 69

[84] B. M. Adams, L. E. Bauman, W. J. Bohnhoff, K. R. Dalbey, M. S. Ebeida, J. P. Eddy, M. S. Eldred, P. D. Hough, K. T. Hu, J. D. Jakeman, J. A. Stephens, L. P. Swiler, D. M. Vigil, and T. M. Wildey. **Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.3 Users Manual**. Technical report, Sandia National Laboratories, November 2015. 71, 185

[85] Charles Teddlie and Fen Yu. **Mixed Methods Sampling: A Typology With Examples**. *Journal of Mixed Methods Research*, **1**(1):77–100, 2007. 97

[86] V. Roshan Joseph, Ying Hung, and Agus Sudjianto. **Blind kriging: A new method for developing metamodels**. *Journal of mechanical design*, **130**(3), 2008. 115

[87] Wang XiaoDong and Kang Shun. **Application of polynomial chaos on numerical simulation of stochastic cavity flow**. *Science China Technological Sciences*, **53**(10), October 2010. 124

[88] Dongbin Xiu. *Numerical Methods for Stochastic Computations*. Princeton University Press, 2010. 124

[89] Prasad Cheema and Gareth Vio. **A Non-Intrusive Polynomial Chaos Method to Efficiently Quantify Uncertainty in an Aircraft T-Tail**. *Applied Mechanics and Materials*, **846**:512–517, 2016. 124

[90] R. Abgrall, P.M. Congedo, C. Corre, and S. Galera. **A Simple Semi-Intrusive Method For Uncertainty Quantification of Shocked Flows, Comparison with a Non-Intrusive Polynomial Chaos method**. In *V European Conference on Computational Fluid Dynamics*, Lisbon, Portugal, 14-17 June 2010. 124

[91] Dinesh Kumar and C. Lacor. **Heat conduction in a 2D domain with geometrical uncertainty using intrusive polynomial chaos method**. In *9th National Congress on Theoretical and Applied Mechanics*, Brussels, 9-11 May 2012. 124

[92] Emil Brandt Kærgaard. **Spectral Methods for Uncertainty Quantification**, 2013. 124

[93] Mariano Gasca and Thomas Sauer. **Polynomial interpolation in several variables**. *Advances in Computational Mathematics*, **12**:377–410, 2000. 124

[94] Nestor V. Queipoa, Raphael T. Haftkaa, Wei Shyya, Tushar Goela, Rajkumar Vaidyanathana, and Kevin Tucker. **Surrogate-based analysis and optimization**. *Progress in Aerospace Sciences*, **41**:1–28, 2005. 124

[95] Wallace G. Ferreira and Alberto L. Serpa. **Ensemble of metamodels: the augmented least squares approach**. *Structural Multidisciplinary Optimization*, **53**:1019–1046, 2016. 124, 125, 132

[96] G. Gary Wang and S. Shan. **Review of Metamodeling Techniques in Support of Engineering Design Optimization**. *Journal of Mechanical design*, 2006. 124

[97] Victoria C.P. Chen, Kwok-Leung Tsui, Russell R. Barton, and Martin Meckesheimer. **A review on design, modeling and applications of computer experiments**. *IIE Transactions*, **38**(4):273–291, February 2006. 124

[98] Dirk Gorissen, Tom Dhaene, and Filip De Turck. **Evolutionary Model Type Selection for Global Surrogate Modeling**. *Journal of Machine Learning Research*, **10**:2039–2078, 2009. 124

[99] Songqing Shan and G. Gary Wang. **Survey of modeling and optimization strategies to solve high-dimensional design problems with computationally-expensive black-box functions**. *Struct Multidisc Optim*, **41**:219–241, 2010. 124

[100] Jie Zhang, Souma Chowdhury, Junqiang Zhang, Achille Messac, and Luciano Castillo. **Adaptive Hybrid Surrogate Modeling for Complex Systems**. *American Institute of Aeronautics and Astronautics*, 2013. 124

[101] Jie Zhang, Souma Chowdhury, and Achille Messac. **An Adaptive Hybrid Surrogate Model**. *Structural and Multidisciplinary Optimization*, **46**(2):223–238, 2012. 124

[102] Luis E. Zerpaa, Nestor V. Queipoa, Salvador Pintosa, and Jean-Louis Salagerb. **An optimization methodology of alkalinesurfactantpolymer flooding processes using field scale numerical simulation and multiple surrogates**. *Journal of Petroleum Science and Engineering*, **47**:197–208, 2005. 124

[103] Christopher M. Boshop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995. 124, 153

[104] Felipe A. C. Viana. *Multiple Surrogates for Prediction and Optimization.* PhD thesis, University of Florida, 2011. 125

[105] Juliane Muller and Robert Piche. **Mixture surrogate models based on Dempster-Shafer theory for global optimization problems**. *Journal of Global Optimization*, **51**:79–104, 2011. 125

[106] Felipe A. C. Viana, Raphael T. Haftka, and Layne T. Watson. **Why Not Run the Efficient Global Optimization Algorithm with Multiple Surrogates?** In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2010. 125

