THÈSE DE DOCTORAT DE L'UNIVERSITÉ PIERRE ET MARIE CURIE

Spécialité: Mécanique des Fluides

École Doctorale de Sciences Mécaniques, Acoustique, Electronique et Robotique de Paris (ED 391)

Présenté par:
Jan Willem VAN LANGENHOVE

Pour obtenir le grade de
DOCTEUR DE L'UNIVERSITÉ PIERRE ET MARIE CURIE

Adaptive control of deterministic and stochastic approximation errors in simulations of compressible flow

---

Dirigée par Anca BELME et Didier LUCOR à l'Institut Jean le Rond ∂'Alembert
Soutenue à l'UPMC le 25 octobre 2017 devant le jury composé de

MM. P. CINNELLA	Prof. ENSAM	Rapporteur
MR. T. COUPEZ	Prof. Ecole Centrale de Nantes	Rapporteur
MR. P. FREY	Prof. UPMC	Examinateur
MR. R. DWIGHT	Assistant Professor TU Delft	Examinateur
MR. F. ALAUZET	DR CNRS, INRIA	Examinateur
MR. J. PETER	Maître de Recherche ONERA	Examinateur
MM. A. BELME	Maître de Conférences UPMC	Directeur de thèse
MR. D. LUCOR	DR CNRS, LIMSI	Directeur de thèse

Université Pierre et Marie Curie

Abstract

Fluid Mechanics
Institut Jean le Rond d'Alembert

Doctor of Philosophy

Adaptive control of deterministic and stochastic approximation errors in simulations of compressible flow

by Jan Willem VAN LANGENHOVE

This thesis addresses several open questions regarding robust adaptive uncertainty quantification. More precisely, we have considered two situations when propagating uncertainties through an approximate model: (a) no information regarding the deterministic error contained in each sample is available and (b) the case where estimates of this error can be computed and used to reduce the overall error.

Concerning (a), we propose a non-iterative robust numerical method for the non-intrusive uncertainty quantification of multivariate stochastic problems with reasonably compressible polynomial representations. The approximation is robust to data outliers or noisy evaluations which do not fall under the regularity assumption of a stochastic truncation error but pertains to a more complete error model, capable of handling interpretations of physical/computational model (or measurement) errors. The method relies on the cross-validation of a pseudospectral projection of the response on generalized Polynomial Chaos approximation bases; this allows an initial model selection and assessment yielding a preconditioned response. We then apply a $\mathbf{L}_1$–penalized regression to the preconditioned response variable. Nonlinear test cases have shown this approximation to be more effective in reducing the effect of scattered data outliers than standard compressed sensing techniques and of comparable efficiency to iterated robust regression techniques.

With respect to be (b), we focus on parametrized complex flow simulations leading to output responses with low regularity and discontinuities for which a novel adaptive approach to control errors in stochastic CFD problems is proposed. More precisely, we focus on errors committed on a stochastic quantity of interest. A goal-based *a priori* error estimation is proposed as indicator for adaptivity in the deterministic space while adaptivity in the stochastic (uncertain parameter) space is driven by a control of the interpolation error. An optimization problem for optimal control of the distinct sources of errors is formulated in the continuous framework of Riemannian metric space. From the solution of this problem, a unit mesh in the Riemannian metric can be obtained, which upon transformation to the Euclidean space yields a stretched anisotropic mesh that automatically takes into account the anisotropic features of the solution field. This approach, initially developed for mesh adaptation in CFD problems, is extended to the stochastic space to adaptively refine the surrogate model consisting of linear stochastic simplices. We demonstrate the capability of this method to accurately capture discontinuities (also) in the stochastic response for compressible CFD problems with two uncertain flow parameters with different probability distributions.

Contents

Abstract **iii**

1 Introduction **1**
- 1.1 Uncertainty quantification 1
- 1.2 Robust adaptive UQ 2
  - 1.2.1 Model error as noise 3
  - 1.2.2 Error control 3
- 1.3 Thesis overview 4

2 Robust uncertainty quantification in absence of sample error information **5**
- 2.1 Introduction 5
- 2.2 Different formulations for the generalized Polynomial Chaos approximation 8
  - 2.2.1 Galerkin projection 9
  - 2.2.2 Least-Squares Minimization 9
    - Robust regression 10
    - Least squares minimization with $\mathbf{L}_1-$regularization 11
  - 2.2.3 Model validation 12
- 2.3 Preconditioning and weight selection 12
  - 2.3.1 Cross-validation 13
  - 2.3.2 Quadrature-based leave-one-out error estimation 14
  - 2.3.3 LOO-weighted preconditioned $\mathbf{L}_1-$minimization approximation . . 16
- 2.4 Numerical Examples 18
  - 2.4.1 Sparse polynomial test functions 18
  - 2.4.2 Higher-dimensional non-polynomial test function 21
  - 2.4.3 2D compressible, inviscid flow test cases 22
    - NACA0012 airfoil 23
    - Scramjet engine internal flow 25
- 2.5 Conclusions 27

3 Riemannian metric framework for anisotropic adaptation **31**
  - 3.0.1 Overview of this chapter 32
- 3.1 Metrics in the context of mesh adaptation 32
  - 3.1.1 Euclidean metric space 32
    - Geometric interpretation 33
  - 3.1.2 Riemannian metric space 34
- 3.2 Continuous mesh model 36
  - 3.2.1 From Riemannian metric spaces to anisotropic unit meshes 38
- 3.3 Continuous linear interpolation error 39
- 3.4 Defining the optimal mesh 40

4 Metric-based stochastic error control 43
- 4.1 Introduction 43
- 4.2 Formal error estimation 44
  - 4.2.1 Stochastic error estimate 46
- 4.3 Stochastic continuous model 47
  - 4.3.1 Formal resolution 49
- 4.4 Stochastic adaptive strategy 52
  - 4.4.1 Simplex-Stochastic Collocation 52
- 4.5 Numerical Applications 56
  - 4.5.1 Validation of stochastic test functions 56
    - Test 1: uniform probability density function in two dimensions . . . 57
    - Test 2: discontinuous probability density function in two dimensions 61
    - Test 3: uniform distributions in three dimensions 65
  - 4.5.2 Fluid mechanics stochastic piston problem 67
    - Two uncertain parameters 68
    - Three uncertain parameters 73
- 4.6 Conclusions 74

5 Metric-based adaptive error control in deterministic and stochastic spaces 77
- 5.1 Introduction 77
- 5.2 Goal-oriented deterministic error estimate 78
  - 5.2.1 Application to nonlinear conservation laws: compressible Euler flow 80
  - 5.2.2 Continuous error model 81
- 5.3 Adaptive strategies for total error control 82
  - 5.3.1 Deterministic adaptive strategy 83
  - 5.3.2 Optimal adaptive strategy 83
- 5.4 Numerical applications 85
  - 5.4.1 Supersonic inlet/isolator 85
  - 5.4.2 Scramjet internal flow 96
  - 5.4.3 NACA0012 105
- 5.5 Conclusions 107

6 Conclusions and Perspectives 109
- 6.1 Perspectives 110

Bibliography 113

List of Figures

2.1 Schematic illustration of quadrature breaking down for leave-one-out error estimation (here for a Gauss-Legendre grid in $D = 2$ dimensions). Quadrature weight magnitudes (before adjustment) are proportional to the circle diameters. . . . 16

2.2 Continuous response surfaces of sparse Legendre polynomials obtained from different approximation methods based on a CC grid of level $l = 5$ (left) and KP $l = 4$ (right). Randomly selected outliers data are identified by red circles. Green circles represent normally distributed data samples subject to background stochastic noise. The SC curve refers the stochastic collocation based on Lagrange interpolation; IRLS is an iterative reweighted least square approximation with the same predictors as the gPC pseudospectral representation. The reference curve is the target noiseless QoI response. . . 18

2.3 In reference to the results of Figure 2.2-(left): (a) overview of the model errors vs. polynomial total degree p, (b) polynomial coefficients magnitude u_j^2 vs. p, (c) sample weights and finally model cross-validated mean square errors vs. λ for LASSO (d) and for weighted-LASSO (e). . . . 20

2.4 Same caption as in Figure 2.3, but in reference to the results of Figure 2.2-(right). . . . 21

2.5 Comparison of different approximation errors: i.e. μ (a), σ^2 (b), $\mathbf{L}_1$ (c) and $\mathbf{L}_\infty$ (d), of a three-dimensional non-polynomial noisy Genz corner-peak function obtained from different methods relative to the noise-free reference solution. All approximations are based on 7^3 KP sampling grid. Data outliers affect $\kappa = 15\%$ of the total number of samples and $\sigma_\chi = 0.03$. The test is repeated 500 times for different random initial conditions and stochastic noise. . . . 22

2.6 NACA0012: density field closeup for different angles of attack: $\mathrm{AoA} = 0$ (left), $\mathrm{AoA} \approx 6$ (middle) and $\mathrm{AoA} = 8$ (right). Note the presence of small shocks close to the leading edge at large angles of attack. Computational meshes are not displayed but have been adapted and refined to capture all relevant flow features. . . . 24

2.7 NACA0012: averaged stagnation pressure P_a response surfaces vs. AoA, obtained from different approximation methods based on a level $l = 5$ Kronrod-Patterson data sampling. Two classes of discretization meshes of the Euler flow are investigated: – uniform coarse mesh (red circles) vs. – fine mesh adapted to each flow incidence (green stars). . . . 24

2.8 NACA0012: averaged stagnation pressure P_a response surfaces vs. AoA and M_∞, based on a 9^2 Clenshaw-Curtis data sampling: – pseudospectral gPC expansion with $p = 4$ (left) and – LOO-weighted LASSO (right). Each deterministic CFD simulation is performed on a non-adapted mesh with ~ 7000 cells. . . . 25

2.9 Same caption as the previous figure but based on a 17^2 Clenshaw-Curtis data sampling: – pseudospectral gPC expansion with $p = 8$ (left) and – LOO-weighted LASSO (right). 25
2.10 Scramjet geometry; the QoI is integrated on Γ. 26
2.11 Distributions of iso-Mach contours for an inviscid Mach number $M_\infty = 3$ scramjet inlet at zero angle of attack; (a): uniform coarse mesh (3818 elements), (b) adapted fine mesh (11707 elements). 27
2.12 Scramjet integrated outlet pressure response surfaces vs. M_∞ and α, based on a 17^2 CC data sampling: pseudospectral gPC expansion with $p = 8$ (a), LASSO (b) and LOO-weighted LASSO (c-d) with $p = 15$. Most Euler deterministic CFD simulations (white circles) are performed on a non-adapted mesh (cf. 2.11-(a)), except for the simulations* (red circles) carried out on QoI-oriented adapted meshes (example of such a mesh in 2.11-(b)). 28

3.1 Mapping corresponding to the metric $\mathcal{M}$. On the left $h_i = \lambda_i^{-\frac{1}{2}}$ (λ_i being the eigenvalues of $\mathcal{M}$) and on the right $h_i = 1$ (as the eigenvalues of $\mathcal{I}_2$ are all equal to one). 34
3.2 The surface $\mathcal{S}$ corresponding to $z = \sigma(x, y) = \sin(x^2 + y^2)$ (left) and a visualisation of the Riemannian metric space $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in[-\frac{\pi}{2},\frac{\pi}{2}]\times[-\frac{\pi}{2},\frac{\pi}{2}]}$ on the right where the unit ball corresponding to $\mathcal{M}(\mathbf{x})$ is plotted at different points (the size of the ellipses has been scaled down to allow for easier visualization). The length of the path between two points on Ω (right) computed w.r.t. $\mathcal{M}(\mathbf{x})$ is equal to the length of the curved path projected onto the surface $\mathcal{S}$ (left). 36

4.1 Lagrange basis of degree 1 plotted on four elements. On the left, the Lagrange basis function corresponding to the first vertex of each element is drawn in red (element 1), green (element 2), yellow (element 3) and blue (element 4). In the middle frame the Lagrange basis function corresponding to the second vertex is drawn (same colours as before) and on the right the Lagrange basis functions corresponding to the third vertex is presented (colours as before). The open red circles show the higher degree quadrature on which $\rho_{\boldsymbol{\xi}}$ will be evaluated in order to compute the integration weights corresponding to the three vertices of each element. In this example the fine quadrature is a third degree Newton-Cotes quadrature, but quadratures up to degree 8 have been implemented in 2D and up to degree 6 in 3D. 55
4.2 Newton-Cotes quadrature points on the 2D reference element. 55
4.3 Newton-Cotes quadrature points on the 3D reference element. 56
4.4 Test function with multiple discontinuities. 57
4.5 Approximated solution on the mesh at each refinement. 58
4.6 Convergence of the interpolation error and the error estimate. The interpolation error is computed on a third degree Newton-Cotes quadrature, a sort of subgrid constructed in order to evaluate the real interpolation error. . . . 59
4.7 Histogram of the exponent κ and constant β for 50 runs of the test problem with $N_{\xi,0} = 30$. For each run the convergence of the error estimate and the actual evaluated error is fitted to βN_ξ^κ, the histograms show the distribution of the parameters β and κ. 60
4.8 The mesh plotted is the one obtained at the end of three iteration steps and for which the interpolation error is shown in 4.10b. The color background represents the discontinuous probability density function. 62

4.9 Approximated solution on the mesh at each refinement step where the discontinuous probability density function is taken into account. 63
4.10 Convergence of the interpolation error and the error estimate. The interpolation error is computed on a third degree Newton-Cotes quadrature, a sort of subgrid constructed in order to evaluate the real interpolation error. . . . 63
4.11 Histogram of the exponent κ and constant β for 50 runs of the test problem with $N_{\xi,0} = 30$; similar to Figure 4.7. 65
4.12 The adapted $3D$ mesh with 4290 samples. 66
4.13 Convergence of the interpolation error and the error estimate with three uncertain variables. The interpolation error is computed on a third degree Newton-Cotes quadrature, a sort of subgrid constructed in order to evaluate the real interpolation error. 67
4.14 Setup of the piston problem (figure from [Witteveen, Loeven, and Bijl, 2009]). 68
4.15 Adaptive refinement of the response surface. The joint probability density function of u_{piston} and p_{pre} is shown on the top left plot. 69
4.16 Piston problem $2D$: Convergence of the error of the mean, variance and the interpolation error $e_{\mathcal{M}^{opt}}$. The results obtained by [Witteveen, Loeven, and Bijl, 2009] are included for comparison. The legend entry *Error mean* signifies the error of $\mathbb{E}[QoI]$ ($|\mathbb{E}[QoI]_{MC} - \mathbb{E}[QoI_{surrogate}]|$), and analogously the entry *Error variance* the error of $\text{Var}[QoI]$ while $\mathbb{E}[\text{interp err}]$ is $\overline{\eta}$, the expectation of the interpolation error estimate. 69
4.17 On the left: mesh adaptation sequence when the two random variables follow a lognormal distribution with $\mu_{u_{\text{piston}}} = \mu_{p_{\text{pre}}} = 1$ and a coefficient of variation of $CoV = 10\%$; the underlying contour map is a representation of the probability density function. On the right: mesh adaptation sequence when the two random variables follow a uniform distribution. 71
4.18 Piston problem $2D$: Response surface obtained with the ME-gPC method for the piston problem with two uncertain variables following a uniform probability distribution. 72
4.19 Comparison of convergence of metric-based SSC and ME-gPC for the piston problem with two uncertain variables both following a uniform probability distribution. 73
4.20 Piston problem $3D$: Convergence of the error of the mean, variance and the interpolation error $e_{\mathcal{M}^{opt}}$. The results obtained by [Witteveen, Loeven, and Bijl, 2009] are included for comparison. 74

5.1 Schematic illustrating the mesh adaptation process. The couple (mesh, solution) is denoted $(\mathcal{H}_i, \mathcal{S}_i)$ where the subscript i denotes the fixed-point iteration number; $\mathcal{S}_0^0$ is the initial solution on mesh $\mathcal{H}_0$ whereas $\mathcal{S}_i^0$ is the initial solution interpolated on the mesh $\mathcal{H}_i$. Each discrete mesh $\mathcal{H}_i$ is based on the continuous metric $\mathcal{M}_i$. 83
5.2 Inlet problem: geometry configuration with an illustration of the targeted area Γ. 85
5.3 Overview of iso-density lines for some selected parameter sets on initial, non-adapted meshes. 87
5.4 Overview of iso-density lines for the same parameter sets as shown in Figure 5.3, but this time a goal-based mesh adaptation was preformed on each mesh each mesh with a mesh complexity of $\mathcal{C} = 1000$. 88
5.5 Zoom of the results shown before in Figure 5.4, showing the position where the shock hits the lower wall relative to the pressure sensor. 89
5.6 Response surface of the pressure coefficient on the sensor. 90

5.7 Error control by adaptive refinement (multiple views of the same plot). The total error is controlled by either refining the mesh complexity each deterministic computation using Algorithm 2 (dotted lines) or by increasing the mesh complexity in the parameter space following Algorithm 1 (full lines). In the first computation the maximal mesh complexity in each deterministic computation was set to $\mathcal{C}_{\mathbf{x}}^{max} = 2000$ while in a second computation it was set to 128000. 91

5.8 Behaviour of $\overline{\varepsilon}$ and $\overline{\eta}$ for the supersonic inlet test case with $\mathcal{C}_{\mathbf{x}}^{max} = 128000$. Dotted lines indicate deterministic refinement following Algorithm 2 while full lines represent refinement in the parametric space following Algorithm 1. 93

5.9 Inlet step 0: Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space. 94

5.10 Inlet step 1: Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space. 94

5.11 Inlet step 2: Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space. 95

5.12 Inlet step 3: Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space. 95

5.13 Inlet step 4: Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space. 96

5.14 Inlet problem: some results of the coupled adaptation strategy for error control. Results are displayed at the end of step 0 (column (a)), step 2 (b) and step 19 (c). Adapted stochastic partitions (first row) and isocontour maps are shown in the parametric space: the surface response of the approximated QoI $j(M_\infty, \alpha)$ (2nd row), – spatial discretization error maps $\varepsilon(M_\infty, \alpha)$ (3rd row) and – corresponding optimal spatial complexity map $\mathcal{C}_{\boldsymbol{x}}(M_\infty, \alpha)$ (bottom row). 97

5.16 Scramjet: Density field solution for $M_\infty = 3$ and $\alpha = 0$: uniform mesh (left image) vs. goal-based adapted mesh (right image) 98

5.15 Scramjet geometry; the QoI is integrated on Γ. 98

5.17 Response surfaces of the QoI for the scramjet case. On the left, the Delaunay triangulation on the randomly drawn samples; in the middle the response surface corresponding to the first data point in Figure 5.18; and on the right the final mesh corresponding to the last data point in aforementioned figures. 99

5.18 Behaviour of $\overline{\varepsilon}$ and $\overline{\eta}$ for the scramjet test case with $\mathcal{C}_{\mathbf{x}}^{max} = 128000$. Dotted lines indicate deterministic refinement following Algorithm 2 while full lines represent refinement in the parametric space following Algorithm 1. . . 100

5.19 Map showing the error on each deterministic computation and the mesh complexity of those computations for the scramjet case on the initial Delaunay triangulation on the parameter space. 101

5.20 Scramjet: Deterministic adapted meshes at $M_\infty = 3$. By line: zoom in the middle of the geometry vs. zoom at the outflow, for deterministic complexity $\mathcal{C}_{\boldsymbol{x}} = 8000$, 16000 and 32000. 102

5.21 Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space for the scramjet case. 102

5.22 Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space for the scramjet case. 103

5.23 Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space for the scramjet case. 104
5.24 Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space for the scramjet case. 104
5.25 NACA0012 with pressure sensor. 105
5.26 NACA0012: Mach field for $M_\infty = 0.77$ and $\alpha = 1$ (A), vs. Mach field for $M_\infty = 0.83$ and $\alpha = 2$ (B). 106
5.27 NACA0012 with pressure sensor. 106
5.28 NACA0012: Evolution of the expectation and variance during refinement of the stochastic mesh. 107

List of Tables

2.1 In reference to the results of Figure 2.2-(left): overview of different functional error indicators for a sparse polynomial test case with (top) and without (bottom) stochastic background noise and for different choices of α. The best overall result in bold. $p^{(TD)}$ is the chosen total degree of the polynomial approximation basis; for the weighted-LASSO approach, the value in parenthesis is the optimal order obtained from the original pseudospectral gPC representation resulting in the lowest overall cross-validation LOO error. . . 19
2.2 Same caption as in Table 2.1, but in reference to the results of Figure 2.2-(right). 20
2.3 Similar caption as Table 2.1 for $D = 3$ dimensions Genz corner-peak functional. But this time, all errors are averaged as the test cases have been repeated 500 times for different initial conditions. 22
2.4 Same caption as Table 2.1 for $D = 5$ dimensions Genz corner-peak test case. 23

3.1 Continuous-discrete duality 39

4.1 A least squares fit of the error convergence to the model βN_ξ^κ is performed. For the convergence curve of the interpolation error estimate, the parameters are called β_{est} and κ_{est}, for the evaluated interpolation error they are called β_{eval} and κ_{eval} (interpolation errors were evaluated on a NC subgrid quadrature of degree 5). Both the average and standard deviation after 50 repetitions are given for the case when the initial random mesh is composed of $N_{\xi,0} = 10$ samples and for the case where $N_{\xi,0} = 30$. 60
4.2 Similar to Table 4.1 except that now the underlying pdf is the discontinuous pdf given by (4.31). 64
4.3 Overview of the sum of the quadrature weights for the discontinuous pdf (4.31) on the meshes in Figure 4.9. The subgrid NC quadrature degree used to compute the weights according to (4.29) is indicated in the column on the left. 66

5.1 Inlet statistics and errors with $\mathcal{C}_x^{max} = 2000$. 92
5.2 Inlet statistics and errors with $\mathcal{C}_x^{max} = 128000$. 92
5.3 Scramjet statistics and errors. 105
5.4 NACA0012: Evolution of deterministic and stochastic error components. . . 107

List of Abbreviations

CC	**C**lenshaw-**C**urtis
CFD	**C**omputational **F**luid **D**ynamics
CoV	**C**oefficient **o**f **V**ariation
CS	**C**ompressed **S**ensing
DoE	**D**esign **o**f **E**xperiments
ENO	**E**ssentially **N**on-**O**scillatory
gPC	**g**eneralized **P**olynomial **C**haos
IRLS	**I**teratively **R**eweighted **L**east **S**quares
KP	**K**ronrod-**P**atterson
LASSO	**L**east **A**bsolute **S**hrinkage (and) **S**election **O**perator
LHS	**L**atin **H**ypercube **S**ampling
LS	**L**east **S**quares
LOO	**L**eave-**O**ne-**O**ut
MAD	**M**edian **A**bsolute **D**eviation
MADN	Normalized **M**edian **A**bsolute **D**eviation
Mb-SSC	**M**etric-**b**ased **S**implex **S**tochastic **C**ollocation
MC	**M**onte **C**arlo
ME-PCM	**M**ulti-**E**lement **P**robabilistic **C**ollocation **M**ethod
MME-SC	**M**inimal **M**ulti-**E**lement **S**tochastic **C**ollocation
MSE	**M**ean **S**quared **E**rror
NC	**N**ewton-**C**otes
PCE	**P**olynomial **C**haos **E**xpansion
pdf	**p**robability **d**ensity **f**unction
QoI	**Q**uantity **o**f **I**nterest
RM	**R**iemannian **M**etric
SC	**S**tochastic **C**ollocation
SSC	**S**implex **S**tochastic **C**ollocation
UQ	**U**ncertainty **Q**uantification
WLS	**W**eighted **L**east **S**quares

Chapter 1

Introduction

Science and engineering have benefited greatly from the developments in numerical methods and computing power over the last few decades. As a consequence, numerical simulation has become an essential ingredient of the design process. In this thesis, the field of interest will be that of Computational Fluid Dynamics (CFD). There, real-life applications are often defined on complex, $3D$ geometries for which exact solutions are rarely available. Discretization techniques and numerical schemes introduce errors and these numerical approximation errors need to be controlled. Besides these errors there can also be numerous uncertainties.

To distinguish *uncertainty* from *errors* we recall the definitions from AIAA's "*Guide for the Verification and Validation of Computational Fluid Dynamics Simulations*", where uncertainty is defined as:
"A potential deficiency in any phrase or activity of the modeling process that is due to the lack of knowledge" and error as:
" A recognizable deficiency in any phase or activity of modeling and simulation that is not due to lack of knowledge".

An example of an uncertainty occurs when one needs to specify the inputs for the simulation: initial conditions, boundary conditions, and sometimes parameters such as geometry parametrization or even model parameters (for instance parameters of a turbulence model). Errors in these inputs due to a lack of knowledge can lead to solutions that no longer represent the real-world flow one intended to simulate. Hence, in order to instill confidence in the predictions obtained from computational models, both the uncertainties and errors need to be accounted for.

1.1 Uncertainty quantification

The aim of uncertainty quantification (UQ) is to numerically evaluate the *stochastic* response of some physical quantity of interest (QoI) $y(\boldsymbol{\xi})$, dependent upon various uncertain model parameters $\boldsymbol{\xi} := (\xi_1, \ldots, \xi_D)$. In this thesis we will always assume that these uncertainty models are correct and the focus will solely be on *non-intrusive* uncertainty propagation techniques: i.e. the sampling of N model solutions is done using any legacy solver for the deterministic problem as a black box.

Ideally one would like to know the full distribution of the stochastic output response, but in practice this is often unattainable. Instead one usually tries to estimate the statistical moments that characterize the output distribution. The computation of statistical moments is essentially an integration problem in a domain of possible high dimensionality where the probability measure needs to be taken into account. The traditional mainstay for this sort of application is the Monte Carlo simulation. It is robust, simple to implement and easy to parallelize. The convergence rate however is rather slow at $(1/\sqrt{N})$ where N is the number of samples. Interestingly, the convergence of the Monte Carlo method does not depend on the dimension (i.e. the number of uncertain parameters) of the stochastic space. In spite of this, because of the slow convergence rate, the number of samples needed for accurate results so high that it often becomes prohibitively expensive for practical applications.

When the number of uncertain input sources of a complex model is too large ($D \gg 1$), the efficient propagation of these uncertainties through the system remains an open problem due to the so-called curse of dimensionality. In any case, as the computational cost of a single simulation model is often high, e.g. for computational fluid dynamics simulations (CFD), a compulsory approach for UQ is to build a surrogate model $\tilde{y}$ that approximates the exact response of the QoI as accurately as possible based on the smallest number of *observations* or *samples* [Bijl et al., 2013]. Once the samples are acquired, the construction and interrogation of the surrogate model itself should be computationally efficient so that the predictive capability of the surrogate model is fully harnessed. A surrogate model not only provides access to the statistics of interest but also provides a continuous representation of the stochastic QoI response. This latter fact can a requirement if one is interested in Robust Design Optimization (RDO) or Bayesian calibration.