[107] Zongzhao Zhou, Yew Soon Ong, Meng Hiot Lim, and Bu Sung Lee. **Memetic Algorithm using Multi-Surrogates for Computationally Expensive Optimization Problems**. *Soft Computing*, **11**(10):957–971, August 2007. 125

[108] Haitao Liu and Shengli Xu. **Sampling Strategies and Metamodeling Techniques for Engineering Design: Comparison and Application**. In *ASME Turbo Expo: Turbine Technical Conference and Exposition*, 13-17 June 2016. 132

[109] Felipe A. C. Viana. *Multiple Surrogates for Prediction and Optimization*. PhD thesis, University of Florida, 2011. 132

[110] Stephen M. Stigler. **Regression towards the mean, historically considered**. *Statistical Methods in Medical Research*, **6**:103–114, 1997. 139

[111] Frederick P. Brooks. *The Mythical Man-Month*. Addison-Wesley, 1995 [1975]. 154

[112] O.P. Le Maitre and O.M. Knio. *Spectral Methods for Uncertainty Quantification With Applications to Computational Fluid Dynamics*. Springer, 2010. 155, 160

[113] Xiaoliang Wan and George Em Karniadakis. **Long-term behavior of polynomial chaos in stochastic flow simulations**. *Computer Methods in Applied Mechanics and Engineering*, **195**:5582–5596, 2006. 155

[114] Mikel Landajuela. **Burgers Equation**. Internship - summer, Basque Center for Applied Mathematics, 2011. 159

[115] Cleve Moler. *Numerical Computing with MATLAB*. SIAM, 2008. 160, 166

[116] *An Introduction to Splines for Use in Computer Graphics and Geometric Modelling*. CA: Morgan Kaufmann, 1998. 160, 166

[117] Doron Levy. *Introduction to Numerical Analysis*, 2010. 160

[118] Desmond J. Higham. **Monotonic piecewise cubic interpolation with applications to ODE plotting**. *Journal of Computational and Applied Mathematics*, **39**:287–294, 1992. 160

[119] Hiroshi Akimo. **A New Method of Interpolation and Smooth Curve Fitting Based on Local Procedures**. *Journal of the ACM (JACM)*, **17**:589–602, October 1970. 160

[120] Vasily Strela. *Multiwavelets: Theory and Applications*. PhD thesis, Massachusetts Institute of Technology, June 1996. 160

[121] O. P. Le Maitre, H. N. Najm, R. G. Ghanem, and O.M. Knio. **Multiresolution Analysis of Wiener-type Uncertaitny Propagation Schemes**. *Journal of Computational Physics*, January 2004. 160

[122] S. L. Lee, A. Sharma, and H. H. Tan. **Spline Interpolation and Wavelet Construction**. *Applied and Computational Harmonic Analysis*, **5**:249–276, 1998. 160

[123] D. Schiavazzi, A. Doostan, and G. Iaccarino. **Sparse multiresolution stochastic approximation for uncertainty quantification**. *Journal of Advanced Computer Science and Applications*, 2013. 160

[124] Xiaoliang Wan and George Em Karniadakis. **Multi-Element Generalized Polynomial Chaos for Arbitrary Probability Measures**. *Journal on Scientific Computing*, **28**(3):901–928, January 2006. 160

[125] Jing Li and Panos Stinis. **Mesh refinement for uncertainty quantification through model reduction**. *Journal of Computational Physics*, **280**:164–183, 2015. 161

[126] Xiaoliang Wan and George Em Karniadakis. **An Adaptive Multi-Element Generalized Polynomial Chaos Method for Stochastic Differential Equations**. *Journal of Computational Physics*, **209**:617–642, November 2005. 161

[127] Xiaoliang Wan and George Em Karniadakis. **Error Control in Multi-Element Generalized Polynomial Chaos Method for Elliptic Problems with Random Coefficients**. *Communications in Computational Physics*, **5**(2-4):793–820, . 161

[128] Gaurav Kewlani and Karl Iagnemma. **A Multi-Element Generalized Polynomial Chaos Approach to Analysis of Mobile Robot Dynamics under Uncertainty**. In *The IEEE/RSJ International Conference on Intelligent Robots and Systems*. St. Louis, USA, October 2009. 161

[129] Jeroen A.S. Witteveen and Gianluca Iaccarino. **Simplex stochastic collocation with ENO-type stencil selection for robust uncertainty quantification**. *Journal of Computational Physics*, **239**:1–21, 2013. 161

[130] D. Lucor, J. Witteveen, P. Constantine, D. Schiavazzi, and G. Iaccarino. **Comparison of adaptive uncertainty quantification approaches for shock wave-dominated flows**. Proceedings of the 2012 ctr summer program, Stanford University, 2012. 161

[131] J.S. Hesthaven, S.M. Kaber, and L. Lurati. **Pade-Legendre Interpolants for Gibbs Reconstruction**. *Journal of Scientific Computing*, **28**(2):337–359, September 2006. 161

[132] A. L. Tampos, J. E. C. Lope, and Jan S. Hesthaven. **Accurate Reconstruction of Discontinuous Functions Using the Singular Pad-Chebyshev Method**. *IAENG International Journal of Applied Mathematics*, **42**(4):242–249, 2012. 161

[133] Eric W. Weisstein. **Vandermonde Matrix**, 2016. 164

[134] Eric W. Weisstein. **Moore-Penrose Matrix Inverse**, 2016. 165

[135] Eric W. Weisstein. **Cubic Spline**, 2016. 166

[136] M.C. Kennedy and A. O'Hagan. **Predicting the output from a Complex computer code when fast approximations are available**. *Biometrika*, **87**(1):1–13, 2000. 183

[137] Yuichi Kuya, Kenji Takeda, Xin Zhang, and Alexander I. J. Forrester. **Multifidelity Surrogate Modeling of Experimental and Computational Aerodynamic Data Sets**. *AIAA Journal*, **49**(2), February 2011. 183

[138] Seok-Ho Son, Do-Hyun Park, Kyeong-Jun Cha, and Dong-Hoon Choi. **Constrained Global Design Optimization Using a Multi-fidelity Model**. In *10th World Congress on Structural and Multidisciplinary Optimization*, Orlando, Florida, USA, 2013. 183

[139] Zhiguang Qian, Carolyn Conner Seepersad, V. Roshan Joseph, Janet K. Allen, and C. F. Jeff Wu. **Building Surrogate Models Based on Detailed and Approximate Simulations**. *Journal of Mechanical Design*, **128**(4):668–677, 2005. 183

[140] Alexander I.J Forrester, Andras Sobester, and Andy J Keane. **Multi-fidelity optimization via surrogate modelling**. In *Proc. R. Soc. A*, pages 3251–3269, December 2007. 183

[141] Shifeng Xiong, Peter Z. G. Qian, and C. F. Jeff Wu. **Sequential design and analysis of high-accuracy and low-accuracy computer codes**. *Technometrics*, **55**(1):37–46, 2013. 183, 212

[142] Daeyeon Lee, Nhu Van Nguyen, Maxim Tyan, Hyung-Geun Chun, Sangho Kim, and Jae-Woo Lee. **Enhanced multi-fidelity model for flight simulation using global exploration and the Kriging method**. *Journal of Aerospace Engineering*, 2016. 183

[143] Jiandao Zhu, Yi-Jen Wang, and Matthew Collette. **A multi-objective variable-fidelity optimization method for genetic algorithms**. *Engineering Optimization*, **46**(4):521–542, 2014. 184

[144] T. D. Robinson, K. E. Willcox, M. S. Eldred, and R. Haimesk. **Multifidelity Optimization for Variable-Complexity Design**. In *11th AIAA/ISSMO multidisciplinary analysis and optimization conference*, 2006. 184

[145] Andrew I. March. *Multifidelity Methods for Multidisciplinary System Design.* PhD thesis, Massachusetts Institute of Technology, 2012. 184

[146] Leo W. T. Ng and Karen E. Willcox. **Multifidelity approaches for optimization under uncertainty**. *International Journal for Numerical Methods in Engineering*, 2014. 184

[147] Leo Way-Tsun Ng. *Multifidelity Approaches for Design Under Uncertainty.* PhD thesis, Massachusetts Institute of Technology, 2013. 184

[148] John W. Bandler, Radoslaw M. Biernacki, Shao Hua Chen, Piotr A. Grobelny, and Ronald H. Hemmers. **Space Mapping Technique for Electromagnetic Optimization**. *IEEE Transactions on Microwave Theory and Techniques*, **42**(12), December 1994. 184

[149] John W. Bandler, Radoslaw M. Biernacki, Shao Hua Chen, Ronald H. Hemmers, and Piotr A. Grobelny. **Electromagnetic Optimization Exploiting Agressive Space Mapping**. *IEEE Transactions on Microwave Theory and Techniques*, **43**(12):2874–2882, January 1996. 184

[150] Mohamed H. Bakr, John W. Bandler, Natalia Georgieva, and Kaj Madsen. **A Hybrid Aggressive Space-Mapping Algorithm for EM Optimization**. *IEEE Transactions on Microwave Theory and Techniques*, **47**(12), December 1999. 184

[151] S. Vivier, D. Lemoine, and Guy Friedrich. **Fast optimization of an IPMSM with Space Mapping Technique**. In *2011 IEEE Energy Conversion Congress and Exposition (ECCE)*, 17-22 September 2011. 184

[152] Mohamed H. Bakr, John W. Bandler, Mostafa A. Ismail, Jose Ernesto Rayas-Sanchez, and Qi-Jun Zhang. **Neural Space-Mapping Optimization for EM-Based Design**. *IEEE Transactions on Microwave Theory and Techniques*, **48**(12), December 2000. 184

[153] John W. Bandler, Radoslaw M. Biernacki, Shao Hua Chen, and Dzevat Omeragic. **Space Mapping Optimization of Waveguide Filters using Finite Element and Mode-matching Electromagnetic simulators**. *International Journal of RF and Microwave Computer-Aided Engineering*, **9**(1):54–70, January 1999. 184