Different methods are available to construct these surrogate models, examples are Gaussian processes (Kriging) [Williams and Rasmussen, 2006], Support Vector Machines [Smola and Schölkopf, 2004], stochastic interpolation [Tatang et al., 1997; Xiu and Hesthaven, 2005a] or stochastic spectral methods such as polynomial-based representations. Of the latter class are generalized Polynomial Chaos (gPC) Xiu and Karniadakis, 2002 approximations which consist in constructing a parametrized polynomial approximation of the QoI response. The implementation using a Stochastic Galerkin (SG) method is a so called *intrusive* method which requires modifications to existing code and results in a large system of coupled equations to be solved. The nonintrusive variant of this method, uses a pseudospectral projection or a linear regression to find the coefficients for this polynomial basis, which is constructed to be orthogonal w.r.t. the underlying probability density. Stochastic Collocation (SC) methods on the other hand, aim at constructing interpolation functions for given coefficients [Babuška, Nobile, and Tempone, 2007; Xiu and Hesthaven, 2005b]. Both PCE and SC methods allow for the use of legacy solvers as a black box similar to MC methods.

1.2 Robust adaptive UQ

When the problem under study is sensitive to variations in the input parameters, small variations in these parameters can yield large, nonlinear changes and discontinuities in the stochastic response. The presence of shock waves in the physical solution or sudden regime changes then propagate to the stochastic space. In that case the global polynomial approximation methods mentioned above can suffer from Gibbs oscillations which severely degrade the response surface. More robust alternatives to these global polynomial approximations make use of $h-$ and $hp-$ adaptation. Such methods are the Multi-Element gPC (ME-gPC) method [Wan and Karniadakis, 2005; Wan and Karniadakis, 2006], and the Multi-Element Probabilistic Collocation Method (ME-PCM) [Foo and Karniadakis, 2010; Foo, Wan, and Karniadakis, 2008] which use a decomposition into hypercubes. A similar decomposition but this time based on wavelets and not on polynomials has been proposed in [Le Maıtre et al., 2004a; Le Maıtre et al., 2004b]. Different decomposition methods have been proposed in the Minimal Multi-Element Stochastic Collocation (MME-SC)

[Jakeman, Narayan, and Xiu, 2013] and the Simplex-Stochastic Collocation (SSC) method [Witteveen, Loeven, and Bijl, 2009]. In the MME-SC method a discontinuity detector is used to decompose the domain into elements of arbitrary shape following the discontinuity lines while the SSC method obtains a continuous representation of the stochastic response surface by forming a tessellation of simplices similar to the meshes used to discretize the physical domain. The anisotropic nature of the discontinuous features in the response surface are well captured by the MME-SC and SSC methods, but the aforementioned Multi-Element methods are limited by their hypercube decomposition and the degrading effects will still be present in the elements traversed by a discontinuity. Furthermore, these Multi-Element methods place the samples following a tensor structure resulting in a fast increase in the number of samples as the number of elements increases and on top of that, not all samples may be reused after the splitting of an element.

Furthermore, as usually no analytical solution is available for realistic engineering problems, the solver will compute an approximate solution. Thus, every sample in the stochastic space (the QoI from a deterministic computation at a fixed combination of the random input parameters) will contain an error. This deterministic error affecting each sample is called the model error and its magnitude will vary from sample to sample. Note that in the interpretation used in this thesis, model error encompasses both the modelling error (discrepancy between the mathematical model and the physics) and the numerical approximation error present in the simulation result of the model.

Two cases can be distinguished: the case where the deterministic solver provides information on the error contained in each sample (for instance by computing an error estimate) and the case where no error information is available.

1.2.1 Model error as noise

When the model error is not explicitly available, an often used approach is to see it as an unknown noise corrupting the samples in the stochastic space. In literature, this noise is mostly assumed to be a centered independent identically distributed (i.i.d.) random variable with uniformly bounded variance. In practice however, the model error rarely adheres to this assumption. In real life, model errors are often biased, not uncorrelated and uniform nor are they normally distributed. Reducing the impart of such noise that goes beyond the usual assumptions is a question that will be addressed in this thesis.

1.2.2 Error control

In the case where error estimates for the deterministic computations are available, one can control the error contained in each sample using error control methods such as mesh adaptation. Note that this will control only the numerical approximation error and not the modelling error. One is often interested in a specific Quantity of Interest (QoI) that is a functional of the solution field (e.g. the drag or lift of an airfoil). In this case goal-oriented error estimates for functional outputs may be used to drive the mesh adaptation. These error estimates require the solution of an adjoint problem which takes into account the QoI. While there is a cost associated to solving this additional adjoint problem, goal-based mesh adaptation allows for a faster convergence in the QoI. For a more complete overview of mesh adaptation the reader is referred to [Alauzet and Loseille, 2016a] and references therein.

An open question remains: is it better to adapt the surrogate model (add samples for instance) or to refine existing samples, and in the latter case, which existing samples should be refined by how much? To formulate an answer to this question it is essential to have error indicators of both the deterministic errors and the errors contained surrogate model on the stochastic space. In order to have computable error estimates on the stochastic space, some authors have proposed an extension of existing a posteriori goal-oriented error estimators to the stochastic space [Almeida and Oden, 2010; Butler, Constantine, and Wildey, 2012; Butler, Dawson, and Wildey, 2011; Le Maître and Knio, 2010; Mathelin and Le Maître, 2007]. The only publication demonstrating error splitting and adaptivity in both spaces is [Bryant, Prudhomme, and Wildey, 2015].

1.3 Thesis overview

The previous sections painted the part of the UQ landscape within this thesis can be situated. Within that scope some deficiencies in existing methods were mentioned and open questions where raised.

The question of how to deal with model error appearing as noise in the stochastic space where the noise not necessarily follows some predefined distribution, is addressed in Chapter 2. In that chapter a numerical approach is proposed that automatically detects noisy samples and data outliers and weighs them with a low level of confidence. For the detection use is made of the leave-one-out (LOO) technique. LOO error estimation is a fundamental tool in statistical learning theory [Hastie, Tibshirani, and Friedman, 2009] and has previously been used for basis selection of gPC expansions [Blatman and Sudret, 2011; Jakeman, Eldred, and Sargsyan, 2015]. The LOO technique is classically used in combination with an (unstructured) sampling approach, and is unavailable for UQ methods relying on structured grids made up of tensorizations of quadrature rules. A remedy will be proposed to overcome the loss of quadrature power resulting from an incomplete quadrature, relatively recent results from the field of Compressed Sensing (CS) are exploited. Using this approach, LOO error estimation also becomes available on quadrature grids.

Chapter 3 contains a review of the Riemannian metric framework in which a continuous model for discrete meshes is introduced. The basic idea is to generate a unit mesh in the Riemannian metric (a mesh where all edges have length ≈ 1), which upon transformation to Euclidean space will yield an anisotropic adapted mesh. This mesh adaptation approach, called metric-based mesh adaptation, was developed for use in CFD applications [Loseille, Dervieux, and Alauzet, 2010; Loseille and Alauzet, 2011a; Loseille and Alauzet, 2011b] and will be extended to the stochastic space in Chapter 4. The aim in that chapter will be to demonstrate a surrogate modelling technique that automatically adds samples where needed in order to provide a continuous representation of the response surface using a simplex tessellation, even when this response surface contains (multiple) anisotropic discontinuities. In contrast to the MME-SC method it does not rely on a discontinuity detector and actually resolves the discontinuities, meaning that the surrogate model includes the discontinuities in the response surface rather then merely cutting the stochastic space around them into elements. Furthermore, the extension of the *a priori* continuous interpolation error estimate for metric-based mesh adaptation to the stochastic space will make it possible to compare deterministic errors (error present in each sample) and stochastic errors (errors in the surrogate model representation of the stochastic response). In Chapter 5 this error splitting along with adaptation in both spaces will be demonstrated on compressible flow problems. Finally, Chapter 6 will summarize the conclusions and given perspectives for future work.

Chapter 2

Robust uncertainty quantification in absence of sample error information

2.1 Introduction

The focus in this chapter will be on generalized Polynomial Chaos (gPC) expansions [Wiener, 1938; Cameron and Martin, 1947; Ghanem and Spanos, 2003; Xiu and Karniadakis, 2002] that are well-suited for functions that belong to the $\mathbf{L}_2$ space, and may be seen as discrete least-squares projection on a polynomial space. The gPC represents a function as a weighted linear combination of P multivariate (polynomial) basis functions that are mutually orthogonal with respect to the probability measure of the uncertain parameters.

The coefficients $\boldsymbol{u}$ in the expansion may be computed in different ways: e.g. – using pseudospectral projection together with high-order (sparse) quadratures that are efficient for functions of moderately high dimensionality [Resmini, Peter, and Lucor, 2015] or – by least-squares regression type of approach based on (random) data sampling. Written in a generic form, we have:

$$y \approx \tilde{y} = \hat{y}(\boldsymbol{\xi}) + e_T,$$

where, for a given model, $\tilde{y}$ represents some numerical simulations or “observations" of the system, $\hat{y}$ is what we call the surrogate model and e_T is the truncation error that obviously depends on $\boldsymbol{\xi}$.

Because we deal with computer experiments, the observations are in practice corrupted by *model* errors. Let us consider a family of models characterized by unobserved random variables $\boldsymbol{\chi}$ (e.g. related to mesh quality criteria). Now, y may be modeled as a functional of $(\boldsymbol{\xi}, \boldsymbol{\chi})$ via a model error e_M, and we have:

$$y(\boldsymbol{\xi}, \boldsymbol{\chi}) = \tilde{y}(\boldsymbol{\xi}) + e_M, \quad \text{where } \tilde{y}(\boldsymbol{\xi}) = \hat{y}(\boldsymbol{\xi}) + e_T(\boldsymbol{\xi}).$$

A frequent assumption found in the literature is the one of a stochastic noise model, where $e_M \equiv e_M(\boldsymbol{\chi})$ are centered i.i.d. (normal) random variables, with uniformly bounded variance. In real life, modeling errors are often biased, not uncorrelated and uniform (heteroscedasticity) and not normally distributed. Therefore, they bear a deterministic noise component that depends on $\boldsymbol{\xi}$ [Chkifa et al., 2015], such that $e_M \equiv e_M(\boldsymbol{\xi}, \boldsymbol{\chi})$. In practice, it is very hard to quantify these modeling errors as we do not directly observe $\boldsymbol{\chi}$. Moreover, the system may not be in the asymptotic regime for which we may have convergence information/estimates for the deterministic model error. The interplay between (the different scales related to) $\boldsymbol{\chi}$ and $\boldsymbol{\xi}$ is also very difficult to apprehend. Ideally, one would want to

---

Chapter based on Van Langenhove, J., D. Lucor, & A. Belme, (2016). "Robust uncertainty quantification using preconditioned least-sqaures polynomial approximations with L_1-regularization". *International Journal for Uncertainty Quantification*, 6(1);
Lucor, D., J.W. Van Langenhove, & A. Belme, (2017). Robust uncertainty quantification of CFD via weighted compressed sensing. *ERCOFTAC Bulletin 89*

marginalize over the model error in order to assess the *conditional* random variable:

$$y(\boldsymbol{\xi}) := \mathbb{E}\left[y(\boldsymbol{\xi},\boldsymbol{\chi})\,|\,\boldsymbol{\xi}\right] = \hat{y}(\boldsymbol{\xi}) + e(\boldsymbol{\xi}), \quad \text{where } \; e(\boldsymbol{\xi}) = e_T(\boldsymbol{\xi}) + \mathbb{E}\left[e_M(\boldsymbol{\xi},\boldsymbol{\chi})\,|\,\boldsymbol{\xi}\right],$$

but this approach is often out of reach [Hampton and Doostan, 2015b]. Another strategy is to perform the regression for a given model, which implicitly coincides with a given value of $\boldsymbol{\chi}$. When the dependence between the solution and the random parameters $\boldsymbol{\xi}$ is smooth, model errors are generally relatively independent from the parameters $\boldsymbol{\xi}$ and their effects often result in some form of biased predictions (e.g. under-prediction in case of model numerical diffusion). When the dependence is not smooth due to some brutal change in the solution physical regime, bifurcations, instabilities, transients, etc.[1], model or computational errors induce some local large-amplitude oscillations that may be seen as data *outliers* and are very detrimental to the stability of the stochastic quantification of the response. In this case, these large errors are unpredictably scattered and increasing the number N of samples of $\boldsymbol{\xi}$ does not help as the prediction essentially fits the model error. This is the well-known problem of *overfitting* when a model fits training data very well, but will do a poor job of predicting results for new data. A first step toward *robust* UQ in this framework would be to automatically detect the data outliers and reduce their influence in order to regularize the response on a given model basis.

A simple way of thinking about the e_M error is to relate it, for instance, to the discretization error of the computational model. Indeed, it is not always possible due to prohibitive computational cost to lower the discretization error to levels that do not play a significant role in UQ. The case of CFD simulations of a compressible flow past an airfoil at random angle of attack presented later in this chapter is a nice illustration of this setup and was the starting point of this work. For a given level of discretization (in practice a given mesh) and Mach number, a small variation in the angle of attack may induce a change in the flow from subsonic to transsonic regime, materialized by the emergence of a flow discontinuity (shock). This shock is then poorly captured if the mesh is not properly adapted in space, inducing large model errors and fluctuations in addition to the truncation error. In this case, the discretization error will strongly affect only few or a short range of the angle of attack realization values.

One of our contributions in this work is to propose a numerical approach that automatically detects data outliers and weighs them with low level of confidence. In this work, the detection and weighting is in part based on exhaustive surrogate model cross-validation namely the leave-one-out (LOO) technique. The LOO error estimation has been used before in the context of basis selection of gPC expansions [Blatman and Sudret, 2011; Jakeman, Eldred, and Sargsyan, 2015], and more generally in statistical learning theory for model selection [Hastie, Tibshirani, and Friedman, 2009]. The negative effects of the outliers on the construction of the surrogate model will then be minimized in order to avoid overfitting.

Another key aspect of this chapter is to take advantage of the sparsity of the solution structure. Indeed, the solution of high-dimensional problems is sometimes *sparse* (or near-sparse) at the stochastic level. This means that it may be accurately represented with only few terms when linearly expanded into a stochastic approximation space, such as the one encompassed by a gPC basis. In this case, the number s of dominant basis functions is small relative to the cardinality P of the full basis and the problem is said to be $s-$sparse. Promising approaches for solving this kind of problem involve compressed sensing (CS)

---

[1] Or due to some soft system faults (e.g. bit-flips), nowadays more frequent in petascale high performance computing.

techniques, also known under the names of Compressive Sensing, $\mathbf{L}_1-$minimization, convex relaxation and $\mathbf{L}_1$-regularized least-squares minimization. Relatively recent results in CS have made it clear that sparse functions can be accurately recovered from much fewer observations than necessary for classical solution methods [Candès, Romberg, and Tao, 2004; Candès and Tao, 2005; Donoho, 2006]. Interestingly, this ability is preserved in the case of sparse solutions tainted by noise, as long as it is sufficiently regular and bears a low signal-to-noise ratio [Candès, Romberg, and Tao, 2006; Donoho, Elad, and Temlyakov, 2006; Fuchs, 2005; Tropp, 2006].
Several research groups have recently been using CS in a gPC framework [Doostan and Owhadi, 2011; Mathelin and Gallivan, 2012; Yan, Guo, and Xiu, 2012; Yang and Karniadakis, 2013] and have considered this noise as the truncation error of the gPC approximation. The efficiency of this approximation depends on the type and cardinality of the gPC approximation basis selected [Blatman and Sudret, 2011; Jakeman, Eldred, and Sargsyan, 2015] and the choice of the collocation samples to be used. The most readily available literature is about sparse Legendre and Hermite polynomials with *random* sampling. For both cases, different sampling strategies are possible: – standard sampling according to the underlying probability measure, and – asymptotic sampling according to the Chebyshev measure for Legendre polynomials, and to Hermite functions for Hermite polynomials [Hampton and Doostan, 2015b; Tang and Zhou, 2014]. For s-sparse Legendre polynomial with maximal degree p, it was shown that the asymptotic relation between the number N of samples drawn according to Chebyshev distribution, s and p, guaranteeing recovery, is given by $N \asymp s\log^4(p)$ [Rauhut and Ward, 2012]. Chebyshev sampling has been shown to be superior to uniform sampling for elliptic stochastic partial differential equations of moderately high dimension ($D \sim 10$) [Yang and Karniadakis, 2013], but the results can not be generalized. In fact, Yan et al. [Yan, Guo, and Xiu, 2012] show that for high-dimensional problems, sampling according to the Chebyshev measure can become less efficient. Interestingly, in case of standard sampling, the Chebyshev probability measure may be imposed afterwards by preconditioning the $\mathbf{L}_1-$minimization problem. This approach inspired us to use data-driven preconditioning to improve approximation robustness. Finally, a recently developed sampling strategy is the *coherence-optimal* sampling [Hampton and Doostan, 2015b], which guarantees recoverability with a number of samples that is bound linearly by the number of basis functions up to a logarithmic factor.

Very recent works have investigated the efficiency of these methods for randomized *quadratures*: i.e. randomly subsampling among structured Gauss quadrature nodes [Tang and Iaccarino, 2014; Zhou, Narayan, and Xiu, 2015]. Using the bounds from [Rauhut and Ward, 2012], Tang & Iaccarino [Tang and Iaccarino, 2014] show that for an efficient recovery of the gPC Legendre expansions, the number of observations scales with the sparsity s and only logarithmically with P. When the number of random dimensions is small to moderate, and more specifically when $p > D$, it is conceivable to directly rely on *complete* structured grids inherited from full or partial (also known as sparse) tensorization of quadrature rules, which is what we propose in our contribution. Moreover, the use of these regular grids minimizes *leverage* effects in regressions due to unusual design points.

The aim in this chapter is to fully harness the capability of CS techniques for UQ, even in the presence of scattered data outliers attributable to computational model errors that do not fall under the common Gaussianity and low signal-to-noise ratio assumptions. We wish to do so by regularizing the system response for a given computational model. We propose preconditioned compressed sensing in order to build robust polynomial surrogate of the stochastic response from sampling on structured grids. More specifically, after selecting the best model by cross-validation using numerical quadrature, the weight for each observation will be based on the inverse of its contribution to the cross-validation error of this model. Using confidence in samples in the form of a weighted least squares solution has been done before, see for example [Zhou and Han, 2008]. Here however, no a priori knowledge about the scale of the noise is needed, nor does one need to know beforehand which observations have been affected by the noise. This weighting of the observations can be used in combination with any available method for constructing the surrogate model. In this study, we have opted for the use of the Least Absolute Shrinkage and Selection Operator (LASSO) technique [Tibshirani, 1996], which is known to be very robust, to compute the coefficients of the surrogate model, but as stated before, other methods can be used as well.

The structure of this chapter is as follows: section 2.2 will briefly recall the key points of the collocated stochastic spectral approximation framework with and without $\mathbf{L}_1-$regularization. This will serve mostly as an introduction for our notations. In section 2.3, we will discuss how we derive observation weights using cross-validation and how it is interwoven with the $\mathbf{L}_1-$minimization constraint. The proposed technique will be demonstrated on several test problems in section 2.4. The chapter ends with some conclusions.

2.2 Different formulations for the generalized Polynomial Chaos approximation

As stated in the introduction, gPC expansions will be used to express the surrogate model in a closed form [Ghanem and Spanos, 2003; Le Maître and Knio, 2010; Xiu and Karniadakis, 2002]. Let $(\Omega, \mathcal{B}, \mathcal{P})$ be the probability space where Ω is the space of random events ω, this domain has a σ-algebra $\mathcal{B}$ and is equipped with a probability measure $\mathcal{P}$. The vector of random parameters can be written as $\boldsymbol{\xi} \equiv \boldsymbol{\xi}(\omega) = (\xi_1, \ldots, \xi_D)$, but we will often omit the dependence on ω to simplify notation. If we consider a $D-$variate functional $y : \mathcal{I}_{\boldsymbol{\xi}} \subseteq \mathbb{R}^D \rightarrow \mathbb{R}$, then any second-order random variable $y(\boldsymbol{\xi}) \in \mathbf{L}_2(\Omega, \mathcal{B}, \mathcal{P})$, can be expressed as a gPC expansion [Xiu and Karniadakis, 2002]:

$$y(\boldsymbol{\xi}) = \sum_{j=0}^{\infty} u_{(j)} \psi_{(\mathrm{j})}(\boldsymbol{\xi}), \tag{2.1}$$

where $\psi_{(\mathrm{j})}(\boldsymbol{\xi}) = \prod_{i=1}^{D} \psi_{i,(j)}(\xi_i)$ are the multivariate basis functions that form a complete basis, orthonormal with respect to the probability measure $\rho_{\boldsymbol{\xi}}$ of the random input, and $\psi_{i,(j)}$ are the univariate basis functions along the i^{th} dimension. The convention used for subscript notations is that a subscript without parenthesis indicates the dimension index of a vector or simply the index of a basis function; subscripts between parenthesis indicate the sample number. Boldface Greek subscripts are multi-indexes. We also assume that all ξ_i are independent and thus $\rho_{\boldsymbol{\xi}} = \prod_{i=1}^{D} \rho_i(\xi_i)$. Note that Ω is a Hilbert space and that we can write its inner product in terms of the expectation operator $< y, g > \equiv \mathbb{E}[y \cdot g]$, in this case:

$$\mathbb{E}\left[y(\boldsymbol{\xi}) g(\boldsymbol{\xi})\right] = \int_{\mathcal{I}_{\boldsymbol{\xi}}} y(\boldsymbol{\xi}) g(\boldsymbol{\xi}) \rho_{\boldsymbol{\xi}} d\boldsymbol{\xi}. \tag{2.2}$$

Instead of indexing the expansion of equation (2.1) on a single integer amounting to the cardinality of the entire approximation space, one can also make use of a multi-index notation that is equivalent. If Λ_p is an index set (to be defined) for multi-index $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_D) \in \mathbb{N}_0^D$, then $\mathbb{P}_{\Lambda_p} \equiv \mathrm{span}\{\psi_{\boldsymbol{\gamma}} \mid \boldsymbol{\gamma} \in \Lambda_p\}$ and we can then write $\psi_{\boldsymbol{\gamma}}(\boldsymbol{\xi}) =$

$\prod_{i=1}^{D} \psi_{i,(\gamma_i)}(\xi_i)$ where $\psi_{i,(\gamma_i)}$ is the γ_i^{th} order basis function in dimension i. Using the notation introduced above, one can write out the truncated gPC expansion approximating y as follows:

$$y(\boldsymbol{\xi}) = \sum_{\boldsymbol{\gamma} \in \Lambda_p} u_{\boldsymbol{\gamma}} \psi_{\boldsymbol{\gamma}}(\boldsymbol{\xi}) + e_T(\boldsymbol{\xi}), \tag{2.3}$$

where $u_{\boldsymbol{\gamma}}$ are the coefficients corresponding to the $\psi_{\boldsymbol{\gamma}}$ basis[2]. We will restrict ourselves to tensor-product polynomial spaces $\mathbb{P}_{\Lambda_p}$, where Λ_p is an index set of "degree" p, and where $P = \dim(\mathbb{P}_{\Lambda_p}) \equiv \#\Lambda_p$, will denote the cardinality of the selected polynomial space. There are different ways of constructing the approximating polynomial spaces that will impact their cardinality:

- Tensor Product (TP): $\mathbb{P}^{\mathrm{TP}}_{\Lambda_p}$ with index set $\Lambda_p^{\mathrm{TP}} = \{\boldsymbol{\gamma} \in \mathbb{N}_0^D : ||\boldsymbol{\gamma}||_\infty \leq p\}$,
- Total Degree (TD): $\mathbb{P}^{\mathrm{TD}}_{\Lambda_p}$ with index set $\Lambda_p^{\mathrm{TD}} = \{\boldsymbol{\gamma} \in \mathbb{N}_0^D : ||\boldsymbol{\gamma}||_1 \leq p\}$, or
- Hyperbolic Cross (HC): $\mathbb{P}^{\mathrm{HC}}_{\Lambda_p}$ with index set $\Lambda_p^{\mathrm{HC}} = \{\boldsymbol{\gamma} \in \mathbb{N}_0^D : \prod_{i=1}^{D}(\gamma_i + 1) \leq p + 1\}$ [Shen and Wang, 2010].

In this manuscript, without any loss of generality, we will be using approximation spaces of total degree (TD), so Λ_p will refer to Λ_p^{TD} in the following.

2.2.1 Galerkin projection

The first way of determining the coefficients is by use of a Galerkin projection [Ghanem and Spanos, 2003]. One can write

$$\mathbb{E}\left[\sum_{\boldsymbol{\gamma} \in \Lambda_p} u_{\boldsymbol{\gamma}} \psi_{\boldsymbol{\gamma}} \psi_{\boldsymbol{\beta}}\right] = \mathbb{E}\left[\psi_{\boldsymbol{\beta}} y\right], \quad \forall\, \boldsymbol{\beta} \in \Lambda_p. \tag{2.4}$$

Here, with some abuse of notation, we have written the above equation as an equality instead of an approximation, the same will be done in the rest of this work. Assuming the basis is *orthonormal*, the coefficients can be found simply by computing:

$$u_{\boldsymbol{\gamma}} = \mathbb{E}\left[\psi_{\boldsymbol{\gamma}} y\right] \quad \text{with} \;\; \boldsymbol{\gamma} \in \Lambda_p, \tag{2.5}$$

making use of a quadrature in the case of a pseudospectral implementation [Le Maître and Knio, 2010].