[154] Slawomir Koziel and John W. Bandler. **Space-Mapping Optimization With Adaptive Surrogate Model**. *IEEE Transactions on Microwave Theory and Techniques*, **55**(3), 2007. 184

[155] Mohamed H. Bakr, John W. Bandler, Kaj Madsen, and Jacob Sondergaard. **An Introduction to the Space Mapping Technique**. *Optimization and Engineering*, **2**(4):369–384, December 2001. 184

[156] Theresa D. Robinson. *Surrogate-Based Optimization using Multi Fidelity models with Variable Parameterization.* PhD thesis, Messachusetts Institute of Technology, 2007. 184

[157] John W. Bandler, Qingsha S. Cheng, Sameh A. Dakroury, Ahmed S. Mohamed, Mohamed H. Bakr, Kaj Madsen, and Jacob Sondergaard. **Space Mapping: The State of the Art**. *IEEE Transactions on Microwave Theory and Techniques*, **52**(1), January 2004. 184

[158] M. S. Eldred and D. M. Dunlavy. **Formulations for Surrogate-Based Optimization with Data Fit, Multifidelity, and Reduced-Order Models**. In *11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference.* Portsmouth, Virginia, 6-8 September 2006. 185

[159] A. Santiago Padron, Juan J. Alonso, Francisco Palacios, Matthew Barone, and Michael S. Eldred. **Multi-Fidelity Uncertainty Quantification: Application to a Vertical Axis Wind Turbine Under an Extreme Gust**. In *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Atlanta, GA, 16-20 June 2014. 185

[160] Natalia M. Alexandrov, Robert Michael Lewis, Clyde R. Gumbert, Lawrence L. Green, and Perry A. Newman. **Approximation and Model Management in Aerodynamic Optimization with Variable-Fidelity Models**. *Journal of Aircraft*, **38**(6), November 2001. 185

[161] Piyush Mehta, Edmondo Minisci, Massimiliano Vasile, Andrew C. Walker, and Melrose Brown. **An Open Source Hypersonic Aerodynamic and Aerothermo-dynamic Modeling Tool**. In *8th European Symposium on Aerothermodynamics of Space Vehicles*, Lisbon, Portugal, March 2015. 222, 224

[162] Piyush M. Mehta, Andrew C. Walker, Melrose Brown, Edmondo Minisci, and Massimiliano Vasile. **Sensitivity Analysis towards Probabilistic Re-Entry Modeling of Spacecraft and Space Debris**. In *AIAA Modeling and Simulation Technologies Conference*, Dallas, June 2015. 222

[163] Eric W. Weisstein. **Runge-Kutta Method**, 2016. http://mathworld.wolfram.com/Runge-KuttaMethod.html. 223

[164] S. A. Schaaf and P. L. Chambre. *High Speed Aerodynamics and Jet Propulsion, ch. Flow of Rarefied Gases.* Princeton Univ. Press, Princeton, NJ, 1958. 223, 224

[165] U.S. Government Printing Office. **U.S. Standard Atmosphere**, 1976. 225

[166] T. J. Scanlon, E. Roohi, C. White, M. Darbandi, and J. M. Reese. **An open source, parallel DSMC code for rarefied gas flows in arbitrary geometries**. *Computers and Fluids*, **39**(10):2078–2089, 2010. 255

[167] A. C. Aikin, J. M. Picone, A. E. Hedin, and D. P. Drob. **NRLMSISE-00 Empirical Model of the Atmosphere: Statistical Comparisons and Scientific Issues**, 2001. 257

APPENDIX A

Let us assume a function, $f(x_1, x_2)$, with two random variables. The variance is computed in the following way

$$\sigma^2 = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left(\left(\int_{c_{x_1}}^{x_1} \frac{\partial f(\xi)}{\partial \xi_1} d\xi_1 + \int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_2} d\xi_2 + \int_{c_{x_1}}^{x_1}\int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_1, \xi_2} d\xi_1 d\xi_2\right) - \mu\right)^2 p(x_1, x_2) dx_1 dx_2 \tag{11.1}$$

Now, let us closely look on the inner part of Eq. (11.1). The inner part can be expanded in the following way

$$\begin{aligned}
&\left(\left(\int_{c_{x_1}}^{x_1} \frac{\partial f(\xi)}{\partial \xi_1} d\xi_1 + \int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_2} d\xi_2 + \int_{c_{x_1}}^{x_1}\int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_1, \xi_2} d\xi_1 d\xi_2\right) - \mu\right)^2 \\
&= \\
&\left(\int_{c_{x_1}}^{x_1} \frac{\partial f(\xi)}{\partial \xi_1} d\xi_1\right)^2 + 2\int_{c_{x_1}}^{x_1} \frac{\partial f(\xi)}{\partial \xi_1} d\xi_1 \int_{c_{x_1}}^{x_1}\int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_1, \xi_2} d\xi_1 d\xi_2 \\
&+ \left(\int_{c_{x_1}}^{x_1}\int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_1, \xi_2} d\xi_1 d\xi_2\right)^2 + 2\int_{c_{x_1}}^{x_1} \frac{\partial f(\xi)}{\partial \xi_1} d\xi_1 \int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_2} d\xi_2 \\
&+2\int_{c_{x_1}}^{x_1}\int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_1, \xi_2} d\xi_1 d\xi_2 \int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_2} d\xi_2 + \left(\int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_2} d\xi_2\right)^2 \\
&-2\int_{c_{x_1}}^{x_1} \frac{\partial f(\xi)}{\partial \xi_1} d\xi_1 \mu - 2\int_{c_{x_1}}^{x_1}\int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_1, \xi_2} d\xi_1 d\xi_2 \mu \\
&-2\int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_2} d\xi_2 \mu + \mu^2
\end{aligned} \tag{11.2}$$

If the same approach as in Sec. 2.2 is followed, the inner part of Eq. (11.1) is expanded in the following way

$$\begin{aligned}
&\left(\int_{c_{x_1}}^{x_1} \frac{\partial f(\xi)}{\partial \xi_1} d\xi_1 - \mu_1\right)^2 + \left(\int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_2} d\xi_2 - \mu_2\right)^2 + \\
&\left(\int_{c_{x_1}}^{x_1}\int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_1, \xi_2} d\xi_1 d\xi_2) - \mu_{12}\right)^2 = \\
&\left(\int_{c_{x_1}}^{x_1} \frac{\partial f(\xi)}{\partial \xi_1} d\xi_1\right)^2 + \left(\int_{c_{x_1}}^{x_1}\int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_1, \xi_2} d\xi_1 d\xi_2\right)^2 + \left(\int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_2} d\xi_2\right)^2 \\
&-2\int_{c_{x_1}}^{x_1} \frac{\partial f(\xi)}{\partial \xi_1} d\xi_1 \mu_1 - 2\int_{c_{x_1}}^{x_1}\int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_1, \xi_2} d\xi_1 d\xi_2 \mu_{12} - 2\int_{c_{x_2}}^{x_2} \frac{\partial f(\xi)}{\partial \xi_2} d\xi_2 \mu_2 \\
&+\mu_1^2 + \mu_2^2 + \mu_{12}^2
\end{aligned} \tag{11.3}$$

Eq. (11.2) clearly differs from Eq. (11.3). Therefore, a sum of the partial variances is not the total variance as it is mentioned in Sec. 2.2.

APPENDIX B

To prove the statement given in Sec. 2.2, let us assume the following equations:

$$F_1(x_1, x_2, x_3, x_4) = 5000 + \frac{x_1^3 - x_2^3 - e^{x_3^2 x_1}}{x_4}$$
$$F_2(x_1, x_2, x_3, x_4) = \frac{x_1^3 - x_2^3 - e^{x_3^2 x_1}}{x_4}$$

where x_i represents a random variable with a uniform distribution and boundaries equal to $[0, 1]$. These two functions are different, however, their derivative (along all directions) are the same. Therefore, according to a given statement, their histograms should have the same shape. Let us now create a histogram using MC simulation with 100000 sample.

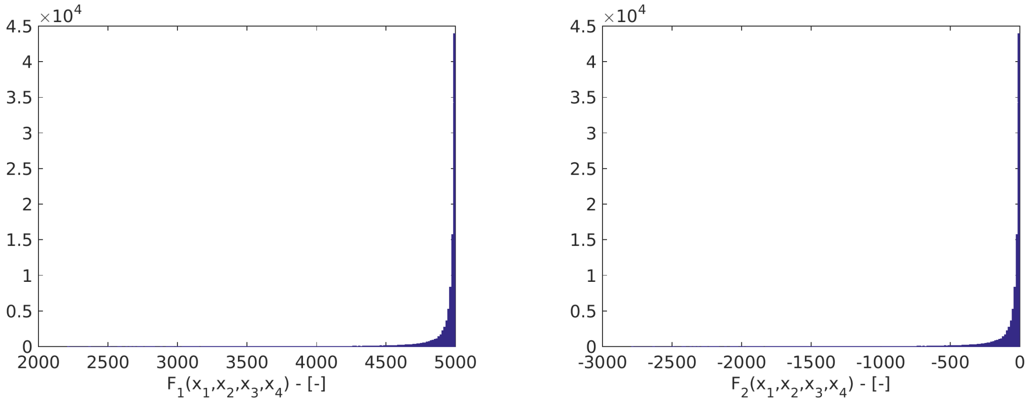


Figure 11.1: Histogram for function $F_1(x_1, x_2, x_3, x_4)$ and function $F_2(x_1, x_2, x_3, x_4)$

From given figures, it is obvious that the shape of both histograms is the same. Therefore, one can conclude that the shape is given only by a derivative of the function of interest.