2.2.2 Least-Squares Minimization

One can also use linear regression to compute the unknown coefficients $u_{\boldsymbol{\gamma}}$, e.g. [Choi et al., 2004; Berveiller, Sudret, and Lemaire, 2006]. The Least-Squares (LS) solution minimizes the residuals, $\boldsymbol{r} \equiv \boldsymbol{y} - \boldsymbol{\psi}_{\Lambda_p}\boldsymbol{u}$ in the $\mathbf{L}_2-$norm and may be written as an optimization problem:

$$\boldsymbol{u} = \underset{\mathbf{u} \in \mathbb{R}^P}{\operatorname{argmin}} \, ||\boldsymbol{y} - \boldsymbol{\psi}_{\Lambda_p}\mathbf{u}||_2, \tag{2.6}$$

where $\boldsymbol{\psi}_{\Lambda_p}$ is the measurement matrix corresponding to the gPC expansion in the index set Λ_p. The solution to (2.6) is obtained by computing the following system written in matrix form:

---

[2] If the functional to approximate is a random process, it may also depend on space and time and in that case the gPC coefficients will be deterministic space- and time-dependent fields.

$$\boldsymbol{u} = \left(\boldsymbol{\Psi}_{\Lambda_p}^{\mathsf{T}} \boldsymbol{\Psi}_{\Lambda_p}\right)^{-1} \boldsymbol{\Psi}_{\Lambda_p}^{\mathsf{T}} \mathbf{y}, \tag{2.7}$$

where $\mathbf{y}$ is a vector of observations of size $N \times 1$, $\boldsymbol{\Psi}_{\Lambda_p}$ the *measurement matrix* of size $N \times P$ with $\Psi_{ij} = \psi_{(\mathrm{j})}(\boldsymbol{\xi_{(i)}})$ (the j^{th} basis function evaluated at the i^{th} sample point), and $\boldsymbol{u}$ the vector of coefficients of size $P \times 1$. There has been a growing interest in understanding the conditions under which problem (2.6) leads to accurate and stable (multivariate) polynomial chaos approximations for data *randomly* and *independently* sampled according (or not) to their natural orthogonality measures [Cohen, Davenport, and Leviatan, 2013; Migliorati et al., 2014; Hampton and Doostan, 2015a]. More specifically, these studies focussed on the relation between the required number of samples and the cardinality of the approximation basis for different sampling measures. If one uses enough sampling points to be able to properly recover the orthonormality of the basis functions $\psi_{(\mathrm{j})}$, then the matrix $\left(\boldsymbol{\Psi}_{\Lambda_p}^{\mathsf{T}} \boldsymbol{\Psi}_{\Lambda_p}\right)$ is the identity matrix and the link between (2.7) and (2.5) becomes clear. The aforementioned works mainly deal with *noiseless* evaluations of the target function, and only few papers consider noisy data samples [Migliorati, Nobile, and Tempone, 2015]. In any case, response fittings based on standard or *ordinary* LS type objective functions are not robust against outliers, i.e. data samples that strongly deviate from assumptions (e.g. of normality). It is said that the LS estimator has a breakdown point of $1/N$ because just one leverage point may cause it to break down [Huber and Ronchetti, 2009]. In this case, the approximation might be biased, with an artificially inflated variance.

Robust regression

In robust statistics, robust regression is a form of analysis designed to circumvent some limitations of traditional parametric and non-parametric methods by dampening the influence of outlying cases [Huber and Ronchetti, 2009]. Most common robust regression methods fall into the class of $M-$estimators[3] which attempt to minimize the sum of a chosen *objective* (also called *loss*) function[4] of the residuals, i.e. $\sum_{j=1}^{N} \rho(\boldsymbol{r}_{(j)})$. This minimization may be equivalently written as a *weighted* LS problem; the weight of each sample being expressed via the *score* function $\upsilon(\boldsymbol{r}) \equiv \partial\rho/\partial\boldsymbol{r}$, i.e. a derivative of the objective function at that point. Because of their connection to the residual values, the weights are iteratively evaluated until numerical convergence. In this framework, iteratively re-weighted least square (IRLS) algorithms are implemented for different choices of objective functions, e.g. leading to Huber , Tukey's *bisquare*, or Cauchy estimators, etc... [Green, 1984; James O. Street and Ruppert, 1988]. In our numerical applications, we will often derive our sample weights from the Cauchy objective function: $\rho(\boldsymbol{r}_j) = \log\left(1 + \boldsymbol{r}_{(j)}^2\right)/2$. The different $M-$estimators are influenced by the *scale* of the residuals $\sigma_{\boldsymbol{r}}$, so a scale-invariant version based on $\tilde{\boldsymbol{r}} = \boldsymbol{r}/\sigma_{\boldsymbol{r}}$ is preferred. A robust estimation for this scale, $\hat{\sigma}_{\boldsymbol{r}}$, is the normalized median absolute deviation (MADN), which is a robust measure of dispersion:

$$\sigma_{\boldsymbol{r}} \approx \hat{\sigma}_{\boldsymbol{r}} \equiv \text{MADN} = \text{MAD}/K, \quad \text{with MAD} = \text{median}\left|r_{(i)} - \text{median}(\boldsymbol{r})\right| \quad \text{and } K = \Phi^{-1}(3/4), \tag{2.8}$$

where Φ is the cumulative distribution function of the standard normal distribution. $M-$estimators may be vulnerable to high-level leverage observations due to unusual design points, but

---

[3] This class of estimators may be regarded as a generalization of "maximum likelihood" estimation, and hence the capital M designation.

[4] This objective function must satisfy certain properties (non-negativity, symmetry, monotonicity in $|\boldsymbol{r}|$, $\rho(0) = 0$). For ordinary LS regression in the case of error terms that are i.i.d and normally distributed, then $\rho(\boldsymbol{r}) \sim \boldsymbol{r}^2$. For robust regressions, the goal is to minimise some sum of less rapidly increasing function of $\boldsymbol{r}_{(j)}$.

this effect is minimized in our case due to our use of regularly/symmetrically spaced sampling grids.
Even with these approaches, gross outliers can still have a considerable negative effect on the model. Moreover, when the number of observations N is smaller than P, LS produces an underdetermined matrix system. This is why we also wish to benefit from the robustness of the $\mathbf{L}_1-$norm type regression techniques described in the next section.

Least squares minimization with $\mathbf{L}_1-$regularization

When a function admits a sparse representation, the sparsest representation is obtained by solving this optimization problem:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\operatorname{argmin}} \|\mathbf{u}\|_0 \text{ subject to } \boldsymbol{\Psi}_{\Lambda_p}\mathbf{u} = \mathbf{y}. \tag{2.9}$$

The $\mathbf{L}_0-$norm of $\boldsymbol{u}$ is just the number of non-zero entries, it is a measure of the sparsity of $\boldsymbol{u}$. This problem, however, is a combinatorial optimization problem: one needs to go through all possible combinations of the columns of $\boldsymbol{\Psi}_{\Lambda_p}$ to find the sparsest solution which is computationally too expensive. One can approximate problem (2.9) instead by an $\mathbf{L}_1-$optimization problem called *basis pursuit*. This problem is convex and can be solved using linear programming:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\operatorname{argmin}} \|\mathbf{u}\|_1 \text{ subject to the constraint } \|\boldsymbol{y} - \boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2 = 0, \tag{2.10}$$

where $\|\boldsymbol{u}\|_1 = \sum_{j=1}^{P} |u_{(j)}|$. When the observations are noisy, the constraint is too strict and needs to be relaxed. If the magnitude of the noise is bounded: $\|\boldsymbol{e}\|_2 = \epsilon$, then we may write:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\operatorname{argmin}} \|\mathbf{u}\|_1 \text{ subject to } \|\boldsymbol{y} - \boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2 \leq \delta, \tag{2.11}$$

with $\delta \geq \epsilon$. This problem is sometimes called *basis pursuit denoising*. It is also a convex minimization problem. One can rewrite equation (2.11) as a corresponding optimization problem in Lagrangian form yielding the so-called LASSO estimate:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2^2 + \lambda\|\mathbf{u}\|_1, \tag{2.12}$$

where an appropriate $\lambda = \lambda(\boldsymbol{y}, \delta)$ is required. In practice the right value of λ depends on the realizations of the underlying random variables more than the random variables themselves. Therefore the delicate selection of this parameter is often left to cross-validation techniques in order to avoid overfitting. The systems (2.11) and (2.12) are equivalent under certain conditions [Donoho, Elad, and Temlyakov, 2006] and depending on the formulation one chooses, one of several existing solution techniques can be used to compute $\boldsymbol{u}$ [Becker, Bobin, and Candès, 2011; Efron et al., 2004; Needell and Tropp, 2009; Van Den Berg and Friedlander, 2008]. Several approaches have been recently proposed in order to enhance the efficiency of the representation resulting from solving Equation (2.11) or (2.12): – (*a priori* or iteratively) re-weighted $\mathbf{L}_1-$minimization: $\boldsymbol{u} = \operatorname{argmin}_{\mathbf{u}} \|\boldsymbol{W}\,\mathbf{u}\|_1$ where $\boldsymbol{W}$ is diagonal weight matrix, subject to $\|\boldsymbol{y} - \boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2 \leq \delta$ in order to enhance sparsity [Peng, Hampton, and Doostan, 2014; Yang and Karniadakis, 2013], – better sampling strategies minimizing the mutual coherence of $\boldsymbol{\Psi}_{\Lambda_p}$ [Hampton and Doostan, 2015b], – Bayesian compressive sensing [Sargsyan et al., 2014], or – adaptive basis selection [Jakeman, Eldred, and Sargsyan, 2015].

In our work, we will solve formulation (2.12), the Least Absolute Shrinkage and Selection Operator (LASSO), to compute the coefficients. The solution will be computed for a range of values of λ, and using cross-validation, the λ value that is best suited will be retained. The cross-validation used in LASSO is independent from the cross-validation used to determine the weight of each observation. As an alternative, one could use the result from [Fuchs, 2005] where a value for λ is computed that guarantees, under certain conditions, that the true sparse representation will be recovered.

2.2.3 Model validation

In the absence of model error $\boldsymbol{e}_M$, *truncation* and *aliasing/sampling* error are the two main potential sources of error in $\mathbf{L}_2-$based approximations. One can further distinguish internal from external aliasing errors [Conrad and Marzouk, 2013]. The former exists when the number and position of the samples do not guarantee the numerical discrete orthogonality *within* the chosen expansion basis. In practice, one can check this by verifying if one of these equalities are satisfied: $\boldsymbol{\Psi}_{\Lambda_p}^{\top}\boldsymbol{\Psi}_{\Lambda_p} = \boldsymbol{I}$ or $\mathbb{E}[\psi_{(i)}\psi_{(j)}] = \delta_{ij}$. For pseudospectral gPC approximations, it is for instance very easy to choose a (sparse) quadrature rule that insures null internal aliasing errors [Resmini, Peter, and Lucor, 2015]. Inversely, based on a given (sparse) quadrature, we know the (sparse) structure and the order $p_{\max}$ of the polynomial approximation basis we can afford.

In this case, the truncation error $\boldsymbol{e_T}$, already defined before, will remain. One way to minimize its contribution is to perform cross-validation of the stochastic approximation in order to identify the optimal approximation space frontier (e.g. $\mathbb{P}_{\Lambda_{p_{\text{opt}}}} \subseteq \mathbb{P}_{\Lambda_{p_{\max}}}$) for the functional of interest. This procedure is also appealing in the presence of model error $\boldsymbol{e}_M$ because cross-validation can reduce the sensitivity to data outliers. This is particularly true for functions that have a smooth noiseless component $\tilde{y}$. In section 2.3, we will show how we use a leave-one-out cross-validation approach as a first step for the preconditioning strategy of the LS minimization with $\mathbf{L}_1-$regularization.

Resorting to CS techniques in order to exploit potential sparsity of the QoI is interesting because it allows the exploration of a larger approximation space for the same sampling budget. It is therefore a way of reducing the truncation error of the approximation at no cost. In terms of model validation, these techniques, with their built-in property to perform basis selection, also prevent overfitting to some degree. In the LASSO formulation, cf. for instance Eq. (2.12), there is a *data fidelity* term related to the $\mathbf{L}_2-$norm and a *sparsity* term in the $\mathbf{L}_1-$norm. The LASSO evaluates the coefficients as a trade-off between these two terms thanks to the adjustment of the λ tolerance parameter. The latter may be determined again from cross-validation, e.g. [Hastie, Tibshirani, and Friedman, 2009; Blatman and Sudret, 2011]. In this chapter, we use a $K = N-$fold cross-validation in the 1D examples and $K = 10$ in the higher dimensional test cases ($K = 10$ is usually a good choice for model selection [Breiman and Spector, 1992]). There is still one more ingredient that may be added, that is the preconditioning of the data fidelity term. This may be done by assigning some weights or “trust indices" to the samples. Again cross-validation is a handy numerical tool used to evaluate the weight of each sample and this is the second step of our preconditioning strategy.

2.3 Preconditioning and weight selection

In this section we explain in detail how we derive the observation weights. We aim to assign small weights to observations in which we have a low level of confidence whilst granting a higher weight to observations which we think are reliable. Weighing the observations is a customary technique used in LS regression when one wants to filter out noise:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\operatorname{argmin}} \|\boldsymbol{W}\boldsymbol{y} - \boldsymbol{W}\boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2. \tag{2.13}$$

The solution to Eq. (2.13) can be computed as follows:

$$\boldsymbol{u} = \left(\boldsymbol{\Psi}_{\Lambda_p}^{\mathsf{T}}\mathbf{W}\boldsymbol{\Psi}_{\Lambda_p}\right)^{-1}\boldsymbol{\Psi}_{\Lambda_p}^{\mathsf{T}}\mathbf{W}\mathbf{y}, \tag{2.14}$$

where $\mathbf{W}$ is a diagonal $N \times N$ matrix containing the observation weights. It is also interesting to note that when the sample points are taken as the abscissa of an appropriate quadrature rule and one chooses the diagonal of $\mathbf{W}$ to be composed of the quadrature weights, then formulations (2.14) and (2.5) are equivalent.

Appointing weights to the observations can also be done in a compressed sensing framework, the weighted equivalent of Eq. (2.11) is:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\operatorname{argmin}} \|\mathbf{u}\|_1 \text{ subject to } \|\boldsymbol{W}\boldsymbol{y} - \boldsymbol{W}\boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2 \leq \delta. \tag{2.15}$$

Analogously, one can formulate the weighted equivalent to Eq. (2.12) as:

$$\boldsymbol{u} = \underset{\mathbf{u}\in\mathbb{R}^P}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{W}\boldsymbol{y} - \boldsymbol{W}\boldsymbol{\psi}_{\Lambda_p}\mathbf{u}\|_2^2 + \lambda\|\mathbf{u}\|_1. \tag{2.16}$$

It is the formulation above that we will be using and the weights will be computed by cross-validation as will be explained in the sections to follow.

2.3.1 Cross-validation

Leave-one-out cross-validation is a form of K-fold cross-validation with replacement where $K = N$. One constructs the surrogate model, using a method at will, with all but one of the samples. If the i^{th} sample has been left out in the construction of the surrogate model, we shall call the result $\hat{\boldsymbol{y}}_{\Lambda_p}^{(-i)}$ to indicate that this is the approximated surrogate model in Λ_p, computed without taking into account the i^{th} sample. In the framework of compressed sensing, the use of classical cross-validation has been investigated in [Ward, 2009], where results were obtained regarding the number of samples that need to be withheld for the cross-validation process to ensure an accurate representation of the error. The cross-validation method investigated there however is not the same as LOO, so while these results do not directly apply here, heuristics indicate that the LOO error yields a satisfactory estimate for the mean squared error [Molinaro, Simon, and Pfeiffer, 2005]. The LOO error is usually computed as:

$$\varepsilon_{\Lambda_p}^{\mathrm{LOO}} = \frac{1}{N}\sum_{j=1}^{N}\left(y(\boldsymbol{\xi}_{(j)}) - \hat{y}_{\Lambda_p}^{(-i)}(\boldsymbol{\xi}_{(j)})\right)^2. \tag{2.17}$$

For LS solution methods based on a random sampling strategy, Eq. (2.17) can be more efficiently computed as :

$$\varepsilon_{\Lambda_p}^{\mathrm{LOO}} = \frac{1}{N}\sum_{j=1}^{N}\left(\frac{y_j - \boldsymbol{\Psi}_{\Lambda_p,j}\,\boldsymbol{u}}{1-h_j}\right)^2, \tag{2.18}$$

where h_j is the j^{th} diagonal term in the matrix $\mathbf{\Psi}_{\Lambda_p}\left(\mathbf{\Psi}^{\mathsf{T}}_{\Lambda_p}\mathbf{\Psi}_{\Lambda_p}\right)^{-1}\mathbf{\Psi}^{\mathsf{T}}_{\Lambda_p}$, and $\mathbf{\Psi}_{\Lambda_p,j}$ is the j^{th} row of $\mathbf{\Psi}_{\Lambda_p}$. When normalized by a variance estimation, this error is called the training error. In this work we will derive this variance from the robust scale estimate introduced in equation (2.8).

2.3.2 Quadrature-based leave-one-out error estimation

Because we have intended to work with quadrature rules, we have to develop an accurate and flexible way of computing LOO errors. That is, every time a point is left out from the grid, quadrature weights of the remaining points need to be adjusted in order to insure adequate polynomial integration capability. In the following, we explain in detail how this is done in $D = 1$ and $D = 2$, the generalization to higher dimensions being straightforward.

In $D = 1$ dimension[5], let us consider a $N-$point: $\Xi_N = \{\xi_1, \ldots, \xi_N\}$ quadrature rule of polynomial accuracy $(N-1)$: $\mathcal{Q}_{N-1}[\cdot]$, and corresponding nodal weights: $\mathcal{W}_N = \{w_1, \ldots, w_N\}$.

We now require the $(N-1)-$ point: $\tilde{\Xi}^{(-i)}_{N-1} = \{\xi_1, \ldots, \xi_{i-1}, \xi_{i+1}, \ldots, \xi_N\}_{i\in\{1\ldots N\}}$ reduced quadratures (which will be missing one point relative to the original grid) to be of accuracy $(N-2)$: $\mathcal{Q}^{(-i)}_{N-2}[\cdot]$, i.e. to integrate exactly all univariate polynomials $\mathbb{P}_{\Lambda_{N-2}}$. Let us decompose a member $y \in \mathbb{P}_{\Lambda_{N-2}}$ in a specific basis: i.e. the Lagrange basis L constructed from the discrete nodal values $\tilde{\Xi}^{(-N)}_{N-1}$: i.e. we left out the last point, ξ_N, for simplicity of notation but the result holds for any other dropped point:

$$y(\xi) = \sum_{j=1}^{N-1} y(\xi_j)L_j(\xi), \tag{2.19}$$

where $L_j(\xi)$ is the Lagrange polynomial associated to ξ_j. Moving to the expectations, we have:

$$\mathbb{E}\left[y(\xi)\right] = \sum_{j=1}^{N-1} y(\xi_j)\mathbb{E}\left[L_j(\xi)\right], \tag{2.20}$$

We call the new weights $\tilde{\mathcal{W}}^{(-N)}_{N-1} = \{\tilde{w}_1, \ldots \tilde{w}_{N-1}\}$. These new weights should satisfy exact integration of y. It then follows quite naturally from Eq. (2.20) that these weights should be $\tilde{w}_i = \mathbb{E}\left[L_i(\xi)\right],\ \text{ for }\ i = 1, \ldots, N-1$, which may be evaluated in turn from the full original quadrature:

$$\begin{aligned} \tilde{w}_i &= \sum_{j=1}^{N} w_j L_i(\xi_j) = w_i L_i(\xi_i) + w_N L_i(\xi_N) \ \ (\text{because } L_i(\xi_j)|_{\substack{j\,=\,1,\ldots,N-1\\ j\neq i}} = 0) \\ &= w_i + w_N L_i(\xi_N), \ \ \text{for } \ i = 1, \ldots, N-1. \end{aligned} \tag{2.21}$$

The updated weights of the truncated quadrature are made of a summation of the weights from the full quadrature plus the Lagrange polynomial contributions evaluated at the missing node weighted by the original weight of that node. The new weights add up to unity:

---

[5] Whereas in the previous section the subscript indices without parenthesis indicated the dimension and the subscript indices with parenthesis indicated the sample number or the basis function number, here since there is no confusion possible in the one-dimensional case, the subscripts without parenthesis will directly indicate the sample number or basis function index.

$$\sum_{j=1}^{N-1}\tilde{w}_j=\sum_{j=1}^{N-1}\left(w_j+w_N L_j(\xi_N)\right)=\sum_{j=1}^{N-1}w_j+w_N\sum_{j=1}^{N-1}L_j(\xi_N)=1-w_N+w_N\times 1=1,$$

and the truncated quadrature is now valid.
The global LOO error is now evaluated on the full quadrature grid as:

$$\varepsilon^{\mathrm{LOO}}_{\Lambda_p}=\sum_{j=1}^{N}w_j\,r^2_{\Lambda_p,j},\ \text{ with }\ r_{\Lambda_p,j}=\left(y(\xi_j)-\hat{y}^{(-i)}_{\Lambda_p}(\xi_j)\right), \tag{2.22}$$

where $\hat{\boldsymbol{y}}^{(-i)}_{\Lambda_p}$ are constructed on the truncated quadratures with adjusted weights $\tilde{\mathcal{W}}^{(-i)}_{N-1}$, for $i\in\{1,\dots,N\}$.
In higher dimensions, it is common practice to rely on the assumption of independence of the random variables to construct full-grid cubatures[6], that are tensor-products of one-dimensional quadrature rules. In our case, we perform the tensorization between one-dimensional quadratures with different number of points and integration power.
In $D=2$ dimensions for instance, we build the quadratures by tensorizing the first and the second dimension. Dropping one point along the first direction results in the case of a *truncated* quadrature rule along the first dimension and a full rule along the second dimension; we form quadratures of the type: $\left(\mathcal{Q}^{(-i_1)}_{N_1-2}\otimes\mathcal{Q}_{N_2-1}\right)_{i_1=1\dots N_1}[\cdot]$, which are exact for any polynomials[7] from $\mathbb{P}_{\Lambda_{N_1-2}}\otimes\mathbb{P}_{\Lambda_{N_2-1}}$, based on a $\tilde{\Xi}^{(-i_1)}_{N_1-1}\times\tilde{\Xi}_{N_2}$ grid of $(N_1-1)N_2$ points and corresponding $\tilde{\mathcal{W}}_{N_1-1}\times\tilde{\mathcal{W}}_{N_2}$ weights. Due to symmetry of the computational grid, we can also build $\left(\mathcal{Q}_{N_1-1}\otimes\mathcal{Q}^{(-i_2)}_{N_2-2}\right)[\cdot]$, for polynomials from $\mathbb{P}_{\Lambda_{N_1-1}}\otimes\mathbb{P}_{\Lambda_{N_2-2}}$, on $\tilde{\Xi}_{N_1}\times\tilde{\Xi}^{(-i_2)}_{N_2-1}$ grid, with weights $\tilde{\mathcal{W}}_{N_1}\times\tilde{\mathcal{W}}_{N_2-1}$.
Figure 2.1 shows how we proceed to combine error estimation at a particular grid point based on those quadratures. In this example, we are interested by the first point, i.e. $(i_1,i_2)=(1,1)$. The first grid retained corresponds to the points selected by the dark blue dashed frame, once the left blue column has been dropped from the full lattice. Based on the remaining points, and once the weights have been adjusted, the updated quadrature is put to use to build a surrogate model of the QoI over the full integration domain. This allows a prediction/error estimation at any point from the shaded blue column, such as the lowest left point in red. Note that we can obtain a model error estimation at that point another way: by dropping the points in the bottom green row. Only the points in the green solid frame would then be retained to construct a surrogate model. These two different errors may be combined in several ways. After some testing, we have opted for the arithmetic mean.
The combined residual error at any particular point $r_{\Lambda_p,\boldsymbol{i}}$ is therefore taken as the mean value of the different errors produced by the ensemble of the d surrogate model designs, each built on $N=N_1\times\dots(N_{k=1\dots d}-(D-1))\dots\times N_d$ points.

For the computational setup of the quadratures, all partially truncated grids and corresponding adjusted weights combinations can be stored once and for all, for a given grid. Moreover, this step maybe by facilitated by exploiting the natural symmetry of the original multi-dimensional grid. The evaluation of the LOO errors for a given QoI on that grid are then very fast. As a side remark, we have found that estimating LOO errors from truncated Gauss-based quadratures is not significantly more efficient than estimating errors

---

[6] We will keep the quadrature nomenclature, no matter the integral dimensions.

[7] In practice, we choose $N_1=N_2\equiv\tilde{N}$ and restrain our approximation space to $\mathbb{P}_{\Lambda^{\mathrm{TD}}_{\tilde{N}-2}}$ in order to build the surrogate model.

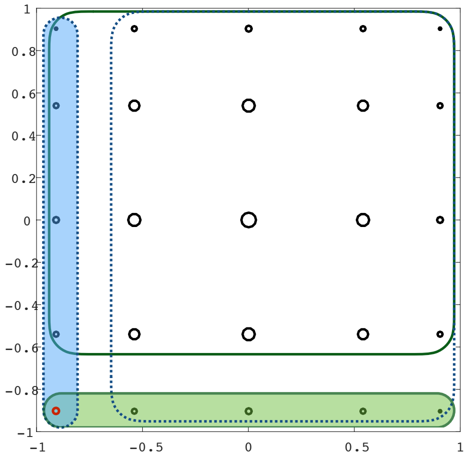


FIGURE 2.1: Schematic illustration of quadrature breaking down for leave-one-out error estimation (here for a Gauss-Legendre grid in $D = 2$ dimensions). Quadrature weight magnitudes (before adjustment) are proportional to the circle diameters.

from quadratures with a lower theoretical integration power such as the Clenshaw-Curtis rule. This is because the truncation automatically deteriorates the integration capability from $\mathbb{P}^{\mathrm{TP}}_{\Lambda_{2N-1}}$ to $\mathbb{P}^{\mathrm{TP}}_{\Lambda_{N-2}}$. In the application section, we will be using Clenshaw-Curtis (CC) or Kronrod-Patterson (KP) quadrature rules.

2.3.3 LOO-weighted preconditioned $\mathbf{L}_1-$minimization approximation

In this section, we review how the different numerical ingredients introduced previously are put together into the general approximation method we propose. There are essentially three main stages, that can be summarized as follows:

1. Selection of a quadrature rule and level, providing a $N-$point grid. Numerical simulations are then performed at these N sampling points and this grid is conserved for the rest of the method.
2. Evaluation of response sample weights that are a measure of confidence in the data obtained and will serve as a preconditioning in the next step.
3. Construction of the cross-validation preconditioned $\mathbf{L}_1-$minimization approximation using a weighted LASSO procedure in order to promote sparsity in a robust way.