APPENDIX C

One of the main advantages of the proposed approach are the increment functions, which can be plotted and one can understand the behaviour of the problem. However, this approach is possible only in a case of the lower stochastic domains and in the higher domains, one need to use the partial histogram to understand the shape of given increment function. Knowing the input histogram and the output partial histogram, the increment function can be reconstructed. The following approach helps to practice the understanding of the higher order partial histograms.

Let us assume the following function:

$$F(x) = \frac{1}{x^2} \tag{11.4}$$

where x represents the random variable with a uniform distribution and defined on interval [0.5, 1.5]. Let us assume an input histogram, which is constructed with 6 bins and the final histogram, which is constructed with 6 bins. To simplify the problem, let us divide the function F(x) (Eq. (11.4)) into 6 segments and replace the function in each segment with a linear approximation. This is showed in Fig. 11.2.

First, it is necessary to realize that the surface of the input histogram and the output histogram is the same. The same applies for each bin of the output histogram (assuming that the input bin is constructed with the same number of samples as the output bin). The linear model basically scales the histogram and one can see that each bin in the input histogram is scaled on the output. Knowing the input histogram and the output histogram, one can, with a little bit of training, reconstruct the increment function using a linear approach.

For the higher stochastic domains, one needs to realize that all the lower stochastic domains are zero and all the work is done in the corners of an increment function. Therefore, if a partial histogram of a higher order increment function is elongated in one direction, it means that one of the corners of the increment function has a steep ascent. The same linear approach can be used for the higher order increment functions.

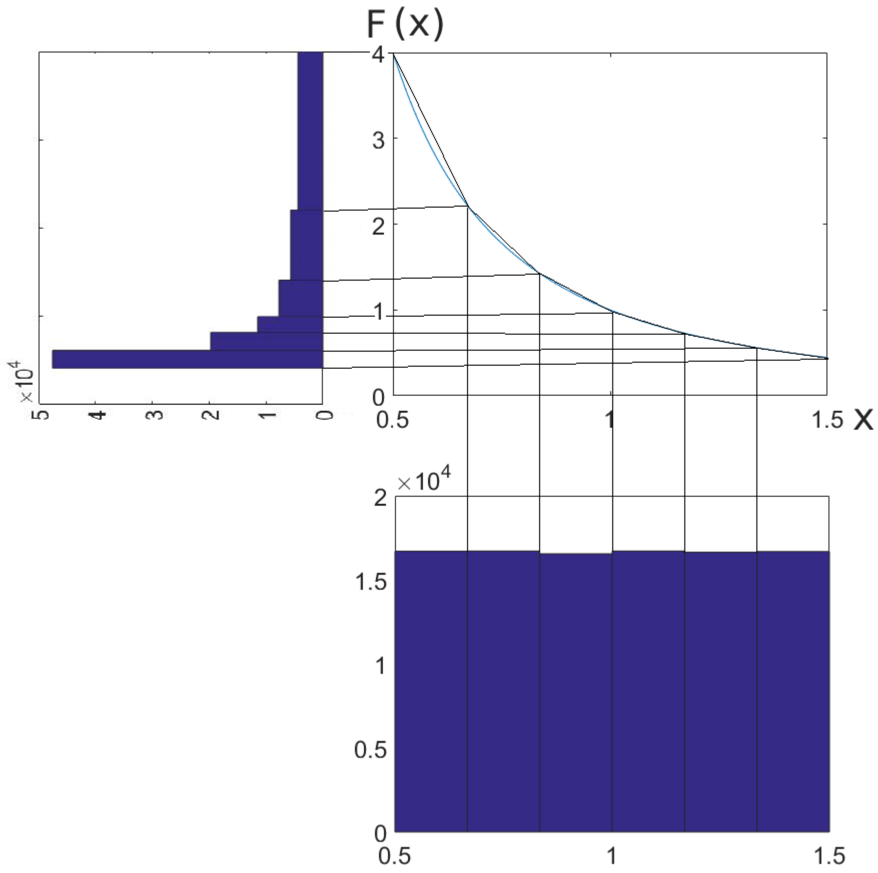


Figure 11.2: Reconstruction approach of an increment function using input and output histogram

APPENDIX D

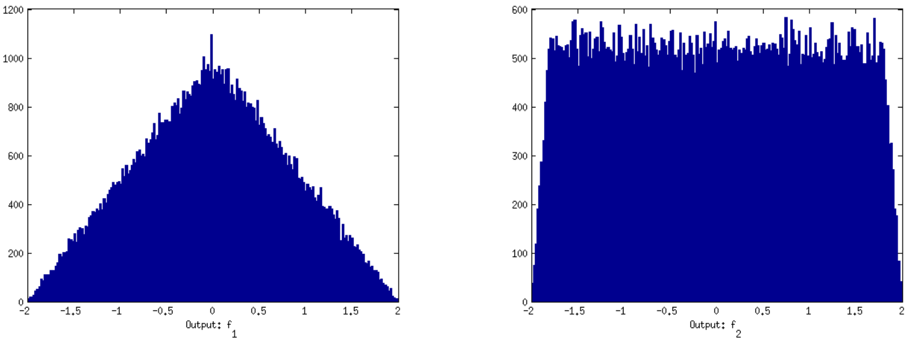
To prove the statements from Sec. 4.3, we consider the following functions:

$$f_1(x_1, x_2) = x_1 + x_2$$
$$f_1(x_1, x_2) = 1.9x_1 + 0.1x_2$$

Random variables x_1 and x_2 have the following distributions:

Variable	Min	Max
x_1	0	1
x_2	0	1

The histograms are showed in Fig. (11.3), where the histogram for function $f_1(x_1, x_2)$ is closer to the Gaussian distribution than the histogram for function $f_2(x_1, x_2)$.