The second stage requires more explanations as it is made from of several courses of action. The main idea is to take advantage of cross-validation for the estimation of prediction error in order to guide our model selection and perform robust model assessment. Here the different steps are: 2.i. to rely on the global LOO error of Eq. (2.22) to determine the most accurate polynomial approximation of the problem response, with the constraint that the aliasing error must be minimized. Then 2.ii. (this step is optional) in order to make the process more robust, not only the optimal approximation but several levels of approximation that are within a certain error threshold are retained. Finally, 2.iii and 2.iv. sample weights are computed as normalized score functions taken at the (averaged) residual error contribution of the retained approximation(s).
More specifically, here are those main steps, revisited in more detail:

2.i. Designation of optimal total degree approximation space $\mathbb{P}_{\Lambda_{p_{\rm opt}}}$ for pseudospectral gPC representation of the data: $p_{\rm opt} = \text{argmin}_{p_l \in \mathbb{N}_0^{p_{\max}}} \varepsilon^{\rm LOO}_{\Lambda_{p_l}}$, will provide the lowest truncation error in the LOO cross-validation criterion with the guarantee of no internal aliasing error. $p_{\max}$ is the highest degree authorized by the quadrature while maintaining no internal aliasing error. The different error estimations $\varepsilon^{\rm LOO}_{\Lambda_{p_l}}$ are computed from Eq. (2.22).

2.ii. (this step is optional) Choice of a model cross-validation tolerance parameter $\alpha \geq 1$. Definition of "neighbor" approximation spaces $\mathbb{P}_{\Lambda_{p\in\mathcal{L}}}$ with $\mathcal{L} = \{l \in \{1,\dots,p_{max}\} \,|\, \varepsilon^{\rm LOO}_{\Lambda_l} \leq \alpha \cdot \varepsilon^{\rm LOO}_{\Lambda_{p_{\rm opt}}}\}$ with lower errors than threshold and that will be used in the following.

2.iii. Collect the residual errors at each grid point for the retained surrogate models: $r_{\Lambda_{l\in\mathcal{L}},\, i\in\{1,\dots,N\}}$.

2.iv. Estimation of the (averaged) preconditioning weights as:

$$\mathrm{w}_{(i)} = \frac{1}{|\mathcal{L}|} \sum_{l\in\mathcal{L}} \frac{\upsilon(r_{\Lambda_l,(i)})}{r_{\Lambda_l,(i)}}, \ \ \forall i = 1,\dots,N, \tag{2.23}$$

where $|\mathcal{L}|$ is the set cardinality. This averaging is not always necessary (i.e. if $\alpha = 1$ and $l = p_{\rm opt}$) but sometimes helps in particular when the LOO error function is not clearly convex nor the choice of $p_{\rm opt}$ sharp. It is in some sense reminiscent of the damped version of the re-weighting procedure of Peng *et al.* on p.8 [Peng, Hampton, and Doostan, 2014].
In this work, Huber, Tukey bisquare and Cauchy score functions have been tested [James O. Street and Ruppert, 1988] in the numerical applications.

The third stage then consists of the weighted regression and regularization on a space of approximation of total degree larger than the one identified in step 2. i.

The algorithmic complexity and scaling of the computational framework just underlined can be split in different components. The main effort obviously lies in 1. the solution sampling at the grid points. Full cubatures scale exponentially $\mathcal{O}(N^{(D)})$ while sparse cubatures somewhat alleviate the cost $\mathcal{O}\big(N^{-r}(\log N)^{(D-1)(r+1)}\big)$, especially if the solution has high bounded mixed partial derivatives of order r and is isotropic. Then, 2. the determination of response sample weights requires cross-validation evaluations that involve multiple pseudospectral projections and arithmetic averaging. This part is computationally very efficient, even for a large number of dimensions, as long as adjusted weights necessary to the truncated cubatures have been tabulated and stored prior to the computation (cf. discussion at the end of section 2.3.2). Finally, 3. a regularized weighted regression must be carried out. The computational cost of a $\mathbf{L}_1$ LASSO-type minimization associated to the problem of Eq. (2.12) is always more expensive than ordinary or weighted LS methods for the same problem. This is due to the "search" for the best parameter λ. Nevertheless, we have noted that our LOO-weighted version sped up the computation. This is due to the preconditioning of the solution. Computational savings differ depending on the problem size and complexity. We have noted savings $5-40\%$ in computational time (savings are more substantial for larger sample points number N). Further improvements may make use of the preconditioning information in order to restrain the search range of λ.
The proposed method will now be demonstrated on several illustrative test problems of different dimensionality, sparsity and complexity.

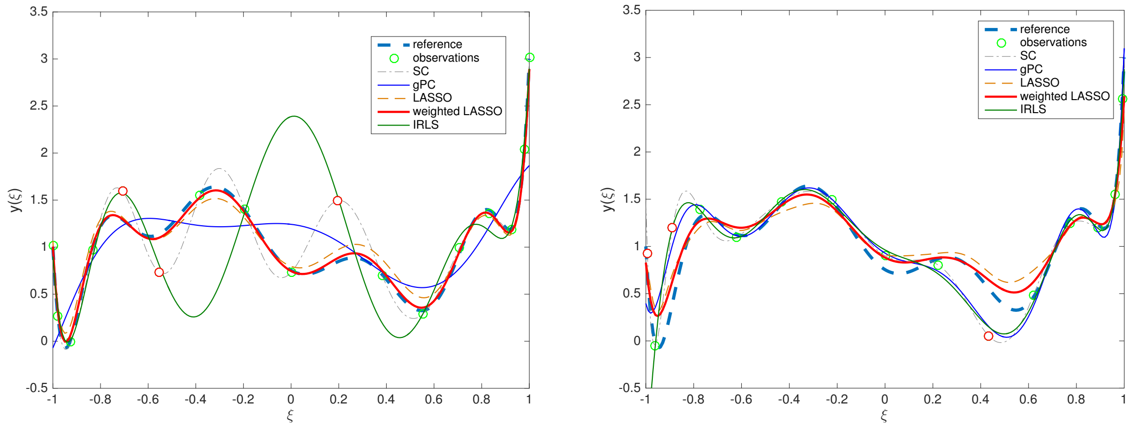


FIGURE 2.2: Continuous response surfaces of sparse Legendre polynomials obtained from different approximation methods based on a CC grid of level $l = 5$ (left) and KP $l = 4$ (right). Randomly selected outliers data are identified by red circles. Green circles represent normally distributed data samples subject to background stochastic noise. The SC curve refers the stochastic collocation based on Lagrange interpolation; IRLS is an iterative reweighted least square approximation with the same predictors as the gPC pseudospectral representation. The reference curve is the target noiseless QoI response.

2.4 Numerical Examples

2.4.1 Sparse polynomial test functions

The point of these tests is to check the robustness through an analysis of the mechanisms of the proposed method on the approximation of sparse nonlinear polynomial functionals corrupted by few randomly selected data outliers, accounting for deterministic noise. Consider the function $z(\xi) = \mathbb{P}_{10}(\xi) + \mathbb{P}_3(\xi) + \mathbb{P}_0(\xi)$, where $\mathbb{P}_k$ is the k^{th} degree univariate Legendre polynomial, known at some discrete points, in the presence of a stochastic noise component, we have: $y_i = z(\xi_i) + \chi_i$, with $i \in \{1, \ldots, N\}$ and χ_i are centered i.i.d. random variables distributed according to $\mathcal{N}(0, \sigma_\chi)$. These data also contain some outliers that do not match this definition. In practice we have considered σ_χ values about one order of magnitude lower than the variability scale associated to the outliers. The noiseless version of this functional has been previously tested with iterative adaptive polynomial approximations [Poëtte, Birolleau, and Lucor, 2015]. Here, the random variable $\xi \sim \mathcal{U}_{[-1,1]}$ is *uniformly* distributed. Continuous approximations will be constructed from discrete sampling on regular grids. Without loss of generality, we will be presenting 1. a CC quadrature rule of level $l = 5$ (17 points) and 2. a KP quadrature rule of level $l = 4$ (15 points). Finer grids have also been tested with success.

For case 1., the function we try to approximate by projection is of maximum order 10 which is out reach for the polynomial integration capability of our grid. Standard pseudospectral methods are not able to capture the correct solution in this case, but the function being sparse (only 3 active basis functions are needed), we expect that the $\mathbf{L}_1$-regularization term will help in approaching the right solution. As stated before, we will be using LASSO in order to solve Eq. (2.12), but our proposed technique for weighting the observations can also be used in combination with other solution methods. The results are presented in Figure 2.2-(left). The outliers data points are plotted as red open circles, the other points as green open circles. The number of outliers are arbitrarily chosen and affect $N_o = \kappa(\%) \times N$ samples (with $\kappa \approx 18\%$), while $\chi_i \sim \mathcal{N}(0, 4 \cdot 10^{-2})$. The chosen example is tricky as

Grid	Approximation	p^{TD}	μ	σ^2	R^2	$\mathbf{L}_1$	$\mathbf{L}_\infty$
$\mathrm{CC}_{l=5}$	gPC	8	$4.04{\cdot}10^{-2}$	$3.97{\cdot}10^{-1}$	$2.90{\cdot}10^{-1}$	5.70	1.13
	LASSO	16	$2.82{\cdot}10^{-2}$	$3.25{\cdot}10^{-1}$	$9.74{\cdot}10^{-1}$	1.53	$1.61{\cdot}10^{-1}$
	weighted$_{\alpha=1}$-LASSO	(3) 16	$\mathbf{3.99{\cdot}10^{-4}}$	$\mathbf{1.23{\cdot}10^{-1}}$	$\mathbf{9.96{\cdot}10^{-1}}$	$\mathbf{6.06{\cdot}10^{-1}}$	$1.08{\cdot}10^{-1}$
	weighted$_{\alpha=1.35}$-LASSO	(3) 16	$1.03{\cdot}10^{-2}$	$1.61{\cdot}10^{-1}$	$9.92{\cdot}10^{-1}$	$8.51{\cdot}10^{-1}$	$\mathbf{9.32{\cdot}10^{-2}}$
	IRLS	8	$5.04{\cdot}10^{-2}$	1.53	$5.01{\cdot}10^{-1}$	5.69	1.63
$\mathrm{CC}_{l=5}$	gPC	8	$5.08{\cdot}10^{-2}$	$4.05{\cdot}10^{-1}$	$2.93{\cdot}10^{-1}$	5.70	1.12
	LASSO	16	$4.16{\cdot}10^{-2}$	$3.42{\cdot}10^{-1}$	$9.68{\cdot}10^{-1}$	1.64	$1.93{\cdot}10^{-1}$
	weighted$_{\alpha=1}$-LASSO	(3) 16	$\mathbf{1.92{\cdot}10^{-2}}$	$\mathbf{1.61{\cdot}10^{-1}}$	$\mathbf{9.94{\cdot}10^{-1}}$	$\mathbf{7.50{\cdot}10^{-1}}$	$\mathbf{8.53{\cdot}10^{-2}}$
	weighted$_{\alpha=1.35}$-LASSO	(3) 16	$2.47{\cdot}10^{-2}$	$1.84{\cdot}10^{-1}$	$9.92{\cdot}10^{-1}$	$8.88{\cdot}10^{-1}$	$1.04{\cdot}10^{-1}$
	IRLS	8	$9.49{\cdot}10^{-2}$	$7.20{\cdot}10^{-1}$	$5.69{\cdot}10^{-1}$	5.16	1.41

TABLE 2.1: In reference to the results of Figure 2.2-(left): overview of different functional error indicators for a sparse polynomial test case with (top) and without (bottom) stochastic background noise and for different choices of α. The best overall result in bold. $p^{(TD)}$ is the chosen total degree of the polynomial approximation basis; for the weighted-LASSO approach, the value in parenthesis is the optimal order obtained from the original pseudospectral gPC representation resulting in the lowest overall cross-validation LOO error.

the outliers are placed within the $[\min_{\xi\in[-1,1]} y(\xi), \max_{\xi\in[-1,1]} y(\xi)]$ range. The reference noiseless curve is depicted as a full dashed light blue line. The three solutions that are clearly off the marks are the standard gPC (full blue line), the IRLS (full green line) and the stochastic collocation (thin dotted-dashed gray line) which are also shown for sake of completeness and exhibit too little or too large oscillations. LASSO solution (thin purple dashed line) performs better but not as good as the LOO-weighted LASSO. It is clear that the preconditioned $\mathbf{L}_1-$regularized approximations perform best. These qualitative observations are quantitatively confirmed in Table 2.1. Table 2.1 shows the errors in the $\mathbf{L}_1$, $\mathbf{L}_\infty$ norm and the R^2 (*goodness of fit*) and the errors in the global statistical moments. Results with stochastic noise-free (green) samples but bearing the same outlying (red) cases are also included. In Figure 2.3 some of the internal workings of the proposed technique are exposed. Subplot (a) shows how the cross-validation with LOO technique clearly predicts, despite the noise, that polynomial approximation of total degree $p = 3$ will minimize prediction errors within the range of affordable polynomial orders. This is coherent with the 0^{th}- and 3^{rd}-order components present in the functional. Error estimation solely based on domain integrated gPC residuals are lower as expected but less robust and the optimal polynomial order choice within $\{3, \dots, 8\}$ is therefore less obvious. Subplot (c) shows the weights assigned to the samples for a Cauchy score function. As expected, levels of confidence are lower for data outliers (represented by red circles). They are also low for the boundary samples that are negatively affected due to their distance to the low-order approximation. Last two subplots show the subtle differences in the mean square errors (MSE) distribution vs. λ for the LASSO (d) and the weighted-LASSO (e). Very low values of λ, to the right of these plots, lead to approximations dominated by the first term of Eq. (2.16). Despite the preconditioning, the $\mathbf{L}_2$ minimization alone produces larger errors with large error bars. Once the optimal λ selected, very low MSE errors are obtained and coefficients amplitude in subplot (b) shows that the three leading modes of the functional, including the 10^{th}-order, are almost perfectly captured, despite some weak spurious energy in the 5^{th} and 12^{th} modes.

The next test case has a similar setup but a larger noise, e.g. $\chi_i \sim \mathcal{N}(0, 7 \cdot 10^{-2})$ and consequently more severe data outliers, on a different sampling grid. For case 2., the

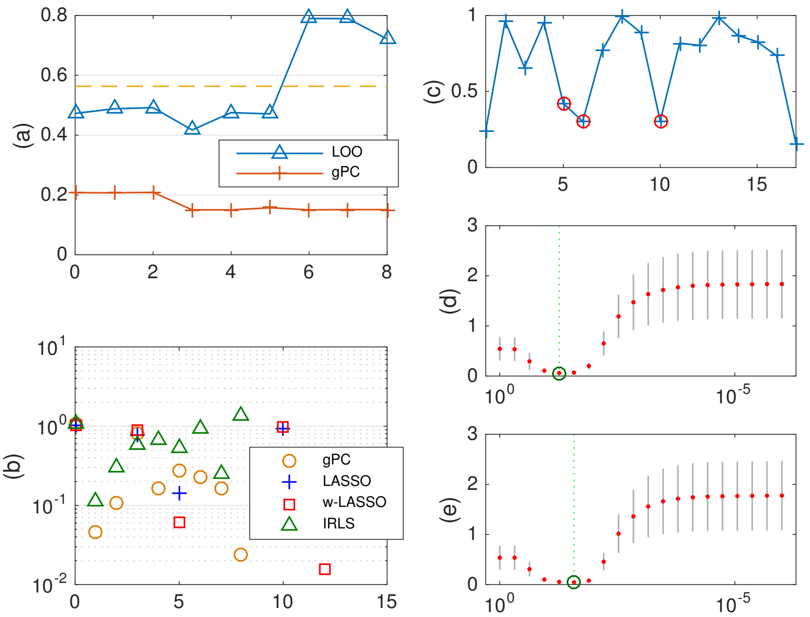


FIGURE 2.3: In reference to the results of Figure 2.2-(left): (a) overview of the model errors vs. polynomial total degree p, (b) polynomial coefficients magnitude u_j^2 vs. p, (c) sample weights and finally model cross-validated mean square errors vs. λ for LASSO (d) and for weighted-LASSO (e).

Grid	Approximation	p^{TD}	μ	σ^2	R^2	$\mathbf{L}_1$	$\mathbf{L}_\infty$
KP$_{l=4}$	gPC	10	$\mathbf{1.48{\cdot}10^{-3}}$	$\mathbf{2.10{\cdot}10^{-1}}$	$8.47{\cdot}10^{-1}$	2.58	$6.18{\cdot}10^{-1}$
	LASSO	20	$5.62{\cdot}10^{-2}$	$5.57{\cdot}10^{-1}$	$7.43{\cdot}10^{-1}$	2.39	$5.25{\cdot}10^{-1}$
	weighted-LASSO	(10) 20	$4.48{\cdot}10^{-2}$	$3.58{\cdot}10^{-1}$	$\mathbf{9.17{\cdot}10^{-1}}$	**1.56**	$\mathbf{3.07{\cdot}10^{-1}}$
	IRLS	10	$1.92{\cdot}10^{-2}$	$4.27{\cdot}10^{-1}$	$5.89{\cdot}10^{-1}$	3.82	1.80

TABLE 2.2: Same caption as in Table 2.1, but in reference to the results of Figure 2.2-(right).

KP quadrature/grid combination has a higher integration capability than the previous grid. This time, results presented in Figure 2.2-(right) are not visually as impressive, but weighted-LASSO still performs best in most of the error norms, cf. Table 2.2.

Additional results collected in Figure 2.4 better point to some of the differences with the previous case. This time, the cross-validation is able to predict that polynomial of order 10 is also crucial to the approximation. Data outliers are then endowed with low confidence but a few other data samples are misleadingly granted with low weights as well (c). The LASSO error distribution plots (d-e) show that the error levels remain low even for very small values of λ. In this case, the $\mathbf{L}_1-$regularization term does not contribute significantly in terms of the accuracy improvement. However, the preconditioning still helps the LASSO algorithm in better finding the optimal λ value. Looking at the emerging modal coefficients in (b), we notice again that despite its better results, weighted-LASSO is not as sparse as the standard LASSO approximation.

Other one-dimensional tests were pursued in the same spirit, for non-sparse non-polynomial functions. For instance, results for a noisy data set obtained from $z(\xi) = (-3\xi^5 + \xi^2 + \xi) \times \tanh(\xi)$ and corrupted by four data outliers (results not presented here) confirmed the performance advantage of weighted-LASSO with respect to LASSO and standard gPC.

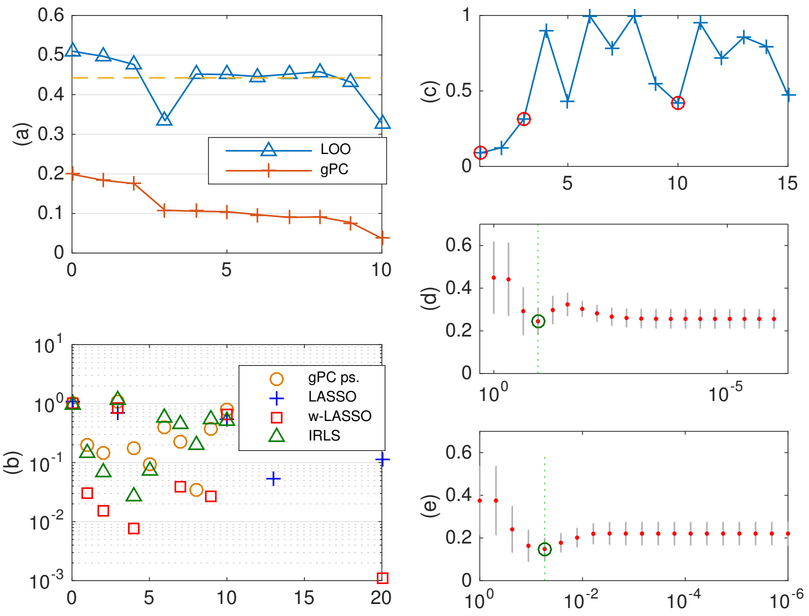


FIGURE 2.4: Same caption as in Figure 2.3, but in reference to the results of Figure 2.2-(right).

2.4.2 Higher-dimensional non-polynomial test function

Now, we consider a higher-dimensional test function that is not necessarily compressible so that we do not favor $\mathbf{L}_1-$type regression method over robust iterative weighted least-square approximations. We assess the continuous approximation of an algebraic noisy version of the Genz corner-peak function, known at some discrete locations, which provides a flexible test for the proposed method:

$$y_{(i)} = \left(1 + \sum_{k=1}^{D} c_k \xi_{k,(i)}\right)^{-(D+1)} + \chi_{(i)}, \text{ with } i = 1, \ldots, N, \tag{2.24}$$

and $\chi_{(i)}$ are centered i.i.d. random variables distributed according to $\mathcal{N}(0, \sigma_\chi)$ and $\boldsymbol{\xi} \equiv (\xi_1, \ldots, \xi_D) \sim \mathcal{U}_{[0,1]^D}$. Specifically, the coefficients c_k can be used to control the effective dimensionality and the compressibility of this function. In the following, we first test the $d = 3-$dimensional version using the anisotropic coefficients $c_k = 1/k^2$ defined in [Jakeman, Eldred, and Sargsyan, 2015]. The function is computed on a 7^3 KP grid but similar tests have been performed for finer grids as well as for grids of different nature without affecting the overall conclusion. The outlying cases amount to $N_o = \kappa(\%) \times N$ randomly selected samples in the domain. In practice, the outlier locations are randomly distributed in the computational domain with a uniform distribution. For these high-dimensional cases, their magnitude is automatically drawn from either: - a non-normal distribution or - a normal distribution with a standard deviation of one order of magnitude larger than σ_χ. The latter definition has been used for the results presented next. Moreover, the procedure has been repeated 500 times (i.e. with different initial conditions both for outlier locations and magnitudes and for the stochastic noise). Statistical results are presented in Figure 2.5. They show that the standard gPC approximation is not robust. LASSO improves the error statistics, in particular in the $\mathbf{L}_1$ and $\mathbf{L}_\infty$ norms. Mean values extracted from these distributions and reported in Table 2.3 confirm these findings.

One-shot LOO-weighted LASSO improves the results even further, coming close to the

Grid	Approximation	p^{TD}	$\overline{\mu}$	$\overline{\sigma^2}$	$\overline{R^2}$	$\overline{\mathbf{L}_1}$	$\overline{\mathbf{L}_\infty}$
$\text{KP}_{l=3}$	gPC	5	$1.46e{-}02$	$2.07e{-}01$	$8.33e{-}01$	$1.75e{+}01$	$3.10e{-}01$
	LASSO	9	$1.14e{-}02$	$1.52e{-}01$	$9.60e{-}01$	$7.24e{+}00$	$9.49e{-}02$
	weighted-LASSO	(1.2) 9	$6.82e{-}03$	$6.55e{-}02$	$9.89e{-}01$	$\mathbf{3.88e{+}00}$	$5.38e{-}02$
	IRLS	5	$\mathbf{3.67e{-}03}$	$\mathbf{3.21e{-}02}$	$\mathbf{9.91e{-}01}$	$4.05e{+}00$	$\mathbf{5.05e{-}02}$

TABLE 2.3: Similar caption as Table 2.1 for $D = 3$ dimensions Genz corner-peak functional. But this time, all errors are averaged as the test cases have been repeated 500 times for different initial conditions.

IRLS iterative scheme.
A single test following a similar setup is carried out for $d = 5$ dimensions. We recall that this dimensional limitation closely connected to memory requirement is inherited from the scaling of full cubature sampling grid, but would be alleviated for a sparse cubature. The error results summarized in Table 2.4 demonstrate again that LOO-weighted LASSO and IRLS are the best two contenders for robustness and that our approach performs well compared to the iterative scheme.

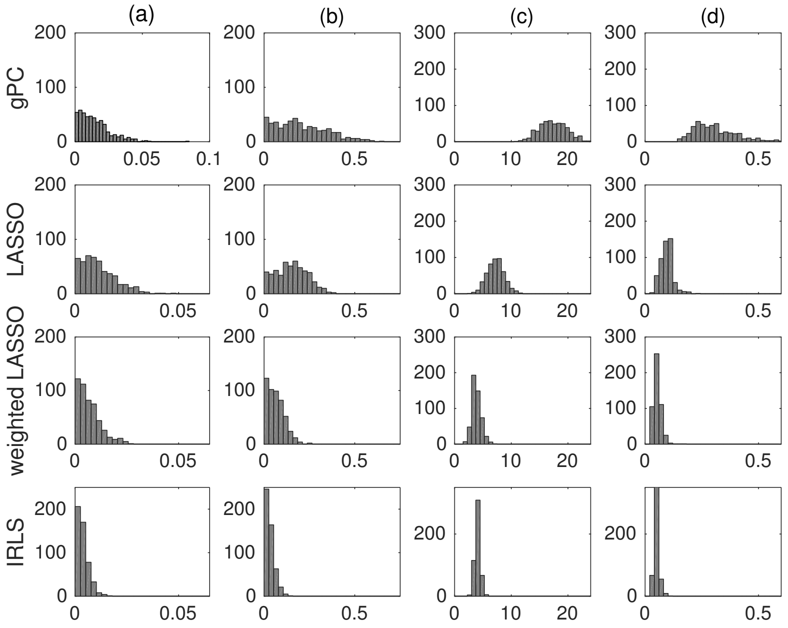


FIGURE 2.5: Comparison of different approximation errors: i.e. μ (a), σ^2 (b), $\mathbf{L}_1$ (c) and $\mathbf{L}_\infty$ (d), of a three-dimensional non-polynomial noisy Genz corner-peak function obtained from different methods relative to the noise-free reference solution. All approximations are based on 7^3 KP sampling grid. Data outliers affect $\kappa = 15\%$ of the total number of samples and $\sigma_\chi = 0.03$. The test is repeated 500 times for different random initial conditions and stochastic noise.

2.4.3 2D compressible, inviscid flow test cases

We rely on an compressible Euler equations solver named `Wolf` [Alauzet and Loseille, 2016b]. It is based on a compressible mixed-element-volume method on unstructured grids and is meant to deal with highly anisotropic meshes. For our computations we used a second-order HLLC approximate Riemann solver, MUSCL-type scheme with limiters and an implicit matrix free method for time advancing. The adjoint capabilities of

Grid	Approximation	p^{TD}	μ	σ^2	R^2	$\mathbf{L}_1$	$\mathbf{L}_\infty$
KP$_{l=3}$	gPC	5	$2.70{\cdot}10^{-3}$	$2.88{\cdot}10^{-2}$	$9.71{\cdot}10^{-1}$	$3.71{\cdot}10^{2}$	$2.70{\cdot}10^{-1}$
	LASSO	9	$3.38{\cdot}10^{-3}$	$1.93{\cdot}10^{-2}$	$9.98{\cdot}10^{-1}$	$8.29{\cdot}10^{1}$	$4.44{\cdot}10^{-2}$
	weighted-LASSO	(2) 9	$\mathbf{1.28{\cdot}10^{-3}}$	$\mathbf{5.20{\cdot}10^{-3}}$	$\mathbf{9.99{\cdot}10^{-1}}$	$\mathbf{5.37{\cdot}10^{1}}$	$2.79{\cdot}10^{-2}$
	IRLS	5	$3.35{\cdot}10^{-3}$	$1.04{\cdot}10^{-2}$	$9.99{\cdot}10^{-1}$	$6.06{\cdot}10^{1}$	$\mathbf{1.91{\cdot}10^{-2}}$

TABLE 2.4: Same caption as Table 2.1 for $D = 5$ dimensions Genz corner-peak test case.

the solver is also exploited to drive goal-oriented mesh adaptivity as it will be specified hereafter.

NACA0012 airfoil

The study of compressible flows around a NACA0012 airfoil are considered in this example. The functional of interest is the stagnation pressure P_a integrated along the airfoil profile Γ :

$$P_a = \frac{1}{L(\Gamma)p_{a_\infty}} \int_\Gamma p_a \mathrm{d}\Gamma, \tag{2.25}$$

where $p_a = p\left(1 + \frac{\gamma-1}{2}M_\infty^2\right)^{\frac{\gamma}{\gamma-1}}$ (respectively p_{a_∞}) is the (upstream) stagnation pressure, γ is the specific heat ratio, here fixed at 1.4 and the free-stream Mach number $M_\infty = 0.5$, $L(\Gamma)$ denotes the length of the airfoil. We have considered for this analysis one uniformly distributed uncertain parameter: the angle of attack $\mathrm{AoA} \equiv \xi \sim \mathcal{U}_{[0;8^\circ]}$. It is well known that for subcritical Euler flows at zero or moderate angle of incidence, the stagnation pressure should be exactly equal to unity [Peter, Nguyen-Dinh, and Trontin, 2012]. In practice, even for low Mach numbers this exact value is unreachable due to numerical (e.g. discretization) errors. Moreover, for larger angles of incidence (i.e. $\mathrm{AoA} \gtrapprox 6$), the loss of symmetry is such that the flow switches from the subsonic to a transonic regime, with a shock appearing on the upper surface close to the leading edge. Figure 2.6 illustrates this phenomenon with three snapshots of the density field for $\mathrm{AoA} = 0$ (left image), $\mathrm{AoA} = 6$ (middle image) and $\mathrm{AoA} = 8$ (right image), respectively. The rise of this shock noticeably modifies the flow features and negatively affects accuracy of the prediction if no adjustment is made to the model in order to account for it. It especially impacts the discretization error if the retained mesh is too coarse and not adapted at the shock location. In this case the model error magnitude will depend on the value of the AoA in quite an unpredictable manner as can be seen in the stagnation pressure response pictured in Figure 2.7 (red circles). The response was obtained for each AoA from the same coarse mesh: with about five thousand mesh cells regularly distributed around the airfoil and referred as the uniform $5K$ mesh. We notice strong oscillations in the transcritical region for AoA larger than about six degrees. Consequently, high-order pseudospectral gPC approximation (dotted blue curve) is corrupted with errors as expected. Cross-validation preconditioned regularized approximation (solid red curve) does much better at filtering out small spurious fluctuations in the left region where P_a is not dependent on the AoA, as well as controlling and erasing large unphysical P_a oscillations on the right hand side of the domain.

Interestingly, for this problem, it is possible and still affordable to produce results that are almost model error-free. By refining the mesh to 38000 mesh cells ($38K$), the discretization error is drastically reduced. These refined meshes are adapted to *each* AoA scenario in order to capture the critical physics (e.g. shocks). Computations on the refined and adapted meshes are represented by the green stars. We observe a flat zone corresponding to low

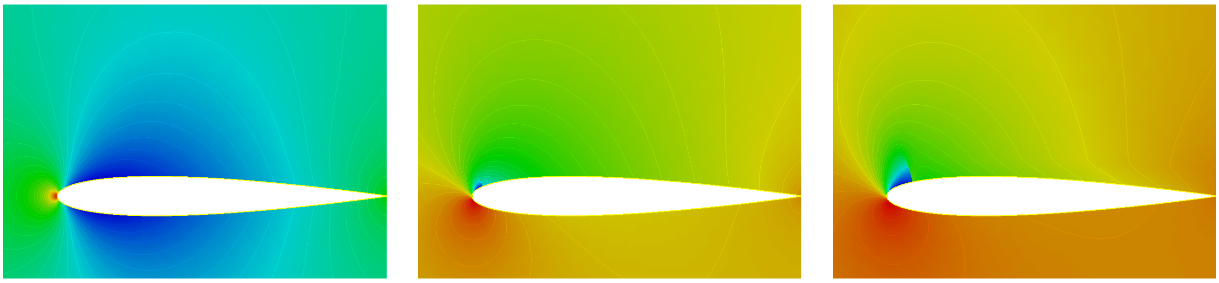


FIGURE 2.6: NACA0012: density field closeup for different angles of attack: AoA $= 0$ (left), AoA ≈ 6 (middle) and AoA $= 8$ (right). Note the presence of small shocks close to the leading edge at large angles of attack. Computational meshes are not displayed but have been adapted and refined to capture all relevant flow features.

angles of attack where the stagnation pressure is very close to, but lower than unity (due to still present numerical diffusion), followed by a sharp almost linear decrease for larger angles of attack. In this case, our method does not alter the data and produces a smooth response (solid brown curve) while perfectly maintaining the right slope.

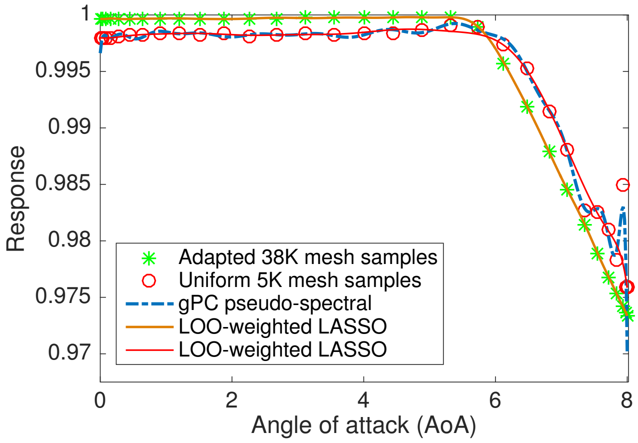


FIGURE 2.7: NACA0012: averaged stagnation pressure P_a response surfaces vs. AoA, obtained from different approximation methods based on a level $l = 5$ Kronrod-Patterson data sampling. Two classes of discretization meshes of the Euler flow are investigated: – uniform coarse mesh (red circles) vs. – fine mesh adapted to each flow incidence (green stars).

If we now consider that the Mach number is also uncertain: for instance, $M_\infty \sim \mathcal{U}_{[0.3,0.5]}$, the stagnation pressure value departs from unity at a critical angle that depends on the Mach number; this angle being larger for lower Mach numbers. This induces a narrow region with a steep slope that is difficult to capture accurately by standard projection techniques and induces spurious oscillations, cf. Figures 2.8 and 2.9. Again the proposed method increases the regularity of the surrogate where it is needed, with no *a priori* information nor significant computational overhead, while capturing relevant local sharp features even on the coarser mesh. We notice in particular that the surface goes through the data samples much better in the transcritical region.

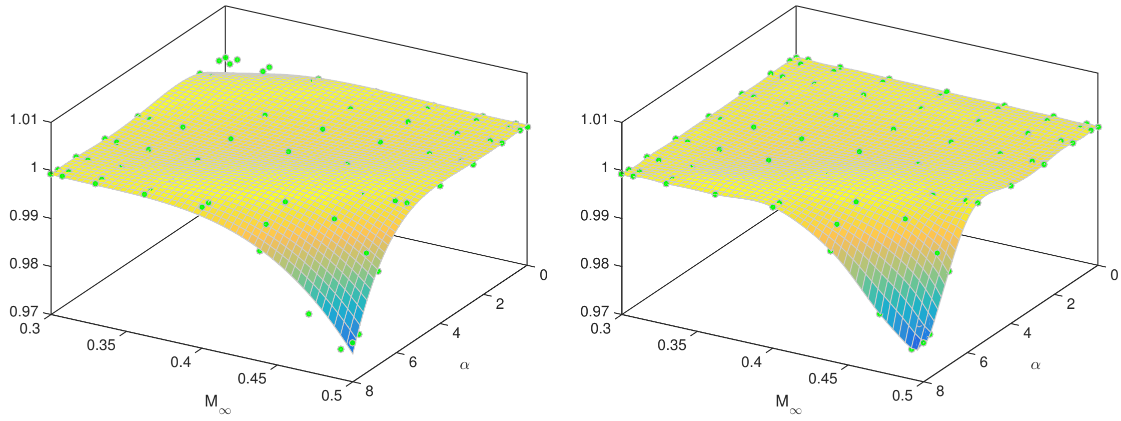


FIGURE 2.8: NACA0012: averaged stagnation pressure P_a response surfaces vs. AoA and M_∞, based on a 9^2 Clenshaw-Curtis data sampling: – pseudospectral gPC expansion with $p = 4$ (left) and – LOO-weighted LASSO (right). Each deterministic CFD simulation is performed on a non-adapted mesh with ~ 7000 cells.

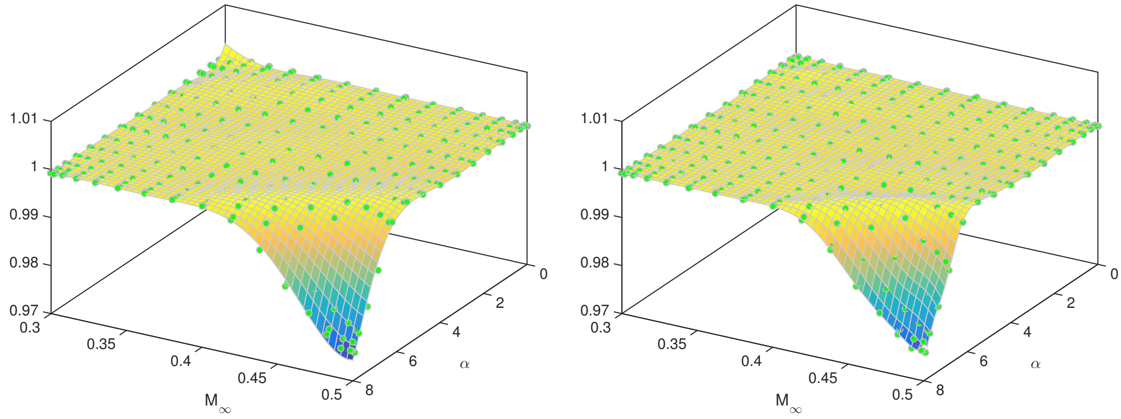


FIGURE 2.9: Same caption as the previous figure but based on a 17^2 Clenshaw-Curtis data sampling: – pseudospectral gPC expansion with $p = 8$ (left) and – LOO-weighted LASSO (right).

Scramjet engine internal flow

Major progress has been achieved over the past decades in CFD capabilities to provide engineers with powerful tools for analysis of complex flows, such as high-speed compressible flows. This has been motivated in part by the growing interest in the hypersonic flight regime for which the engine of choice is the supersonic combustion ramjet (scramjet), for atmospheric propulsion of both hypersonic aircraft and missiles. The scramjet engine is fundamentally simple in concept – it was proposed in the 1950s [Curran, 2001] – but difficult in realization as current simulation capability of in-flight performance is overwhelmed by numerous uncertainties (e.g. natural variability of flight scenario, effect of geometrical variability and manufacturing tolerances, fuel conditions and combustion kinetics,...) and errors due to the multi-physics nature of the problem [Witteveen, Duraisamy, and Iaccarino, 2011]. Here we will be interested in the internal flow through a scramjet inlet [Alauzet and Loseille, 2016b] at moderate to high Mach numbers. The geometry is shown in Figure 2.10. While its configuration with sharp angles induces a solution with numerous shock waves (shock-train), as air compression occurs across shocks

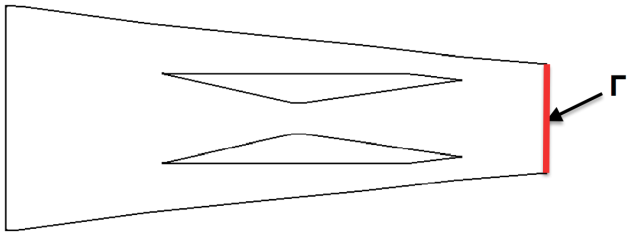


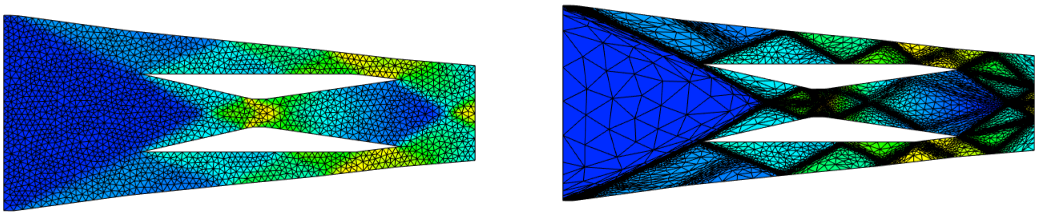
FIGURE 2.10: Scramjet geometry; the QoI is integrated on Γ.

(fluidic compressor), the position of the shock-train directly affect combustion and engine performance. The shock's number, orientation and amplitude is strongly dependent on the free stream Mach number and the angle of attack of the incoming flow that we consider here incertain and independent with bounded distributions. Another challenge of hypersonic flows is the simulation of the strong interaction between those shocks and the turbulent boundary layers. In this preliminary work, we neglect viscosity and do not address this issue.
We are interested by the effect of uncertain parameters on the outlet flow pressure – i.e. at the downstream boundary condition Γ, cf. 2.11 – just upstream of the location where the fuel is mixed with air and combust in the combustor chamber of the full scramjet configuration. More specifically, we consider the following QoI :

$$j(w) = \int_{\Gamma} \left(\frac{p - p_{\infty}}{p_{\infty}} \right)^2 dy, \tag{2.26}$$

that is integrated along the outlet edge. This choice is also motivated from the numerical point of view. Indeed, it has been shown that advanced mesh adaptivity methods are needed to capture intricate shocks and contact discontinuities developing at the rear boundary [Alauzet and Loseille, 2016b]. It is interesting to analyze how this behavior evolves when the operating condition changes, and thus the impact it has on our QoI.
The Mach number and the angle of attack are uniformly distributed with $M_{\infty} \sim \mathcal{U}_{[2.4,4]}$ and $\alpha \sim \mathcal{U}_{[0,6]}$ degrees respectively. The Mach number lower bound is purposely chosen to be within the subsonic regime in order to make the case more challenging. The results show that the sensitivity of the pressure distribution is mainly to the Mach number. Circle markers in the graphs of 2.12 show the QoI predicted from the solver sampling at the nodes of a CC grid of level $l = 5$ (i.e. $N = 17^2$). We notice that for an intermediate range of Mach numbers the QoI sharply increases due to the transition to supersonic regime. For higher Mach numbers, the integrated quantity decreases again due to the specific combination of shocks distribution at the outlet. Moreover, for angles of attack > 4 degrees, the supersonic transition at the entrance of the inlet geometry is slightly delayed to higher Mach numbers. This has implication on the outlet shocks distribution as well and induces a shift in the maximum QoI response to higher Mach numbers. The stochastic response surface that we have to approximate from the solver discrete samples is therefore extremely challenging without any sampling nor reconstruction adaptation.
We present the results of different approximation methods based on Legendre polynomial chaos in 2.12. Reference results obtained from standard pseudospectral projection on a space of approximation satisfying orthogonality constraints: $\mathbb{P}_{\Lambda_{p=8}}$ (a) does not provide satisfactory results and exhibit strong oscillations. The results obtained with the Lasso formulation on $\mathbb{P}_{\Lambda_{p=15}}$ show some improvements and the regularization somewhat dampens the oscillations. Thanks to the sample weights adjustment, the proposed LOO-weighted

LASSO formulation provides, on the same approximation space, the smoothest and most accurate approximation, as the surface response closely approaches the data while avoiding excessive overfitting. Nevertheless, we notice that the approximation most likely still under-predicts the maximum response. This is because the formulation has difficulty discerning large and sudden amplitude peak variation from model noise. In order to be more confident in the samples corresponding to the maximum response, a new set made of refined simulations (i.e. 17 simulations at $M_\infty = 2.89$ and various angles of attack, illustrated by the red circle markers in 2.12-(d)) is generated and added to the data set. For each of these new simulations, we rely on an anisotropic adapted mesh where the refinement criteria is based on an *a priori* estimation of the discretization error of the QoI [Loseille, Dervieux, and Alauzet, 2010]. A metric field is then computed for each mesh node which provide optimal directions and sizes for remeshing. The optimal adaptive mesh is generated for a given target number of mesh nodes and these steps are iteratively repeated (in practice 4 or 5 times) until mesh convergence. An example of such an adapted mesh is presented in 2.11-(b); we notice that flow features are much better captured, which improves considerably the accuracy of the QoI prediction. In our case, the peak integrated pressure values increase relative to their reference values. Consequently, the results of the refined simulations are now better trusted and maximum weight values of unity are then assigned to each sample. The LOO-weighted LASSO formulation is then repeated with this new constraint. The results of 2.12-(d)) show that the approximation directly adjusts to the new weight distribution and includes the new samples, while controlling the growth of excessive spurious oscillations.

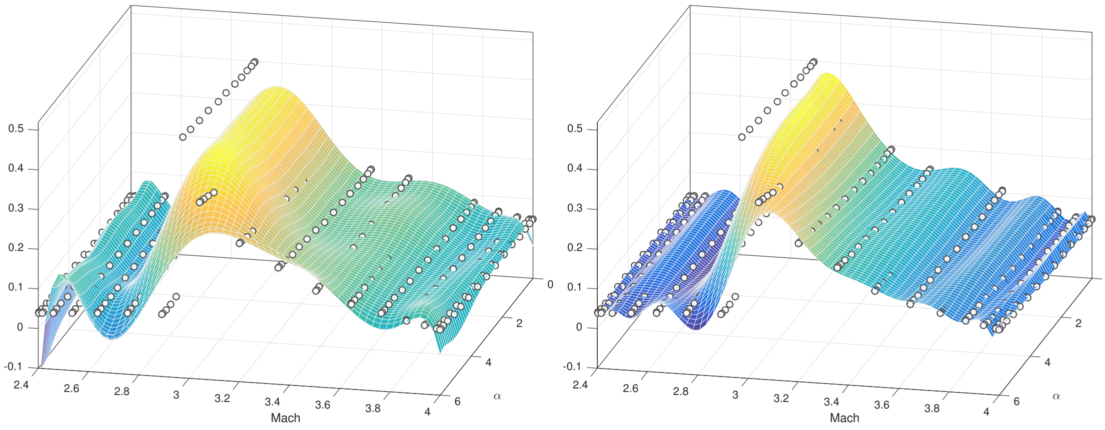


(A) Uniform mesh. (B) Adapted mesh.

FIGURE 2.11: Distributions of iso-Mach contours for an inviscid Mach number $M_\infty = 3$ scramjet inlet at zero angle of attack; (a): uniform coarse mesh (3818 elements), (b) adapted fine mesh (11707 elements).

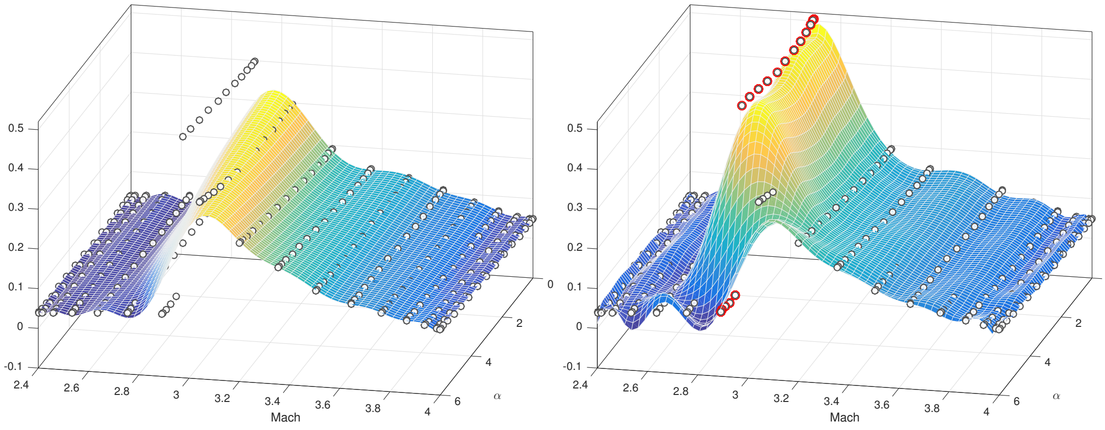
2.5 Conclusions

The main contribution of this chapter was to propose a non-iterative robust numerical method for the uncertainty quantification of reasonably compressible multivariate stochastic solutions. The goal was to make the approximation capable of dampening the effect of outlying data to that do not fit the assumption of small additive stochastic noise represented by centered i.i.d. (normal) random variables with uniformly bounded variance; in particular, noise which does not fall under the regularity assumption of the stochastic truncation error but pertains to a more complete error model. The method required a preconditioning prior to a dimension reduction of the solution, i.e.: 1. a $\mathbf{L}_2$−based cross-validation of a generalized Polynomial Chaos approximation of the response; this allowed a first model selection and the computation of (preconditioning) weights (i.e. confidence measures) associated to the samples, followed by 2. a preconditioned least-squares polynomial approximation with regularization using the weighted Least Absolute Shrinkage and Selection Operator. For the first step, observation weights were computed from



(a) pesudospectral

(b) LASSO



(c) weighted-LASSO

(d) weigthed-LASSO*

FIGURE 2.12: Scramjet integrated outlet pressure response surfaces vs. M_∞ and α, based on a 17^2 CC data sampling: pseudospectral gPC expansion with $p = 8$ (a), LASSO (b) and LOO-weighted LASSO (c-d) with $p = 15$. Most Euler deterministic CFD simulations (white circles) are performed on a non-adapted mesh (cf. 2.11-(a)), except for the simulations* (red circles) carried out on QoI-oriented adapted meshes (example of such a mesh in 2.11-(b)).

sample contributions to the cross-validation leave-one-out error of the selected surrogate model. For the second step, other algorithms may be used to solve the optimization problem resulting from the $\mathbf{L}_1-$regularization. Numerical test cases treated above have proved the numerical method to be more effective in automatically canceling out or reducing the influence of data outliers than standard compressed sensing techniques and of comparable efficiency to iterative robust regression techniques.

A particularity of this work was to make use of quadrature rules/grids as opposed to random sampling. This zero-variability sampling brings reliability to the recovery procedure but is better suited for low to moderate dimensional problems (with possibly high-order representation). However, the approach remains general and could be applied to higher dimensions by using random sampling or quadrature subsampling schemes taking advantage of recent advances in terms of polynomial recovery optimization. In this case, the use of other cross-validation techniques with potentially lower variance error estimations is also conceivable.

Chapter 3

Riemannian metric framework for anisotropic adaptation

Over the past decades, developments in numerical methods for solving partial differential equation together with the advancements in computational power made the solving of engineering problems by use of simulations ever more practical. Complex engineering problems formulated on sophisticated geometries are nowadays reachable, even for realistic unsteady problems. This is partially due to progress made in properly defining the meshes associated to these complex problems. A mesh can be defined as *optimal* if it captures the physics of the problem to within a prescribed accuracy at an acceptable cost. Such optimal meshes are often obtained with an adaptive process and the resulting mesh will be called the adapted mesh. An indicator will be needed to identify mesh regions that need refinement or coarsening. Error estimators have been proven be a valuable such indicator.

If the exact solution is given by u and the approximated solution by u_h, then it would make sense to use the error $||u - u_h||$ as the refinement indicator. Unfortunately this quantity cannot be computed since u is not known. In order to overcome this problem, computable error estimators are used to estimate this error. One can distinguish two classes of error estimators: *a posteriori* and *a priori* error estimators. The former are based on the truncation error while the latter quantify how well u verifies the (approximate) model. Another important aspect is how, once the error indicator computed, this information is used to generate a refined mesh. The simplest method is to divide each mesh element where the error estimate is above a certain limit into smaller elements of the same shape. This is called isotropic refinement. While this refinement method will lead to smaller elements and a reduction in the discretization error in regions where the error is estimated to be large, it has the disadvantage of lacking the capacity to capture the anisotropic nature of physics. When the physics of the problem at hand behaves differently in different directions, isotropic meshes are not suitable.
A novel class of methods emerged to address the anisotropy issue by making use of metrics. The underlying idea is to generate a mesh where the edge lengths are unit with respect to some Riemannian metric. This method, known as *metric-based mesh adaptation* saw its initial development in [Diaz et al., 1997; Hecht and Mohammadi, 1997; Vallet, 1992]; it has since been used in a variety of research fields and a non exhaustive list of mesh generators using this metric concept is: `BAMG` [“BAMG: Bidimensional Anisotropic Mesh Generator”], `BL2D` [Laug and Borouchaki, 2003], `YAMS` [Frey, 2001], `Feflo.a` [Loseille and Löhner, 2010], `EPIC` [Michal and Krakos, 2012], `Forge3d` [Coupez, 2000], `Fun3d` [Jones, Nielsen, and Park, 2006], `Gamanic3d` [George, 2002], `MAdLib` [Compere et al., 2010], `MeshAdap` [Li, Shephard, and Beall, 2005], `Mmg3d` [Dobrzynski and Frey, 2008], `Mom3d` [Tam et al., 2000], `Tango` [Bottasso, 2004], `LibAdaptivity` [Pain et al., 2001]. For a review on metric based mesh adaptation and the current state of the art, the reader is

referred to [Alauzet and Loseille, 2016a] and references therein.