(a) **(b)**

Figure 11.3: Visualization of histogram for function $f_1(x_1, x_2)$ - (a) and function $f_2(x_1, x_2)$ - (b)

APPENDIX E

Resemblance between Polynomial models and High Dimensional Model Representation

To show the resemblance between these techniques, let us first assume the following 2-D polynomial function:

$$f(x_1, x_2) = c_0 + c_1 x_1 + c_2 x_1^2 + c_3 x_2 + c_4 x_2^2 + c_5 x_1 x_2 \tag{11.5}$$

where c_i represents an unknown coefficients yet to be established. To establish mentioned coefficients, one can use various techniques such as least square estimate. However, it should be noted that these coefficients are data driven and not model driven.

The cut-HDMR for the function with two random variables reads:

$$f(x_1, x_2) = f({}^c x_1, {}^c x_2) + dF_1(x_1) + dF_2(x_2) + dF_{12}(x_1, x_2) \tag{11.6}$$

where ${}^c x_i$ represents the central point. Let us now assume a quadratic polynomial model for each increment function. Each increment function in Eq. (11.6) is represented with a polynomial expansion in the following shape:

$$\begin{aligned} &dF_1(x_1) = c_0^1 + c_1^1 x_1 + c_2^1 x_1^2 \\ &dF_2(x_2) = c_0^2 + c_1^2 x_2 + c_2^2 x_2^2 \\ &dF_{12}(x_1, x_2) = c_0^{12} + c_1^{12} x_1 + c_2^{12} x_1^2 + c_3^{12} x_2 + c_4^{12} x_2^2 + c_5^{12} x_1 x_2 \end{aligned} \tag{11.7}$$

where c_i^k represents an unknown coefficient yet to be established. Next step represents substitution of the increment functions into Eq. (11.6). The final equation reads

$$\begin{aligned} f(x_1, x_2) = (f({}^c x_1, {}^c x_2) + c_0^1 + c_0^2 + c_0^{12}) + (c_1^1 + c_1^{12})x_1 \\ +(c_2^1 + c_2^{12})x_1^2 + (c_1^2 + c_3^{12})x_2 \\ +(c_2^2 + c_4^{12})x_2^2 + c_5^{12} x_1 x_2 \end{aligned} \tag{11.8}$$

one can quickly notice, that the structure of Eq. (11.8) is same as in Eq. (11.5).

Therefore, one can re-write coefficients c_i as follows

$$\begin{aligned}
c_0 &= f({}^c x_1, {}^c x_2) + c_0^1 + c_0^2 + c_0^{12} \\
c_1 &= c_1^1 + c_1^{12} \\
c_2 &= c_2^1 + c_2^{12} \\
c_3 &= c_1^2 + c_3^{12} \\
c_4 &= c_2^2 + c_4^{12} \\
c_5 &= c_5^{12}
\end{aligned} \tag{11.9}$$

One can conclude that using the polynomial models for each increment function is same as using polynomial model for the whole problem.

APPENDIX F

Increment functions for the Borehole problem

The collection of all increment functions, which are interpolated in the Borehole problem is given in Tab. 11.1 for accuracy 0.01 and in Tab. 11.2 for accuracy 0.001

Increment function	Partial Mean	Partial Variance	Mean Sensitivity	Variance Sensitivity
1	1.8167	531.4599	0.6368	0.6719
2	0.0003	1.6899e-05	0.0001	2.1367e-08
3	-4.2337e-06	5.9210e-10	1.4841e-06	7.4865e-13
4	-0.0276	72.0510	0.0096	0.0911
5	-0.0042	0.0005	0.0014	7.5009e-07
6	-0.0258	72.1023	0.0090	0.0911
7	0.9307	71.2253	0.3262	0.0900
8	0.0030	16.7087	0.0010	0.0211
7.8	-0.0011	0.2449	0.0003	0.0003
6.8	-0.0012	0.2390	0.0004	0.0003
6.7	-0.0013	1.0657	0.0004	0.0013
4.8	0.0002	0.2379	9.5616e-05	0.0003
4.7	-0.0006	1.0646	0.0002	0.0013
1.8	0.0045	1.7586	0.0015	0.0022
1.7	0.0153	7.5383	0.0053	0.0095
1.6	0.0068	7.5906	0.0024	0.0095
1.4	0.0128	7.5988	0.0045	0.0096

Table 11.1: Results of the increment functions for the Borehole problem using residual: 0.01