3.0.1 Overview of this chapter

This chapter reviews the metric-based mesh adaptation as it has been developed for use in CFD applications. No new results are developed in this chapter, it merely summarizes existing work on metric-based mesh adaptation and introduces the notation that will be used later on.

After having introduced some notions from differential geometry, the continuous mesh model will be presented. The discrete counterpart to the interpolation error in this continuous framework will then be introduced followed by an explanation on how discrete meshes are derived from the continuous formulation.

3.1 Metrics in the context of mesh adaptation

3.1.1 Euclidean metric space

The inner product between two vectors $\mathbf{u}$ and $\mathbf{v}$ belonging to a $\mathbb{R}^D$ spanned by basis vectors $(\mathbf{e}_i)_{i=1,\dots,D}$ can be expressed as:

$$\langle \mathbf{u}, \mathbf{v} \rangle = \left\langle \sum_i^D u_i \mathbf{e}_i, \sum_j^D v_j \mathbf{e}_j \right\rangle \tag{3.1}$$

$$= \sum_i^D u_i \left\langle \mathbf{e}_i, \sum_j^D v_j \mathbf{e}_j \right\rangle \tag{3.2}$$

$$= \sum_i^D \sum_j^D u_i \langle \mathbf{e}_i, \mathbf{e}_j \rangle v_j. \tag{3.3}$$

One can now define a matrix $\mathcal{M}$ whose components m_{ij} are given by

$$m_{ij} = \langle \mathbf{e}_i, \mathbf{e}_j \rangle$$

which leads to the following definition of the inner product on a Euclidean metric space:

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{M}} = \langle \mathbf{u}, \mathcal{M}\mathbf{v} \rangle = \mathbf{u}^{\mathsf{T}} \mathcal{M} \mathbf{v}. \tag{3.4}$$

When the basis vectors $\mathbf{e}_i$ are orthonormal the matrix $\mathcal{M}$ becomes the identity matrix as in that case $m_{ij} = \langle \mathbf{e}_i, \mathbf{e}_j \rangle = \delta_{ij}$. In this case, (3.4) is reduced to $\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{I}_D} = \mathbf{u}^{\mathsf{T}} \mathcal{I}_D \mathbf{v}$ or

$$\langle \mathbf{u}, \mathbf{v} \rangle_{\mathcal{I}_D} = \sum_i^D u_i v_i.$$

In the more general case however, where one does not necessarily have an orthonormal basis, one needs to take into account this matrix $\mathcal{M}$, which is called the *metric tensor* or simply the *metric*.

Formally, the Euclidean metric space of dimension D is denoted as $\left(\mathbb{R}^D, \mathcal{M}\right)$ where $\mathcal{M}$ is a symmetric positive definite tensor that defines the inner product. $\mathcal{M}$, represented by a $D \times D$ matrix, should have the following properties

1. $\forall(\mathbf{u},\mathbf{v}) \in \mathbb{R}^D \times \mathbb{R}^D$, $\langle \mathbf{u},\mathbf{v}\rangle_{\mathcal{M}} = \langle \mathbf{v},\mathbf{u}\rangle_{\mathcal{M}}$ (symmetric)
2. $\forall \mathbf{u} \in \mathbb{R}^D$, $\langle \mathbf{u},\mathbf{u}\rangle \geq 0$ (positive)
3. $\langle \mathbf{u},\mathbf{u}\rangle_{\mathcal{M}} = 0 \iff \mathbf{u} = \mathbf{0}$ (definite)

The properties imposed on $\mathcal{M}$ ensure that it defines an inner product. This, in turn, means that $\mathbb{R}^D$ is now a normed vector space $(\mathbb{R}^D, ||\cdot||_{\mathcal{M}})$ and a metric vector space $(\mathbb{R}^D, d_{\mathcal{M}}(\cdot,\cdot))$. The norm induced by the inner product will be

$$\forall \mathbf{u} \in \mathbb{R}^D,\ ||\mathbf{u}||_{\mathcal{M}} = \sqrt{\langle \mathbf{u},\mathbf{u}\rangle_{\mathcal{M}}},$$

and the distance function $d_{\mathcal{M}}(\cdot,\cdot)$ is given by

$$\forall(\mathbf{u},\mathbf{v}) \in \mathbb{R}^D \times \mathbb{R}^D,\ d_{\mathcal{M}}(\mathbf{u},\mathbf{v}) = ||\mathbf{v}-\mathbf{u}||_{\mathcal{M}} = \ell_{\mathcal{M}}(\mathbf{uv}).$$

With the structure of the Euclidean metric space now in place, one can also compute volumes and angles with respect to the metric tensor $\mathcal{M}$. The volume of some bounded subset K of a Euclidean metric space $\mathbb{R}^D$ with respect to $\mathcal{M}$ is given by

$$|K|_{\mathcal{M}} = \sqrt{\det \mathcal{M}}|K|_{\mathcal{I}_D},$$

and the angle between $\mathbf{u}$ and $\mathbf{v}$, both non-zero vectors, is

$$\cos(\theta_{\mathcal{M}}) = \frac{\langle \mathbf{u},\mathbf{v}\rangle_{\mathcal{M}}}{||\mathbf{u}||_{\mathcal{M}}||\mathbf{v}||_{\mathcal{M}}}.$$

While in the Euclidean metric space the length of the segment between $\mathbf{u}$ and $\mathbf{v}$ is just the distance between these two points from a bird's-eye view, this will not be the case in Riemannian metric spaces which will be introduced shortly.

Geometric interpretation

The metric tensor $\mathcal{M}$ can be diagonalized as follows

$$\mathcal{M} = \mathcal{R}\Lambda\mathcal{R}^{\mathsf{T}},$$

with $\mathcal{R}$ being the orthonormal matrix of eigenvectors and Λ the diagonal matrix with the eigenvalues $(\lambda_i)_{i=1,\dots,D}$ of $\mathcal{M}$ on the diagonal. The squared distance between two points $\mathbf{a}$ and $\mathbf{b}$ with respect to the metric $\mathcal{M}$ can thus be written as

$$d_{\mathcal{M}}(\mathbf{a},\mathbf{b})^2 = (\mathbf{b}-\mathbf{a})^{\mathsf{T}}\mathcal{M}(\mathbf{b}-\mathbf{a}) = (\mathbf{b}-\mathbf{a})^{\mathsf{T}}\Lambda(\mathbf{b}-\mathbf{a}).$$

The unit ball $\mathcal{B}_{\mathcal{I}}$ about a point $\mathbf{a}$ is the set of points whose distance in the metric $\mathcal{I}$ (the identity matrix) is equal to one. Obviously this will just be a circle in 2D, a sphere in 3D etc. The unit ball $\mathcal{B}_{\mathcal{M}}$ at a point $\mathbf{a}$ of the domain $\Omega \subset \mathbb{R}^D$ can be defined as follows

$$\begin{aligned}\mathcal{B}_{\mathcal{M}}(\mathbf{a}) &= \left\{ \boldsymbol{x} \in \Omega | \sum_{i=1}^{D} \lambda_i (x_i - a_i)^2 \leq 1 \right\} \\ &= \left\{ \boldsymbol{x} \in \Omega | \sum_{i=1}^{D} \left(\frac{x_i - a_i}{h_i} \right)^2 \leq 1 \right\}.\end{aligned}$$

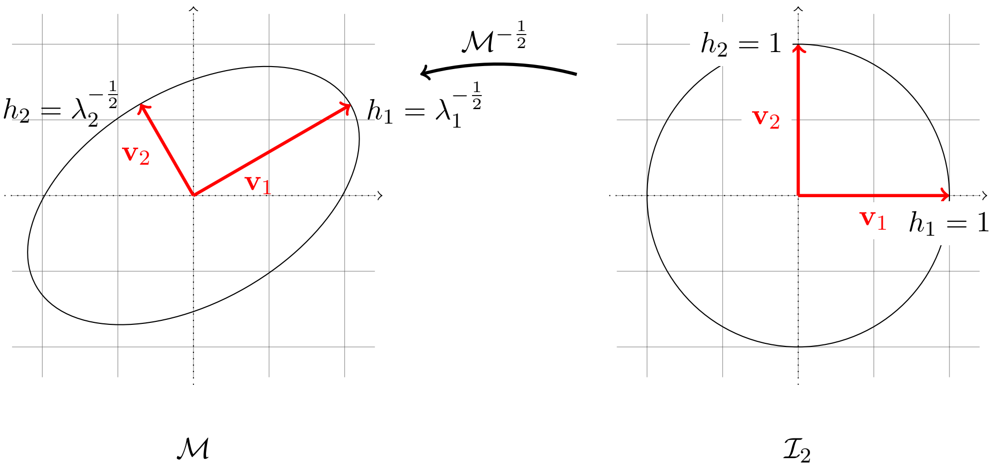


FIGURE 3.1: Mapping corresponding to the metric $\mathcal{M}$. On the left $h_i = \lambda_i^{-\frac{1}{2}}$ (λ_i being the eigenvalues of $\mathcal{M}$) and on the right $h_i = 1$ (as the eigenvalues of $\mathcal{I}_2$ are all equal to one).

$\mathcal{B}_{\mathcal{B}}$ will be an ellipsoid centered at $\mathbf{a}$ with the axes tilted according to the directions indicated by the eigenvectors of $\mathcal{M}$ and the size of each axis will be given by $h_i = \lambda_i^{-\frac{1}{2}}$. The mapping from the unit ball $\mathcal{B}_{\mathcal{I}}$ to the unit ball $\mathcal{B}_{\mathcal{M}}$ is given by $\mathcal{M}^{-\frac{1}{2}}$, where:

$$\mathcal{M}^{-\frac{1}{2}} = \mathcal{R}\Lambda^{-\frac{1}{2}}\mathcal{R}^{\mathsf{T}}.$$

For example, in 2D one will obtain:

$$\mathcal{M}^{-\frac{1}{2}} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 \end{bmatrix} \begin{bmatrix} \lambda_1^{-\frac{1}{2}} & 0 \\ 0 & \lambda_2^{-\frac{1}{2}} \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 \end{bmatrix}^{\mathsf{T}}.$$

A graphic representation of this mapping in 2D between $\mathcal{B}_{\mathcal{I}_2}$ and $\mathcal{B}_{\mathcal{M}}$ is shown in Figure 3.1.

3.1.2 Riemannian metric space

Up until now the metric $\mathcal{M}$ has been treated as a constant. That restriction will now be lifted and the metric $\mathcal{M}(\mathbf{x})$ is allowed to smoothly vary in the entire space. Such a space, with a metric tensor varying smoothly from point to point, is known as a Riemannian manifold. More precisely, a Riemannian manifold or Riemann space is a continuous manifold $\Omega \subset \mathbb{R}^D$ endowed with a metric $\mathcal{M}(\cdot)$ and is denoted by $(\mathbf{x}, \mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$. At each point $\mathbf{x}$ of the manifold, the metric $\mathcal{M}(\cdot)$ defines an inner product on the local tangent space $T_{\mathbf{x}}\Omega$. The tangent space $T_{\mathbf{x}}\Omega$ with the inner product defined by $\mathcal{M}(\cdot)$ is a Euclidean metric space. One can see a Riemannian manifold as an assembly of Euclidean metric spaces, deformed in such a way that the global behaviour of this space is no longer that of a Euclidean metric space.

For the application of mesh adaptation, the manifold will generally not be known. Only the metric $\mathcal{M}(\mathbf{x})$ and the computational domain $\Omega \subset \mathbb{R}^D$ will be known. A simplified space called a Riemannian metric space, denoted by $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$, will therefore be used. This Riemannian metric space can be assimilated to a Cartesian surface embedded in a higher dimensional space $\mathbb{R}^{D+1}$. Such a Cartesian surface in $\mathbb{R}^{D+1}$ is defined as

$$\mathcal{S} = (x_1, x_2, \ldots, x_D, \sigma(x_1, x_2, \ldots, x_D)),$$

where σ is a $\mathcal{C}^2$ scalar function defined on $\mathbb{R}^D$. The link between this surface $\mathcal{S}$ and the Riemannian metric space $(\mathcal{M}(\mathbf{x}))_{x\in\Omega}$ is clarified by a small example.

Example Take as the computational domain $\Omega = \left\{(x, y) \in \mathbb{R}^2 | -\frac{\pi}{2} \leq x, y \leq \frac{\pi}{2}\right\}$ and let $\mathcal{S} \subset \mathbb{R}^3$ be the Cartesian surface defined by

$$\mathcal{S} = \left\{(x, y, z) \in \mathbb{R}^3 | z = \sigma(x, y) = \sin(x^2 + y^2) \text{ with } (x, y) \in \Omega\right\}$$

This surface $\mathcal{S}$, a Cartesian surface embedded in $\mathbb{R}^3$, is shown on the left in Figure 3.2. Consider a curve γ on Ω, parametrized by $t \in [0, 1]$ such that $\gamma(t) = (x(t), y(t))$ and let its image on $\mathcal{S}$ be $\mathbf{c}(t)$. Thus $\mathbf{c}(t)$ can be written as

$$\mathbf{c}(t) = (\gamma(t), \sigma(\gamma(t))) = (x(t), y(t), \sigma(x(t), y(t))),$$

and its length can be computed by

$$\ell(\mathbf{c}) = \int_0^1 \left|\left|\frac{\mathrm{d}\mathbf{c}}{\mathrm{d}t}(t)\right|\right| \mathrm{d}t.$$

Dropping the dependence on t for ease of notation, one can now compute $||\mathrm{d}\mathbf{c}||^2$ on the surface $\mathcal{S}$:

$$\begin{aligned} ||\mathrm{d}\mathbf{c}||^2 &= \mathrm{d}x^2 + \mathrm{d}y^2 + \mathrm{d}z^2 = \mathrm{d}x^2 + \mathrm{d}y^2 + (\mathrm{d}(\sin(x^2 + y^2)))^2 \\ &= \mathrm{d}x^2 + \mathrm{d}y^2 + \left(4x^2\mathrm{d}x^2 + 8xy\mathrm{d}x\mathrm{d}y + 4y^2\mathrm{d}y^2\right)(\cos(x^2 + y^2))^2 \\ &= \begin{bmatrix} \mathrm{d}x^2 & \mathrm{d}y^2 \end{bmatrix} \underbrace{\begin{bmatrix} 1 + 4x^2(\cos(x^2 + y^2))^2 & 4xy(\cos(x^2 + y^2))^2 \\ 4xy(\cos(x^2 + y^2))^2 & 1 + 4y^2(\cos(x^2 + y^2))^2 \end{bmatrix}}_{\mathcal{M}(\mathbf{x})} \begin{bmatrix} \mathrm{d}x^2 \\ \mathrm{d}y^2 \end{bmatrix} \end{aligned}$$

From this example one can see that the Riemannian structure induced by the surface $\mathcal{S}$, defined by the equation $z = \sigma(x, y)$, is a metric space. This induced Riemannian metric space is shown on the right in Figure 3.2 where at given points $\mathbf{x}$ the unit ball of the metric $\mathcal{M}(\mathbf{x})$ is drawn. Furthermore it also becomes clear that computing the length of a segment on Ω (which is flat) with respect to $\mathcal{M}(\mathbf{x})$ will yield the same result as computing the length of the image of this segment on the (curved) surface $\mathcal{S}$.

In the context of mesh adaptation we will be computing the straight line parametrization of edges on Ω with respect to the metric $\mathcal{M}(\mathbf{x})$. The main idea of metric based mesh adaptation is to adapt the mesh by controlling this metric. The above example made clear that it is very straightforward to deduce the Riemannian metric space from the initial curved surface. The inverse problem is not so easy, but that will be of little concern here as the length of line segments on the curved manifold will never be computed; for our needs however, the Riemannian metric space will be an adequate mathematical object.

As the metric on the Riemannian metric space varies smoothly, there are only local definitions of distances, angles and volumes. One can however obtain global notions by applying integration formulae. For example, the length of a line segment $\mathbf{ab} = \mathbf{b} - \mathbf{a}$

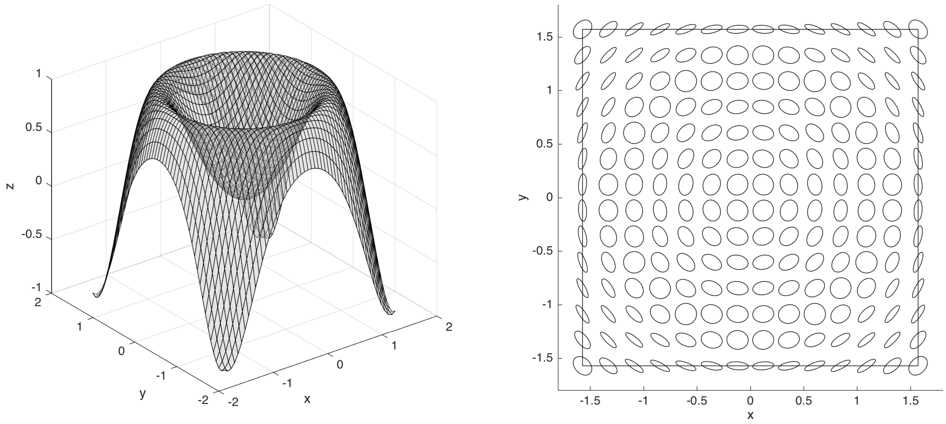


FIGURE 3.2: The surface $\mathcal{S}$ corresponding to $z = \sigma(x, y) = \sin(x^2 + y^2)$ (left) and a visualisation of the Riemannian metric space $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in[-\frac{\pi}{2},\frac{\pi}{2}]\times[-\frac{\pi}{2},\frac{\pi}{2}]}$ on the right where the unit ball corresponding to $\mathcal{M}(\mathbf{x})$ is plotted at different points (the size of the ellipses has been scaled down to allow for easier visualization). The length of the path between two points on Ω (right) computed w.r.t. $\mathcal{M}(\mathbf{x})$ is equal to the length of the curved path projected onto the surface $\mathcal{S}$ (left).

parametrized by $\gamma(t) = \mathbf{a} + t\mathbf{ab}$ is computed by:

$$\ell_{\mathcal{M}}(\mathbf{ab}) = \int_0^1 ||\gamma'(t)||_{\mathcal{M}} \mathrm{d}t = \int_0^1 \sqrt{\mathbf{ab}^{\mathsf{T}}\mathcal{M}(\mathbf{a} + t\mathbf{ab})\mathbf{ab}}\, \mathrm{d}t.$$

Similarly, the volume of a bounded subset K of the Riemannian metric space $\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{x\in\Omega}$ is computed by integrating:

$$|K|_{\mathcal{M}} = \int_K \sqrt{\det(\mathcal{M}(\mathbf{x}))}\, \mathrm{d}\mathbf{x}.$$

And finally, the angle between two line segments $\mathbf{uv}$ and $\mathbf{vw}$ on $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is defined by the unique real value $\theta \in [0, \pi]$ satisfying

$$\cos(\theta) = \frac{\langle \mathbf{uv}, \mathbf{vw} \rangle_{\mathcal{M}(\mathbf{v})}}{||\mathbf{uv}||_{\mathcal{M}(\mathbf{v})}||\mathbf{vw}||_{\mathcal{M}(\mathbf{v})}}.$$

This summarizes the basic properties of Riemannian metric spaces. The reader interested in more details on this subject are referred to [Bottasso, 2004; Berger, Cole, and Levy, 1987a; Berger, Cole, and Levy, 1987b; Berger, 2003].

3.2 Continuous mesh model

The Riemannian metric space can be seen as a continuous representation of a mesh. This continuous mesh can be defined using the following two sets of equivalent variables [Loseille and Alauzet, 2011a; Loseille and Alauzet, 2011b]:

$$(\mathcal{R}(\mathbf{x}), h_1(\mathbf{x}), \ldots, h_D(\mathbf{x})) \quad \text{or} \quad (\mathcal{R}(\mathbf{x}), d(\mathbf{x}), r_1(\mathbf{x}), \ldots, r_{D-1}(\mathbf{x})),$$

where $\mathcal{R}$ is again the matrix containing the eigenvectors of $\mathcal{M}$. Remember that, since $\mathcal{M}$ is symmetric positive definite, its spectral decomposition exists and is given by:

$$\mathcal{M}(\mathbf{x}) = \mathcal{R}(\mathbf{x})\Lambda(\mathbf{x})\mathcal{R}^{\mathsf{T}}(\mathbf{x})$$

with Λ the diagonal matrix of eigenvalues λ_i. As $h_i = \lambda_i^{-\frac{1}{2}}$, the metric $\mathcal{M}(\mathbf{x})$ can be completely defined by the set of variables $(\mathcal{R}(\mathbf{x}), h_1(\mathbf{x}), \ldots, h_D(\mathbf{x}))$.
Alternatively, the second set of variables, called local variables, defines the continuous mesh in terms of the anisotropy ratios $r_i(\cdot)$, the mesh density $d(\cdot)$ and P defined by:

$$P = \prod_{i=1}^{D-1} r_i, \tag{3.5}$$

The anisotropy ratios r_i are computed as follows:

$$r_i = h_i \left(\prod_{k=1}^{D} h_k \right)^{-\frac{1}{D}}, \tag{3.6}$$

and d is the mesh density defined as:

$$d = \frac{1}{\prod_{k=1}^{D} h_k} = \sqrt{\prod_{k=1}^{D} \lambda_k}. \tag{3.7}$$

Using the anisotropy ratios and the mesh density, h_i can be expressed as:

$$h_i = d^{-\frac{1}{D}} r_i \quad \text{with } i = 1 \ldots D-1,$$

and h_D as

$$h_D = d^{-\frac{1}{D}} P^{-1}.$$

$\mathcal{M}(\mathbf{x})$ can then be expressed in a second set of variables:

$$\mathcal{M}(\mathbf{x}) = d^{2/D}(\mathbf{x})\mathcal{R}(\mathbf{x}) \begin{pmatrix} r_i^{-2}(\mathbf{x}) & & \\ & \ddots & \\ & & r_D^{-2}(\mathbf{x}) \end{pmatrix} \mathcal{R}^{\mathsf{T}}(\mathbf{x}).$$

With this formulation, the anisotropy ratios and density of the mesh have been decoupled. The mesh density d affects the local accuracy of the mesh without influencing the anisotropy. One can furthermore define the complexity $\mathcal{C}$ of $\mathbf{M}$:

$$\mathcal{C}(\mathbf{M}) = \int_{\Omega} d(\mathbf{x})\mathrm{d}\mathbf{x} = \int_{\Omega} \sqrt{\det(\mathcal{M}(\mathbf{x}))}\mathrm{d}\mathbf{x}.$$

This complexity can be seen as the continuous equivalent of the number of vertices in the discrete mesh and is a fundamental notion when formulating and solving the mesh adaptation problem.
To go from the continuous representation of a mesh by means of the Riemannian metric to a discrete mesh on which the problem under consideration can be solved, one more step is needed: the notion of *unit elements* and a *unit mesh*. In what follows, we will discuss how one can generate a discrete object, the adapted anisotropic mesh, from the continuous

object of the Riemannian metric.

3.2.1 From Riemannian metric spaces to anisotropic unit meshes

The principal idea of metric based mesh adaptation is to construct a *unit mesh* in the Riemannian metric, which will then translate to an adapted, anisotropic mesh in the Euclidean space. Such a unit mesh will be uniform and isotropic in the Riemannian metric space, but upon transformation to the Euclidean space will yield an adapted and anisotropic mesh.

In order to understand what a *unit mesh* is, it is best to begin by defining what a *unit element* is.

Definition 3.2.1. *Let K be a mesh element, then K is called* unit *with respect to the metric $\mathcal{M}$ is all its edge lengths are unit in the metric $\mathcal{M}$.*

A triangle in 2D or a tetrahedron on 3D is considered a unit element if for each edge $\mathbf{e}_i$ ($i = 1, \cdots, 3$ in 2D, $i = 1, \cdots, 6$ in 3D) the following holds:

$$\ell_{\mathcal{M}}(\mathbf{e}_i) = 1 \text{ with } \ell_{\mathcal{M}}(\mathbf{e}_i) = \sqrt{\mathbf{e}_i^{\mathsf{T}} \mathcal{M} \mathbf{e}_i}. \tag{3.8}$$

A mesh where all elements are unit elements is called a *unit mesh*; in practice however, one can only obtain quasi-unit meshes. This is due to the fact that a mesh filling the computational domain Ω composed of unit elements may not exist. To circumvent this problem of existence, one can relax the definition of unit mesh to: *a mesh where all elements are quasi-unit with respect to a Riemannian metric space* $\mathbf{M}$. Instead of demanding the length of the edges of each element be 1 in the given Riemannian metric space, we want the lengths of all edges $\mathbf{e}_i$ of a quasi-unit element to fall within the interval:

$$\ell_{\mathcal{M}}(\mathbf{e}_i) \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right] \quad \forall i.$$

Note that only controlling the edge sizes of mesh elements does not exclude zero-volume elements where the element collapses onto itself. This can be avoided by adding a constraint on the volume of the elements. Rather than imposing some minimal volume, one usually imposes a minimal *quality of the element* $\mathcal{Q}_{\mathcal{M}}$ which indirectly also works as a constraint on the volume. In 3D, the *quality* of a tetrahedron K is defined as:

$$\mathcal{Q}_{\mathcal{M}}(K) = \frac{36}{3^{\frac{1}{3}}} \frac{|K|_{\mathcal{M}}^{\frac{2}{3}}}{\sum_{i=1}^{6} \ell_{\mathcal{M}}^2(\mathbf{e}_i)}$$

Definition 3.2.2. *An element K is said to be* quasi-unit *with respect to $\mathcal{M}(\mathbf{x})$ if the length of all its edges $\mathbf{e}_i$ satisfy*

$$\ell_{\mathcal{M}}(\mathbf{e}_i) \in \left[\frac{1}{\sqrt{2}}, \sqrt{2}\right],$$

and

$$\mathcal{Q}_{\mathcal{M}}(K) \in [\alpha, 1] \ \textit{with} \ \alpha > 0.$$

The following definition for a *quasi-unit* mesh can now be formulated.

Definition 3.2.3. *Let $\mathcal{H}$ be a discrete mesh of the domain $\Omega \subset \mathbb{R}^D$, then $\mathcal{H}$ is* unit *with respect to the Riemannian metric space $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$ is all its elements are quasi-unit.*

Discrete	Continuous
element K	$\mathcal{M}(\mathbf{x})$
mesh $\mathcal{H}$	$\mathbf{M} = (\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$
# vertices N	$\mathcal{C}(\mathbf{M})$
orientation	$\mathcal{R}(\mathbf{x})$
stretching	$r_i(\mathbf{x})$
element size	$d(\mathbf{x})$

TABLE 3.1: Continuous-discrete duality

Such a unit mesh composed of such quasi-unit elements can always be created in the given Riemannian metric space [Loseille and Alauzet, 2011a].

There exists a bijection between the continuous and the discrete framework, the elements of this duality are summarized in Table 3.1. The previous section has emphasized the role of metric tensors and Riemannian metric spaces as useful mathematical tools to prescribe sizes and directions to adaptive meshers. Additionally, these differential geometry notions are more than just a simple tool for mesh generation. Indeed, the Riemannian metric spaces can be seen as continuous models representing meshes. The fundamental consequence is that all kind of mathematical analysis can be performed using such spaces for which powerful mathematical tools are available. In practice however, the problem at hand is solved on a discrete mesh and the resulting solution will be a discrete solution. Hence the continuous solution, gradient and Hessian are obtained using derivative recovery methods as $\mathbf{L}_2$ projection of Green's formulat (see [Zienkiewicz and Zhu, 1992a; Zienkiewicz and Zhu, 1992b]).

3.3 Continuous linear interpolation error

The previous section introduced the Riemannian metric space and gave some hints as to its use for mesh adaptation. The linear interpolation error $||u - \Pi_h u||_{L^1(\Omega_h)}$ will be used to drive the mesh adaptation. In this formulation, Π_h is the linear interpolation operator, defined by a unit mesh $\mathcal{H}$ on a discretized domain Ω_h. However, this discrete form of the interpolation error is not directly amenable to the adaptation of a continuous mesh. In [Loseille and Alauzet, 2011a] a well-posed continuous linear interpolation operator $\pi_{\mathcal{M}}$ was defined in order to overcome this. The subscript $\mathcal{M}$ in the continuous interpolation operator indicates that the interpolation is done with respect to the continuous mesh $(\mathcal{M}(\mathbf{x}))_{\mathbf{x}\in\Omega}$. The resulting continuous linear interpolation error $||u - \pi_{\mathcal{M}} u||_{L^1(\Omega)}$ is a reliable model for $||u - \Pi_h u||_{L^1(\Omega_h)}$.

In the case where u is a nonlinear twice differentiable function pertaining to some function space $\mathbf{H}(\Omega)$, and $H_u(\mathbf{x})$ its Hessian, then [Loseille and Alauzet, 2011a] show that the continuous linear interpolation error estimate in the $3D$ case is

$$\forall \mathbf{x} \in \Omega,\ e_{\mathcal{M}}(\mathbf{x}) = |u - \pi_{\mathcal{M}} u|(\mathbf{x}) = \frac{1}{20}\text{trace}\left(\mathcal{M}(\mathbf{x})^{-\frac{1}{2}}|H_u(\mathbf{x})|\mathcal{M}(\mathbf{x})^{-\frac{1}{2}}\right) \tag{3.9}$$

$$= \frac{1}{20}\sum_{i=1}^{3} h_i^2(\mathbf{x})\mathbf{v}_i^{\mathsf{T}}(\mathbf{x})|H_u(\mathbf{x})|\mathbf{v}_i(\mathbf{x}), \tag{3.10}$$

and in the $2D$ case

$$\forall \mathbf{x} \in \Omega,\ e_{\mathcal{M}}(\mathbf{x}) = |u - \pi_{\mathcal{M}} u|(\mathbf{x}) = \frac{1}{8}\text{trace}\left(\mathcal{M}(\mathbf{x})^{-\frac{1}{2}}|H_u(\mathbf{x})|\mathcal{M}(\mathbf{x})^{-\frac{1}{2}}\right) \tag{3.11}$$

$$= \frac{1}{8}\sum_{i=1}^{2} h_i^2 \mathbf{v}_i^{\mathsf{T}}(\mathbf{x})|H_u(\mathbf{x})|\mathbf{v}_i(\mathbf{x}), \tag{3.12}$$

where $|H_u|$ is obtained by taking the absolute value of the eigenvalues of H_u.

Relations (3.10) and (3.12) can be cast into the more general form of the following error model:

$$e_{\mathcal{M}} = \left(\sum_{i=1}^{D} h_i^{\beta}\gamma_i\right)^{\alpha} \tag{3.13}$$

with $\gamma_i(\mathbf{x}) = |\mathbf{v}_i(\mathbf{x})^{\mathsf{T}} H_u(\mathbf{x})\mathbf{v}_i(\mathbf{x})|$. Taking $\alpha = 1$ and $\beta = 2$ will yield, up to a constant, the previous expression.

Furthermore, using the anisotropy ratios r_i and mesh density d defined in (3.6) and (3.7) respectively, a simple change of variables leads to:

$$e_{\mathcal{M}}(\mathbf{x}) = \frac{1}{20}\left(d(\mathbf{x})^{-\frac{2}{3}}\sum_{i=1}^{3} r_i(\mathbf{x})^{\frac{2}{3}}\mathbf{v}_i^{\mathsf{T}}(\mathbf{x})|H_u(\mathbf{x})|\mathbf{v}_i(\mathbf{x})\right), \tag{3.14}$$

and

$$e_{\mathcal{M}}(\mathbf{x}) = \frac{1}{8}\left(d(\mathbf{x})^{-1}\sum_{i=1}^{2} r_i(\mathbf{x})\mathbf{v}_i^{\mathsf{T}}(\mathbf{x})|H_u(\mathbf{x})|\mathbf{v}_i(\mathbf{x})\right). \tag{3.15}$$

In the same way, (3.13) can be rewritten as

$$e_{\mathcal{M}} = d^{-\frac{\alpha\beta}{D}}\left(\sum_{i=1}^{D-1} r_i^{\beta}\gamma_i + P^{-\beta}\gamma_D\right)^{\alpha}. \tag{3.16}$$

The importance of this change of variables will become clear in the next section where the computation of optimal meshes is explained.

3.4 Defining the optimal mesh

In this context, the mesh adaptation problem can easily be formulated as an optimization problem where we seek the optimal mesh that minimizes the continuous error model:

$$\text{Find } \mathbf{M}_{L^p} = \underset{\mathbf{M}}{\text{argmin}}\, E_p(\mathbf{M}) = \left(\int_{\Omega} e_{\mathcal{M}}^p \mathrm{d}\mathbf{x}\right)^{\frac{1}{p}} = \left(\int_{\Omega} (u - \pi_{\mathcal{M}} u)^p \mathrm{d}\mathbf{x}\right)^{\frac{1}{p}} \tag{3.17}$$

under the constraint

$$\mathcal{C}(\mathbf{M}) = \int_{\Omega}\left(\prod_{i=1}^{D} h_i\right)^{-1} \mathrm{d}\mathbf{x} = N. \tag{3.18}$$

The advantage of having a continuous formulation is that all the mathematical tools available for convex optimization are at our disposal.

By rewriting the optimization problem in terms of the variables r_i and d, one obtains

$$\mathbf{M}_{L^p} = \underset{((r_i)_{i=1,\dots,D},d)}{\operatorname{argmin}} \int_\Omega d^{-\frac{\alpha\beta p}{D}} \left(\sum_{i=1}^{D-1} r_i^\beta \gamma_i + P^{-\beta}\gamma_D \right)^{\alpha p} \mathrm{d}\mathbf{x} \text{ subject to } \int_\Omega d \,\mathrm{d}\mathbf{x} = N. \quad (3.19)$$

The change of variables has two advantages: firstly, the constraint $\mathcal{C}(\mathbf{M}) = N$ is now linear in the variable d, making it a convex problem, and secondly, the problem is decoupled into a global and a local problem. After solving for the optimal local anisotropy ratios, one can find the optimal density d independently afterwards.

Hence, the optimization problem can be solved analytically [Loseille, 2008] and the resulting optimal metric is:

$$\mathcal{M}_{L^p} = N^{\frac{2}{D}} \left(\int_\Omega \det(|H_u|)^{\frac{p}{2p+D}} \right)^{-\frac{2}{D}} \det(|H_u|)^{\frac{-1}{2p+D}} |H_u|. \quad (3.20)$$

From this optimal metric, a quasi-unit discrete mesh can then be generated which will capture the anisotropy of the physics of the problem. For a more in-depth treatment of this process, interested readers are referred to [Frey and George, 2007; Loseille, 2008; Loseille, Dervieux, and Alauzet, 2010; Loseille and Alauzet, 2011a; Loseille and Alauzet, 2011b].

In the context of UQ, the stochastic response we try to approximate on the stochastic space is not directly the solution of some physical phenomenon. The QoI is rather a functional of some physical solution, parametrized by random variables. Nevertheless the metric-based mesh adaptation approach can be used to construct a surrogate model of the stochastic response. In the next chapter an extension of metric-based mesh adaptation to the stochastic space will be presented, the utility of such a method will be discussed and its performance will be evaluated.

Chapter 4

Metric-based stochastic error control

4.1 Introduction

Popular UQ methods such as Stochastic Galerkin (SG), Polynomial Chaos Expansions (PCE) and Stochastic Collocation (SC), are highly efficient when the Quantity of Interest (QoI) response is sufficiently smooth. However, in the case where the stochastic response is of low regularity and exhibit discontinuities, these global polynomial approximations are not a good choice as they suffer from Gibbs oscillations. In order to address this deficiency, the Multi-Element gPC (ME-gPC) method [Wan and Karniadakis, 2005; Wan and Karniadakis, 2006], and Multi-Element Probabilistic Collocation Method (ME-PCM) [Foo, Wan, and Karniadakis, 2008; Foo and Karniadakis, 2010], were devised. These methods split the parameter space into rectangular subdomains on which a local polynomial approximation is constructed. While some anisotropy can be achieved, unless the discontinuities run along one of the principal axes, this rectangular subdivision will not be able to capture these discontinuities. The degrading effects will still be present in the elements traversed by the discontinuities. Furthermore, these ME methods place the samples following a tensor structure resulting in a fast increase in the number of samples as the number of elements increases and on top of that, not all samples may be reused after the splitting of an element. Most of these problems are alleviated in the Minimal Multi-Element Stochastic Collocation (MME-SC) method [Jakeman, Narayan, and Xiu, 2013], which relies on elements of irregular shapes where a discontinuity detector is used to split the parametric space into a minimal number of elements defined by the discontinuities. Moreover, these numerical techniques are often applied to problems with complex functionals but simple unimodal and smooth underlying probability measures that do not account for multivariate correlated inputs.
In an attempt to cure Gibbs phenomena, the use of a multi-wavelet instead of polynomial basis has been proposed [Le Maıtre et al., 2004a; Le Maıtre et al., 2004b] along with an $h-$refinement technique using hypercubes, similar to the aforementioned ME-gPC method. A different decomposition is proposed with the Simplex-Stochastic Collocation (SSC) method [Witteveen, Loeven, and Bijl, 2009], a stochastic finite element–type method which uses a Newton-Cotes quadrature in (non hypercube) simplex elements. Gibbs oscillations are avoided using Local-Extremum Conserving limiter while later developments include higher order interpolation by use of Essentially Non-Oscillatory (ENO) stencils [Witteveen and Iaccarino, 2013a], subcell resolution [Witteveen and Iaccarino, 2013b], applications to non-hypercube domains [Witteveen and Iaccarino, 2012b] and higher dimensional problems [Witteveen and Iaccarino, 2010; Edeling, Dwight, and Cinnella, 2016].

Most of these adaptive methods use refinement criteria that are heuristics and require tuning to specific problems. More rigorous refinement criteria making use of dual-based

---

Chapter partially based on Van Langenhove J., Lucor D., F. Alauzet and A. Belme. "Goal-oriented control of stochastic system approximations using metric-based anisotropic adaptation". (to be submitted to *Journal of Computational Physics*).

a posteriori error estimates in the stochastic space were developed in [Mathelin and Le Maître, 2007; Butler, Dawson, and Wildey, 2011; Butler, Constantine, and Wildey, 2012; Bryant, Prudhomme, and Wildey, 2015].

Whenever possible, one would like the adaptation technique to work out of the box for any problem. An error estimator driving the adaptation process is therefore preferable over a, possibly problem dependent, heuristic adaptation criterion that requires choosing and tuning extra parameters.

In this chapter an extension of the metric-based mesh adaptation technique introduced in Chapter 3 to the parametric space is proposed. The idea will be to make use of a powerful mathematical framework for efficient *anisotropic* multivariate parametric refinements.

The stochastic response is then approximated on the optimized mesh using a tessellation of linear simplex/tetrahedron elements, well adapted for discontinuous responses. In this respect it resembles the SSC method which uses a similar discretization. In contrast, the method proposed here will, by controlling the interpolation error, drive $h-$adaptivity in the parametric space using metric-based adaptation.

While this clear choice for $h-$adaptivity in the parametric space nullifies the need for user-chosen refinement parameters typical of $hp-$adaptive methods, it also means that in the case of highly regular response surfaces, this method will not have the benefit of the superconvergence brought by $p-$refinement in those situations.

This chapter starts with the formal setup of our stochastic problem while the stochastic error estimate for both the discrete and continuous framework is proposed in Section 4.2. Next, the optimal stochastic mesh as the solution to an optimization problem is computed in Section 4.3. The adaptive strategies and a review of the Simplex-Stochastic Collocation method are outlined in Section 4.4. In Section 4.5.1 the proposed approach is validated in stochastic test functions and the fluid mechanical piston problem. A short conclusion ends this chapter.

4.2 Formal error estimation

An abstract model formulation is first considered in this section. It consists of a boundary-value problem defined on an open bounded domain $\Omega_{\mathbf{x}} \subset \mathrm{R}^{D_{\mathbf{x}}}$ with $D_{\mathbf{x}}$ the dimension of the physical space. Furthermore, we suppose our problem depends on several uncertain parameters that will be defined as random variables. We therefore introduce a probability space $(\Omega_{\boldsymbol{\xi}}, \mathcal{B}, \mathcal{P})$ where $\Omega_{\boldsymbol{\xi}}$ is the sample space, $\mathcal{B}$ is a σ-algebra and $\mathcal{P}$ the probability measure. We denote $\boldsymbol{\xi}(\omega) = (\xi_1(\omega), \xi_2(\omega), \ldots, \xi_{D_{\boldsymbol{\xi}}}(\omega))$ the vector of random variables that is sufficient to quantify our set of uncertain parameters, where $\omega \in \Omega_{\xi}$ and $D_{\boldsymbol{\xi}}$ represents the dimension of the random input space. Throughout this manuscript, it will be assumed that the random input can be represented by a finite-dimensional probability space (the so called *finite-dimensional noise assumption*). This ensures that the input can always be represented by a finite-dimensional set of random variables $\boldsymbol{\xi}(\omega)$. Furthermore let $\rho_{\boldsymbol{\xi}}$ denote the joint probability density function of $\boldsymbol{\xi}$ and let Ξ be the parameter space to which the ξ's belong. Each realization in the probability space corresponds, by a mapping defined by the probability density function of the random variables, to a parameter value. In short: $\boldsymbol{\xi} \in \Xi \equiv \prod_{i=1}^{D_{\boldsymbol{\xi}}} \Xi_i$ where Ξ_i is the image of $\xi_i(\Omega_{\boldsymbol{\xi}})$.
Thus, for a particular set of parameters $\boldsymbol{\xi}_{(i)}$, the model problem can be cast in the following abstract form:

$$\Psi(\boldsymbol{\xi}, w(\boldsymbol{\xi}, \mathbf{x})) = 0 \tag{4.1}$$

where Ψ is a state equation (for example a steady Navier-Stokes or Euler system) having $w(\boldsymbol{\xi}, \mathbf{x})$ for exact solution. As we will see in the following, the "exact" wording does not refer to a finite and fixed solution – as it is a random quantity – but designates the reference solution that is free of numerical errors. Indeed, the exact solution of a model is often out of reach for real-life engineering applications, and one has to rely on approximate numerical methods in order to approach it. For instance, for the previously introduced set of parameters $\boldsymbol{\xi}_{(i)}$, we may have a numerical tool that is capable of solving the corresponding deterministic discrete problem on a given spatial discretization $\mathcal{H}_{h_\mathbf{x}}$, and produce an approximate solution of $\Psi(\boldsymbol{\xi}_{(i)}, w(\boldsymbol{\xi}_{(i)}, \mathbf{x})) = 0$ at the expense of a certain computational cost, i.e.:

$$\Psi_{h_\mathbf{x}}(\boldsymbol{\xi}_{(i)}, w_{h_\mathbf{x}}(\boldsymbol{\xi}_{(i)}, \mathbf{x})) = 0, \tag{4.2}$$

with $w_{h_\mathbf{x}}$ the deterministic discrete solution associated to sample $\boldsymbol{\xi}_{(i)}$ and solved on a mesh $\mathcal{H}_{h_\mathbf{x}}$. As will be seen later, there obviously exists an implicit dependence or "coupling" between the choice of an adequate spatial discretization and the parameter value. For instance, certain parametric values might drastically affect the flow regime which will require an adequate mesh adaptation in order to produce a valid and reliable numerical approximation of the solution.

Here, we will be more interested in an accurate approximation of a *scalar* quantity of interest (QoI) j that is computed from the solution $w(\boldsymbol{\xi}, \mathbf{x})$ (and therefore depends on the uncertain random vector $\boldsymbol{\xi}$), than in the solution of the problem itself. We define our QoI sample obtained from the solution of the deterministic model built for the set of parameters $\boldsymbol{\xi}_{(i)}$ and a given spatial discretization $\mathcal{H}_{h_\mathbf{x}}$ as:

$$j(\boldsymbol{\xi}_{(i)}) = J\big(\boldsymbol{\xi}_{(i)}, w_{h_\mathbf{x}}(\boldsymbol{\xi}_{(i)}, \mathbf{x})\big), \tag{4.3}$$

where J is the observation operator. Point-wise evaluations of the QoI are limited in practice as each evaluation involves a costly simulation. Predictions of j for new parametric values or statistical information (e.g. moments) of interest related to j may be more efficiently computed from a continuous approximation, a surrogate model, which is built across the span of the $\boldsymbol{\xi}$ parametric range. One way to construct these approximations is via a discretization $\mathcal{H}_{h_{\boldsymbol{\xi}}}$ of the multivariate stochastic (uncertain parameter) space.

The random input space will be discretized using the metric-based approach presented in the previous chapter. This discretization relies a finite number of samples (or design of experiments (DoE)) which are mapped through the deterministic model and the surrogate stochastic model for j is built from the computations on these samples:

$$j_{h_{\boldsymbol{\xi}}}(\boldsymbol{\xi}) = J_{h_\xi}\big(\boldsymbol{\xi}, w_{h_\mathbf{x}}(\boldsymbol{\xi}, \mathbf{x})\big). \tag{4.4}$$

We can thus define the *total error* committed on our QoI for a certain set of parameters:

$$\delta j(\boldsymbol{\xi}) \equiv j - j_{h_{\boldsymbol{\xi}}} = J(\boldsymbol{\xi}, w) - J_{h_{\boldsymbol{\xi}}}(\boldsymbol{\xi}, w_{h_\mathbf{x}}). \tag{4.5}$$

where $J_{\boldsymbol{\xi}}(\boldsymbol{\xi}, w_h)$ denotes the approximate QoI.

The *exactness* of the QoI thus depends on two components:

1. on the *deterministic* discrete solution $w_{h_\mathbf{x}}(\xi)$ error and its choice of spatial discretization $\mathcal{H}_{h_\mathbf{x}}$,
2. on the *stochastic* error committed by discretizing the stochastic space $\mathcal{H}_{h_{\boldsymbol{\xi}}}$ and building the surrogate model $J_{h_{\boldsymbol{\xi}}}$.

Usually the two sources of error, deterministic and stochastic, are treated separately, and an error control strategy is applied either in the deterministic approximation space (e.g. mesh control/adaptation) [Venditti and Darmofal, 2003; Fidkowski and Darmofal, 2011; Loseille, 2008; Palacios et al., 2012] or into the stochastic approximation space [Mathelin and Le Maître, 2007; Witteveen, Loeven, and Bijl, 2009; Butler, Dawson, and Wildey, 2011]. However, the interplay between the errors as well as which one dominates the computation of j remain very important questions which will be addressed later in this thesis. In order to give some elements of answer, we first split the total error in two contributions:

$$\delta j(\boldsymbol{\xi}) = \underbrace{J(\boldsymbol{\xi}, w) - J(\boldsymbol{\xi}, w_{h_{\mathbf{x}}})}_{\varepsilon(h_{\mathbf{x}}, h_{\boldsymbol{\xi}})} + \underbrace{J(\boldsymbol{\xi}, w_{h_{\mathbf{x}}}) - J_{h_{\boldsymbol{\xi}}}(\boldsymbol{\xi}, w_{h_{\mathbf{x}}})}_{\eta(h_{\mathbf{x}}, h_{\boldsymbol{\xi}})}. \tag{4.6}$$

where $J(\boldsymbol{\xi}, w_{h_{\mathbf{x}}})$ only accounts for errors introduced by the deterministic approximation of j. While both error contributions, (ε, η) are random quantities depending on $\boldsymbol{\xi}$, we assume that they will be controlled (in a complementary fashion) in the different approximation spaces: – ε will be controlled via *deterministic* refinement (at given $h_{\boldsymbol{\xi}}$) and – η will be controlled via *stochastic* refinement (at given $h_{\mathbf{x}}$).

In practice, δj being a random quantity, we will be interested in lowering the *average* QoI total error, that we express as:

$$\overline{\delta j} \equiv \mathbb{E}\left[\delta j(\boldsymbol{\xi})\right] = \underbrace{\mathbb{E}\left[\varepsilon\right]}_{\overline{\varepsilon}} + \underbrace{\mathbb{E}\left[\eta\right]}_{\overline{\eta}} \tag{4.7}$$

In this chapter, we will focus on controlling $\overline{\eta}$ using $h-$refinement on the stochastic space. The control of both contributions will be the subject of the next chapter.

4.2.1 Stochastic error estimate

The stochastic error contribution in (4.6) will be controlled through the $\mathbf{L}^p$ norm of the interpolation error.

We propose an error estimate of the numerical approximation of the solution in the parametric space inspired by the previous developments. Motivated by the need for anisotropic information, we wish to deploy the notion of Riemannian metric field in the stochastic space. For the type of applications we consider, it is common knowledge that the dependence of the QoI on the random variables is anisotropic as we often encounter singularities and sharp response gradients. To this purpose, following [Loseille, 2008] we will control the stochastic error through the $\mathbf{L}^p$ norm of the interpolation error

$$\left\| J(\boldsymbol{\xi}, w_{h_{\mathbf{x}}}) - \pi_{h_{\boldsymbol{\xi}}} J(\boldsymbol{\xi}, w_{h_{\mathbf{x}}}) \right\|_{\mathbf{L}^p(\Xi)}$$

where $\pi_{h_{\boldsymbol{\xi}}}$ is the (linear) interpolation operator in the parameter space. For the purpose of this manuscript we will focus on the $\mathbf{L}^1$ norm of the interpolation error in the stochastic space. This is a purely practical choice, well adapted for approximation of potentially discontinuous solutions, but there is no restriction in using a different p-norm.

For deterministic problems, this kind of approach has been proposed to capture all the scales/singularities of the system, and has been applied to deterministic CFD problems [Loseille, 2008]. In this case, a sensor is defined (for CFD applications a sensor will be a prescribed field: density, Mach, ...) and some norm of the interpolation error associated to the sensor is controlled by anisotropic mesh refinement. Several differences appear naturally when transposing this approach in the stochastic context, leading to a different interpretation. First, what plays the role of the sensor in our case is the stochastic scalar QoI j. In a deterministic context, this would be seen as a scalar quantity for each DoE and would not be controlled directly. Instead one would have to make use of an adjoint-based approach as described in the previous section. Second, the $\mathbf{L}^p$ norm of the interpolation error of j on the parameter space, now equipped with a probability measure $\mathcal{P}$, introduces the parameters probability density function in the formulation:

$$\bar{\eta} = \mathbb{E}\left[\eta\right] = \int_{\Xi} |J(\boldsymbol{\xi}, w_{h_{\mathbf{x}}}) - \pi_{h_{\boldsymbol{\xi}}} J(\boldsymbol{\xi}, w_{h_{\mathbf{x}}})| \rho_{\boldsymbol{\xi}} \mathrm{d}\boldsymbol{\xi}, \tag{4.8}$$

where $\rho_{\boldsymbol{\xi}}$ is again the joint probability density function (pdf) of $\boldsymbol{\xi}$. The probability density function acts as a weighting of the interpolation error, but the formulation is more straightforward than a deterministic goal-oriented one as it does not involve an adjoint solution. The fact that $\mathbb{E}_{h_{\boldsymbol{\xi}} \to 0}\left[\eta\right] \to 0$ will insure a *convergence in the mean* of the (piecewise linear) interpolated surrogate. Thanks to Markov's inequality, this convergence in the mean will insure that it converges in probability, which in turn implies convergence in distribution. Moreover, triangle inequality[1] will insure that the mean value of the surrogate will converge to the exact QoI mean $\bar{j} \equiv \mathbb{E}[J]$ at least as fast as the expectation of the interpolation error, i.e.:

$$\left|\bar{j} - \mathbb{E}[\pi_h J]\right| \leq \mathbb{E}\left[\eta\right]. \tag{4.10}$$

4.3 Stochastic continuous model

The Estimate (4.8) in the continuous framework of Riemannian metric spaces was derived in Chapter 3 and is recalled here:

$$\mathbf{E}_{\xi}(\mathbf{M}) \;=\; \int_{\Xi} \operatorname{trace}\left(\mathcal{M}^{-\frac{1}{2}}(\boldsymbol{\xi})\, \mathbf{H}_j(\boldsymbol{\xi})\, \mathcal{M}^{-\frac{1}{2}}(\boldsymbol{\xi})\right)\, \mathrm{d}\boldsymbol{\xi} \tag{4.11}$$

where in this case:

$$\mathbf{H}_j(\boldsymbol{\xi}) = \rho_{\boldsymbol{\xi}} \cdot H(j(\boldsymbol{\xi}))$$

with $H(j(\boldsymbol{\xi}))$ the Hessian matrix of $j(\boldsymbol{\xi})$ of size $D_{\boldsymbol{\xi}} \times D_{\boldsymbol{\xi}}$.