Increment function	Partial Mean	Partial Variance	Mean Sensitivity	Variance Sensitivity
1	1.8174	531.4032	0.6402	0.6716
2	0.0003 0 1.6899e-05	0.0001	2.1359e-08	
3	-4.2337e-06	5.921e-10	1.4915e-06	7.4836e-13
4	-0.0276	72.0510	0.0097	0.0910
5	-0.0042	0.0005	0.0014	7.4980e-07
6	-0.0258	72.1023	0.0090	0.0911
7	0.9143	71.5599	0.3221	0.0904
8	0.0030	16.7087	0.0010	0.0211
7.8	-0.0011	0.2449	0.0003	0.0003
6.8	-0.0012	0.2390	0.0004	0.0003
6.7	-0.0013	1.0657	0.00048	0.0013
5.8	-3.2310e-06	8.7318e-06	1.1382e-06	1.1036e-08
5.7	-0.0001	4.3686e-05	6.1972e-05	5.5215e-08
5.6	3.2455e-07	9.3983e-06	1.1433e-07	1.1879e-08
4.8	0.00027276	0.2379	9.6092e-05	0.0003
4.7	-0.0006	1.0646	0.0002	0.0013
4.5	-9.5909e-06	9.4325e-06	3.3788e-06	1.1922e-08
1.8	0.0045	1.7665	0.0015	0.0022
1.7	0.0153	7.5374	0.0053	0.0095
1.6	0.0068	7.5901	0.0024	0.0096
1.5	-0.0005	0.0004	0.0001	6.2257e-07
1.4	0.0128	7.5983	0.0045	0.0096
1.7.8	0.0008	0.0251	0.0002	3.1775e-05

Table 11.2: Results of the increment functions for the Borehole problem using residual: 0.001

APPENDIX G

Mathematical derivation of Multi surrogate approach

The collection of all increment functions, which are interpolated in the Borehole problem using the multi fidelity approach is given in Tab. 11.3 for accuracy 0.01 and in Tab. 11.4 for accuracy 0.001

Increment function	Partial Mean	Partial Variance	Mean Sensitivity	Variance Sensitivity
1	1.8167	531.4599	0.6401	0.6715
2	0.0003	1.7708e-05	0.0001	2.2375e-08
3	-4.2331e-06	5.8271e-10	1.4917e-06	7.3629e-13
4	-0.0276	72.0510	0.0097	0.0910
5	-0.0042	0.0006	0.0014	7.6253e-07
6	-0.0258	72.1023	0.0091	0.0911
7	0.9173	72.0352	0.3232	0.0910
8	0.0030	16.7087	0.0010	0.0211
7.8	-0.0008	0.1551	0.0003	0.0001
6.8	-0.0009	0.1513	0.0003	0.0001
6.7	-0.0010	0.6748	0.0003	0.0008
4.8	0.0002	0.1506	7.6487e-05	0.0001
4.7	-0.0005	0.6741	0.0001	0.0008
1.8	0.0035	1.1136	0.0012	0.0014
1.7	0.0150	8.0493	0.0053	0.0101
1.6	0.0071	8.0394	0.0025	0.0101
1.4	0.0133	8.0471	0.0046	0.0101

Table 11.3: Results of the increment functions for the borehole problem using residual: 0.01 - multi fidelity

Increment function	Partial Mean	Partial Variance	Mean Sensitivity	Variance Sensitivity
1	1.8174	531.4032	0.6397	0.6718
2	0.0003	1.7708e-05	0.0001	2.2389e-08
3	-4.2331e-06	5.8271e-10	1.4901e-06	7.3675e-13
4	-0.0276	72.0510	0.0097	0.0910
5	-0.0042	0.0006	0.0014	7.6300e-07
6	-0.0258	72.1023	0.0090	0.0911
7	0.9177	71.4914	0.3230	0.0903
8	0.0030	16.7087	0.0010	0.0211
7.8	-0.0008	0.1551	0.0003	0.0001
6.8	-0.0009	0.1513	0.0003	0.0001
6.7	-0.0013	1.0657	9.0506e-07	0.0013
5.8	-2.5711e-06	5.5295e-06	4.9277e-05	6.9912e-09
5.7	-0.0001	2.7664e-05	9.0914e-08	3.4978e-08
5.6	2.5827e-07	5.9515e-06	7.6408e-05	7.5248e-09
4.8	0.0002	0.1506	9.6092e-05	0.0001
4.7	-0.0006	1.0646	0.0002	0.0013
4.5	-7.6322e-06	5.9732e-06	2.6867e-06	7.5522e-09
1.8	0.0045	1.7649	0.0015	0.0022
1.7	0.0150	7.6117	0.0052	0.0096
1.6	0.0068	7.5866	0.0024	0.0095
1.5	-0.0004	0.0003	0.0001	3.9438e-07
1.4	0.0128	7.5948	0.0045	0.0096
1.7.8	0.0006	0.0159	0.0002	2.0128e-05

Table 11.4: Results of the increment functions for the borehole problem using residual: 0.001 - multi fidelity