The *stochastic optimisation problem* is then formulated as follows:

$$\text{Find } \mathbf{M}_{\xi}^{opt} = \underset{\mathbf{M}}{\operatorname{argmin}}\ \mathbf{E}_{\boldsymbol{\xi}}(\mathbf{M}), \quad \text{subject to} \;\; \mathcal{C}(\mathbf{M}) = N_{\boldsymbol{\xi}} \tag{4.12}$$

where $N_{\boldsymbol{\xi}}$ denotes a given number of samples (or CFD computations) which corresponds to a targeted computational effort constraint. The notation $\mathbf{M}_{\boldsymbol{\xi}}^{opt}$ holds for the optimal metric that minimises the expectation of the continuous interpolation error in the parameter space. We will use this metric to build a simplex tessellation of the parameter space, which is, roughly the mesh associated with Ξ.

---

[1] This is sometimes called the reverse triangle inequality:

$$\left|\|a\| - \|b\|\right| \leq \|a - b\|\,. \tag{4.9}$$

Let $H(j(\boldsymbol{\xi}))$ denote the Hessian of the stochastic response, then the optimal stochastic metric, solution to the optimisation problem (4.12) can be written as :

$$\mathcal{M}_{\boldsymbol{\xi}}^{opt} = N_{\boldsymbol{\xi}}^{\frac{2}{D_{\boldsymbol{\xi}}}} \left(\int_{\Omega_{\boldsymbol{\xi}}} \det(\rho_{\boldsymbol{\xi}} |H(j(\boldsymbol{\xi}))|)^{\frac{1}{2+D_{\boldsymbol{\xi}}}} \mathrm{d}\boldsymbol{\xi} \right)^{-\frac{2}{D_{\boldsymbol{\xi}}}} \det(\rho_{\boldsymbol{\xi}} |H(j(\boldsymbol{\xi}))|)^{-\frac{1}{2+D_{\boldsymbol{\xi}}}} |\rho_{\boldsymbol{\xi}} H(j(\boldsymbol{\xi}))| \tag{4.13}$$

and the error estimate on this optimal metric is given by:

$$\mathbf{E}_{\boldsymbol{\xi}}^{opt}(\mathbf{M}_{\boldsymbol{\xi}}^{opt}) = D_{\boldsymbol{\xi}} N_{\boldsymbol{\xi}}^{-\frac{2}{D_{\boldsymbol{\xi}}}} \underbrace{\left(\int_{\Xi} \det(\rho_{\boldsymbol{\xi}} |H(j(\boldsymbol{\xi}))|)^{\frac{1}{2+D_{\boldsymbol{\xi}}}} \mathrm{d}\boldsymbol{\xi} \right)^{\frac{2+D_{\boldsymbol{\xi}}}{D_{\boldsymbol{\xi}}}}}_{\mathcal{K}_{\boldsymbol{\xi}}} \tag{4.14}$$

In what follows, where we will explain how these results were obtained, the subscript $\boldsymbol{\xi}$ will be dropped for ease of notation.

We recall that the model for the continuous interpolation error on a continuous mesh $\mathcal{M}(\cdot)$ introduced in Chapter 3 was given by (3.13):

$$e_{\mathcal{M}} = \left(\sum_{i=1}^{D} h_i^{\beta} \gamma_i \right)^{\alpha}$$

where D is the number of dimensions of the domain, α and β are parameters of the error model, $h_i(\cdot)$ are the local edge lengths in each dimension, $\gamma_i(\cdot) = |\mathbf{v}_i(\boldsymbol{\xi})^{\mathsf{T}} H(\boldsymbol{\xi}) \mathbf{v}_i(\boldsymbol{\xi})|$ where $(\mathbf{v}_i)_{i=1,\ldots,D}$ are the principal directions of $\mathcal{M}$ and H is the Hessian of the function we are approximating. These γ_i's are positive scalars that reflect the alignment between the directions of the continuous mesh $(\mathbf{v}_i)_{i=1,\ldots,D}$ and the Hessian H_u.

As explained in Section 3.2, We recall that one can define a mesh using the anisotropy ratios r_i (defined in (3.6)) and mesh density d (defined by (3.7)). In these variables, the error model can be formulated as:

$$e_{\mathcal{M}} = d^{-\frac{\alpha\beta}{D}} \left(\sum_{i=1}^{D-1} r_i^{\beta} \gamma_i + P^{-\beta} \gamma_D \right)^{\alpha} . \tag{4.15}$$

This reformulation has the advantage of making the optimization problem a convex one which is decoupled into a global and a local problem.

An important fact in the optimization of the mesh in this case is that the optimization takes place on the stochastic space and that the variables $\xi_i(\omega)$ are random variables pertaining to the probability space $(\Omega_{\boldsymbol{\xi}}, \mathcal{B}, \mathcal{P})$. Consequently, one should take the probability measure into account in the optimization problem. The optimal mesh will be defined as the mesh for which the expectation of the error to the p^{th} power is minimal (this is analogous to minimization in the L^p norm in the deterministic case). This expectation can be worked out further as

$$\mathbf{E}_p = \mathbb{E}[e_{\mathcal{M}}^p] = \int_{\Xi} e_{\mathcal{M}}^p d\mathcal{P} = \int_{\Xi} e_{\mathcal{M}}^p \rho_{\boldsymbol{\xi}} \mathrm{d}\boldsymbol{\xi}. \tag{4.16}$$

After substituting the error model (4.15) into (4.16), the optimal mesh in the $\mathbf{L}^1$ norm can be found by solving the following optimization problem:

$$\mathbf{M}_{\boldsymbol{\xi}}^{opt} = \underset{((r_i)_{i=1,\ldots,D},d)}{\operatorname{argmin}} \int_\Xi d^{-\frac{\alpha\beta p}{D}} \left(\sum_{i=1}^{D-1} r_i^\beta \gamma_i + P^{-\beta}\gamma_D \right)^{\alpha p} \rho_{\boldsymbol{\xi}} \mathrm{d}\boldsymbol{\xi} \text{ subject to } \int_\Xi d \,\mathrm{d}\boldsymbol{\xi} = N_{\boldsymbol{\xi}}. \tag{4.17}$$

4.3.1 Formal resolution

Let scalar $\tilde{\lambda}$ denote a Lagrange multiplier, and $\mathbf{C}$ the constraint, then the augmented Lagrangian for this minimization problem is

$$\begin{aligned} \mathcal{L}(r_1,\ldots,r_D,d,\tilde{\lambda}) &= \mathbb{E}[e_{\mathcal{M}}^p] + \tilde{\lambda}\mathbf{C} \\ &= \int_\Xi d^{-\frac{\alpha\beta p}{D}} \left(\sum_{i=1}^{D-1} r_i^\beta \gamma_i + P^{-\beta}\gamma_D \right)^{\alpha p} \rho_{\boldsymbol{\xi}} \mathrm{d}\boldsymbol{\xi} + \tilde{\lambda}\left(\int_\Xi d\,\mathrm{d}\boldsymbol{\xi} - N \right). \end{aligned} \tag{4.18}$$

The variation of $\mathbf{E}_p$ can be approximated by

$$\begin{aligned} \delta\mathbf{E}_p(\mathbf{M};\delta\mathbf{M}) &= \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\left(\int_\Xi e_{\mathcal{M}+\varepsilon\delta\mathcal{M}}^p - \int_\Xi e_{\mathcal{M}}^p \right) \\ &\approx \int_\Xi \lim_{\varepsilon\to 0} \frac{e_{\mathcal{M}+\varepsilon\delta\mathcal{M}}^p - e_{\mathcal{M}}^p}{\varepsilon} \\ &\approx \int_\Xi \frac{\partial e_{\mathcal{M}}^p}{\partial\mathcal{M}}\delta\mathcal{M}, \end{aligned}$$

while the variation of the constraint will be zero as N is a constant:

$$\delta\mathcal{C}(\mathbf{M};\delta\mathbf{M}) = \lim_{\varepsilon\to 0}\frac{1}{\varepsilon}\left(\int_\Xi (d+\varepsilon d) - \int_\Xi \delta d \right) = \int_\Xi \delta d = 0.$$

As the necessary conditions for a minimum are $\delta\mathcal{L}(r_i;\delta r_i)=0$, $\delta\mathcal{L}(d;\delta d)=0$ and $\delta\mathcal{L}(\lambda;\delta\lambda)=0$, the following optimality conditions can be formulated:

$$\begin{cases} \delta\mathcal{L}(r_i;\delta r_i) &= \int_\Xi \alpha\beta p\, d^{-\frac{\alpha\beta p}{D}} \left(r_i^{\beta-1}\gamma_i - \frac{P^{-\beta}}{r_i}\gamma_D \right) \left(\sum_{j=1}^{D-1} r_j^\beta\gamma_j + P^{-\beta}\gamma_D \right)^{\alpha p-1} \rho_{\boldsymbol{\xi}}\delta r_i \mathrm{d}\boldsymbol{\xi} = 0 \;\forall\delta r_i \\ \delta\mathcal{L}(d;\delta d) &= \int_\Xi \frac{-\alpha\beta p}{D} d^{-\frac{\alpha\beta p+D}{D}} \left(\sum_{j=1}^{D-1} r_j^\beta\gamma_j + P^{-\beta}\gamma_D \right)^{\alpha p} \rho_{\boldsymbol{\xi}}\delta d\mathrm{d}\boldsymbol{\xi} + \tilde{\lambda}\underbrace{\int_\Xi \delta d\mathrm{d}\boldsymbol{\xi}}_{=\delta C(d;\delta d)=0} = 0 \;\forall\delta d \\ \delta\mathcal{L}(\tilde{\lambda};\delta\tilde{\lambda}) &= \underbrace{\left(\int_\Xi d\mathrm{d}\boldsymbol{\xi} - N \right)}_{\mathcal{C}(\mathbf{M})} \delta\tilde{\lambda} = 0 \;\forall\delta\tilde{\lambda} \end{cases} \tag{4.19}$$

The third condition is again the constraint from the original formulation of the optimization problem. By solving for the first two conditions, whilst taking the constraint into consideration, one can find the expressions for the optimal anisotropy ratios $(r_i)_{i=1,\ldots,D}$ and density d.

In the next paragraphs the necessary conditions for the solution will be derived by proposing valid functions for δd and δr_i.

Optimal anisotropy ratios ($\delta\mathcal{L}(r_i;\delta r_i) = 0$)**:** In order to find the optimal anisotropy ratios, choose $\delta d = 0$, and $r_k = 0$ for $k \neq i$. Since for all i the functions d, r_i, γ_i, P, $\rho_{\boldsymbol{\xi}}$ and the constants α, β and p are strictly positive, solving for the first condition comes down to requiring the middle factor to be zero:

$$\delta\mathcal{L}(r_i;\delta r_i) = \int_\Xi \alpha\beta p\, d^{-\frac{\alpha\beta p}{D}} \underbrace{\left(r_i^{\beta-1}\gamma_i - \frac{P^{-\beta}}{r_i}\gamma_D\right)}_{=0} \left(\sum_{j=1}^{n-1} r_j^\beta \gamma_j + P^{-\beta}\gamma_D\right)^{\alpha p-1} \rho_{\boldsymbol{\xi}}\delta r_i \mathrm{d}\boldsymbol{\xi} = 0\ \forall \delta r_i$$

Requiring

$$r_i^{\beta-1}\gamma_i - \frac{P^{-\beta}}{r_i}\gamma_D = 0,$$

yields for r_i

$$r_i = \left(\frac{\gamma_D}{\gamma_i}\right)^{\frac{1}{\beta}} P^{-1} \quad \text{for } i = 1,\dots,D-1.$$

One can now inject the above expression into the definition of $P = \prod_{i=1}^{D-1} r_i$, which yields

$$P = \gamma_D^{\frac{D-1}{\beta}} \left(\prod_{j=1}^{D-1}\gamma_j\right)^{-\frac{1}{\beta}} P^{1-D} \Rightarrow P = \gamma_D^{\frac{1}{\beta}} \left(\prod_{k=1}^{D}\gamma_k\right)^{-\frac{1}{D\beta}} \quad \text{for } i = 1,\dots,D-1. \tag{4.20}$$

Finally, substituting this result back in the expression for r_i obtained previously, one finds the optimal ratios of anisotropy:

$$r_i = \gamma_i^{-\frac{1}{\beta}} \left(\prod_{k=1}^{D}\gamma_k\right)^{\frac{1}{D\beta}} \quad \text{for } i = 1,\dots,D-1 \tag{4.21}$$

Note that the probability density function does not intervene here, the result for the optimal anisotropy ratio is exactly the same as the one found by [Loseille and Alauzet, 2011b].

Optimal density ($\delta\mathcal{L}(d;\delta d) = 0$)**:** Choose $\delta r_i = 0$ for all i, the second condition then becomes:

$$\delta\mathcal{L}(d;\delta d) = \int_\Xi \frac{-\alpha\beta p}{D} d^{-\frac{\alpha\beta p+D}{D}} \left(\sum_{j=1}^{D-1} r_j^\beta\gamma_j + P^{-\beta}\gamma_D\right)^{\alpha p} \rho_{\boldsymbol{\xi}}\delta d \mathrm{d}\boldsymbol{\xi} + \tilde{\lambda} \underbrace{\int_\Xi \delta d \mathrm{d}\boldsymbol{\xi}}_{=\delta C(d;\delta d)=0} = 0\ \forall \delta d \tag{4.22}$$

The second term in (4.22) needs to be zero because the number of samples N is fixed, thus δC should be zero. Therefore, the following two conditions need to be satisfied:

$$\underbrace{\int_\Xi \frac{-\alpha\beta p}{D} d^{-\frac{\alpha\beta p+D}{D}} \left(\sum_{j=1}^{D-1} r_j^\beta \gamma_j + P^{-\beta}\gamma_D \right)^{\alpha p} \rho_{\boldsymbol{\xi}}}_{\frac{\partial e^p_{\mathcal{M}}}{\partial d}\rho_{\boldsymbol{\xi}}} \delta d \mathrm{d}\boldsymbol{\xi} = 0, \;\; \forall \delta d$$

$$\int_\Xi \delta d \mathrm{d}\boldsymbol{\xi} = 0, \quad \forall \delta d.$$

From this we can conclude that $\frac{\partial e^p_{\mathcal{M}}}{\partial d}\rho_{\boldsymbol{\xi}}$ should be constant. Thus there should exist a constant K_1 such that

$$\frac{-\alpha\beta p}{D} d^{-\frac{\alpha\beta p+D}{D}} \left(\sum_{j=1}^{D-1} r_j^\beta \gamma_j + P^{-\beta}\gamma_D \right)^{\alpha p} \rho_{\boldsymbol{\xi}} = K_1.$$

Using (4.20) and (4.21) we can rewrite the expression in the between the parenthesis above as:

$$\begin{aligned}\sum_{j=1}^{D-1} r_j^\beta \gamma_j + P^{-\beta}\gamma_D &= \sum_{j=1}^{D-1} \left[\left(\prod_{k=1}^{D} \gamma_k \right)^{\frac{1}{D\beta}} \gamma_j^{\frac{-1}{\beta}} \right]^\beta \gamma_j + \left[\gamma_D^{\frac{1}{\beta}} \left(\prod_{k=1}^{D} \gamma_k \right)^{\frac{-1}{D\beta}} \right]^{-\beta} \gamma_D \\ &= \sum_{j=1}^{D} \left(\prod_{k=1}^{D} \gamma_k \right)^{\frac{1}{D}} = D \left(\prod_{k=1}^{D} \gamma_k \right)^{\frac{1}{D}}.\end{aligned}$$

This means that d should verify

$$d = K_2 \left(\prod_{j=1}^{D} \gamma_j \right)^{\frac{\alpha p}{\alpha\beta p+D}} \rho_{\boldsymbol{\xi}}^{\frac{D}{\alpha\beta p+D}}. \tag{4.23}$$

We can now inject the above expression for d into the constraint and solve for the constant K_2:

$$C(\mathbf{M}) = \int_\Xi d \, \mathrm{d}\boldsymbol{\xi} = K_2 \int_\Xi \left(\prod_{j=1}^{D} \gamma_j \right)^{\frac{\alpha p}{\alpha\beta p+D}} \rho_{\boldsymbol{\xi}}^{\frac{D}{\alpha\beta p+D}} \mathrm{d}\boldsymbol{\xi} = N,$$

from which we deduce for K_2 the following expression:

$$K_2 = N \left(\int_\Xi \left(\prod_{j=1}^{D} \gamma_j \right)^{\frac{\alpha p}{\alpha\beta p+D}} \rho_{\boldsymbol{\xi}}^{\frac{D}{\alpha\beta p+D}} \mathrm{d}\boldsymbol{\xi} \right)^{-1}.$$

We can now obtain the optimal density d by substituting the above result into (4.23):

$$d = N \left(\int_\Xi \left(\prod_{j=1}^{D} \gamma_j \right)^{\frac{\alpha p}{\alpha\beta p+D}} \rho_{\boldsymbol{\xi}}^{\frac{D}{\alpha\beta p+D}} \mathrm{d}\boldsymbol{\xi} \right)^{-1} \left(\prod_{j=1}^{D} \gamma_j \right)^{\frac{\alpha p}{\alpha\beta p+D}} \rho_{\boldsymbol{\xi}}^{\frac{D}{\alpha\beta p+D}}. \tag{4.24}$$

Optimal solutions Now that the optimal anisotropy ratio's $(r_i)_{i=1,\dots,D}$ have been found, the expressions for the optimal h_i and λ_i (the optimal edge lengths) can be deduced. From these, one can then derive the expressions for the optimal metric $\mathcal{M}^{opt}$ and the error estimate on the optimal mesh. The details of this operation are similar to the optimal goal-oriented metric given in [Loseille, 2008], and the results are summarized below.

Optimal weighted metric when $\alpha = 1, \beta = 2, p = 1$ is found to be

$$\mathcal{M}^{opt} = N^{\frac{2}{D}} \left(\int_{\Xi} (\det(\rho_{\boldsymbol{\xi}} |H|)^{\frac{1}{2+D}} \, \mathrm{d}\boldsymbol{\xi} \right)^{-\frac{2}{D}} (\det(\rho_{\boldsymbol{\xi}} |H|))^{\frac{-1}{2+D}} \, |\rho_{\boldsymbol{\xi}} H|, \tag{4.25}$$

and the expectation of the continuous interpolation error is given by

$$\mathbb{E}[e_{\mathcal{M}^{opt}}] = D N^{-\frac{2}{D}} \left(\int_{\Xi} \left(\prod_{j=1}^{D} \rho_{\boldsymbol{\xi}} \gamma_j \right)^{\frac{1}{2+D}} \mathrm{d}\boldsymbol{\xi} \right)^{\frac{2+D}{D}} . \tag{4.26}$$

Similar results to (4.25) and (4.26) have been found by [Loseille, 2008] in the deterministic context a of goal-oriented optimal metric. Here however, it is the probability density function $\rho_{\boldsymbol{\xi}}$ that acts as a weighting instead of an adjoint state related to the quantity of interest in the goal-oriented case.

This allows us to exploit the existing numerical tools for both the deterministic and stochastic metric-based mesh adaptation. In this thesis, `METRIX` [Alauzet and Loseille, 2009] will be used for the computation of the metric $\mathcal{M}_{h_{\boldsymbol{\xi}}}$ while `FEFLO` [Loseille and Löhner, 2010] will be used to generate a discrete mesh $\mathcal{H}_{h_{\boldsymbol{\xi}}}$ form the optimal metric. In this process, estimate (4.26) will be used as indication for the estimated interpolation error.

4.4 Stochastic adaptive strategy

In practice, the problem of deriving an optimal mesh is a nonlinear problem that is solved iteratively using a fixed-point loop in order to converge the couple (mesh, solution).

In a deterministic context, this iterative procedure means that at each fixed-point iteration, a new mesh with a new topology is built. However, for computational cost reasons, it is obvious that in the stochastic context we cannot afford to not reuse information already available. Thus, once a sample has been added to the mesh (this corresponds to a new CFD computation), it will be kept throughout all subsequent iteration loops. The adaptation process used to control the stochastic error $\mathbb{E}[e_{\mathcal{M}}]$ is detailed in Algorithm 1, each cycle of the algorithm corresponds to the resolution of one optimisation problem with a fixed number of required samples. The notation used in Algorithm 1 is as follows: the iteration count index will be l and will appear as a subscript. As such $\{\boldsymbol{\xi}\}_l$ is the set of samples at iteration l, $\mathcal{H}_{\xi,l}$ is the discrete mesh on the parametric domain at iteration l containing $N_{\xi,l}$ vertices. The superscript t indicates a target quantity, for example $\mathcal{C}^t_{\xi}$ is the target complexity of the mesh on the parametric domain at iteration l. This target complexity is passed on to the meshing code which, based on the computed optimal metric $\mathcal{M}^{opt}$, constructs a discrete mesh with the specified complexity.

4.4.1 Simplex-Stochastic Collocation

Since the error estimate used to drive mesh adaptation is based on the interpolation error resulting from a linear interpolation, a piecewise linear approximation will be used as the

Algorithm 1 Optimisation of stochastic discretization

```
Generate (or read) initial samples ${\boldsymbol{\xi}}_0$ and form initial mesh $\mathcal{H}_{\xi,0}$ containing $N_{\xi,0}$ vertices by a Delaunay triangulation.
Compute $j_0(\{\boldsymbol{\xi}\}_0)$ for this DoE.
Set target complexity $\mathcal{C}^t_\xi = N^t_\xi > N_{\xi,0}$.
for $l = 1$ to $n_{adap}$ do
    Compute optimal metric $\mathcal{M}^{opt}_{\xi,l}$ for complexity $\mathcal{C}^t_\xi = N^t_\xi$ based numerical approximation constructed on previous mesh $\mathcal{H}_{\xi,l-1}$.
    Generate new mesh $\mathcal{H}_{\xi,l}$ from $\mathcal{M}^{opt}_{\xi,l}$ containing $N_{\xi,l} = N_{\xi,l-1} + N^{new}_{\xi,l}$ samples.
    Run deterministic solver to compute QoI at the $N^{new,l}_\xi$ new samples and update $j(\{\boldsymbol{\xi}\}_l)$.
    Compute the statistical moments of the surrogate model.
end for
```

surrogate model. In the following, a brief outline of a baseline version of the Simplex (elements) Stochastic Collation (SSC) approximation will be provided. This technique was greatly developed and extended by Witteveen *et al.* in the context of robust adaptive uncertainty quantification, see for instance [Witteveen, Loeven, and Bijl, 2009; Witteveen and Iaccarino, 2012a; Witteveen and Iaccarino, 2013b; Witteveen and Iaccarino, 2013a].

It relies on a piecewise multivariate polynomial approximation of a stochastic response j that depends on a vector of random parameters $\boldsymbol{\xi} \in \Xi \subset \mathbb{R}^{D_{\boldsymbol{\xi}}}$, i.e. $j(\boldsymbol{\xi})$. The SSC method discretizes the parameter space into a tessellation of N^{elem}_Ξ simplices.

On a linear Lagrange element $\Xi_{(i)}$ the continuous representation of the response surface is obtained by an interpolation on N_q points; we define the interpolation operator on element $\Xi_{(i)}$:

$$\mathcal{I}_{\Xi_{(i)}} j(\boldsymbol{\xi}) = \sum_{k=1}^{N_q} j(\boldsymbol{\xi}_{(k)}) b_k(\boldsymbol{\xi}),$$

where b_k are the barycentric coordinates of $\boldsymbol{\xi}$ locally in $\Xi_{(i)}$. The global interpolant on mesh $\mathcal{H}_\xi$ can then be written as

$$\mathcal{I}_{\mathcal{H}_\xi} j(\boldsymbol{\xi})|_{\Xi_{(i)}} = \mathcal{I}_{\Xi_{(i)}} j(\boldsymbol{\xi}).$$

The tessellation of the parametric space decomposes integrals over the parametric space into a summation of integrals over N^{elem}_Ξ simplices. Let $\Omega_{\xi,(i)}$ be the i^{th} such element and $\Xi_{(i)}$ its image in the parameter space. The integrals over these elements will be approximated by a Newton-Cotes (NC) quadrature using N_q quadrature points. The quadrature weights of these integrals over the simplex domains however will need to take into account the probability measure. For simplicity, N_q will be assumed to be the same in each element. Hence the expectation will be approximated as

$$\begin{aligned} \mathbb{E}[j(\boldsymbol{\xi})] &= \int_\Xi j(\boldsymbol{\xi}) \rho_{\boldsymbol{\xi}} \mathrm{d}\boldsymbol{\xi} \\ &= \sum_{i=1}^{N^{elem}_\Xi} \int_{\Xi_{(i)}} j(\boldsymbol{\xi}) \rho_{\boldsymbol{\xi}} \mathrm{d}\boldsymbol{\xi} \\ &\approx \sum_{i=1}^{N^{elem}_\Xi} \sum_{k=1}^{N_q} c_{(i,k)} j(\boldsymbol{\xi}_{(i,k)}), \end{aligned} \tag{4.27}$$

where $\boldsymbol{\xi}_{(i,k)}$ and $c_{(i,k)}$ are the parameter value and the quadrature weight corresponding to the k^{th} quadrature point in the i^{th} element. For a given pdf and tessellation, quadrature weights are computed once as the integrals of the Lagrange basis weighted by the pdf:

$$c_{(i,k)} = \int_{\Xi_{(i)}} \mathcal{L}_{(i,k)}(\xi_1, \xi_2, \ldots, \xi_{D_{\boldsymbol{\xi}}}) \rho_{\boldsymbol{\xi}}(\xi_1, \xi_2, \ldots, \xi_{D_{\boldsymbol{\xi}}}) \mathrm{d}\boldsymbol{\xi} \quad \text{with } i = 1, \ldots, N_{\Xi}^{elem}, \tag{4.28}$$

where $\mathcal{L}_{(i,k)}(\boldsymbol{\xi})$ is the Lagrange polynomial in the i^{th} element corresponding to the k^{th} quadrature point evaluated at $\boldsymbol{\xi}$. In this work, first degree NC quadrature rule is used. This corresponds to one quadrature/sample point at each vertex of the $D_{\xi}-$simplex. In that case $L_{(i,k)}$ are linear Lagrange polynomials on element $\Xi_{(i)}$, taking value 1 on vertex j and 0 on all other vertices.
As proposed in [Witteveen, Loeven, and Bijl, 2009], a higher degree quadrature with standard weights is used in order to compute the integral in (4.28). Let $q_{(i,l)}$ be the standard NC quadrature weight from a quadrature rule containing $N_{q_{sub}}$ points, then the weights $c_{(i,k)}$ can be computed as

$$c_{(i,k)} \approx \sum_{l=1}^{N_{q_{sub}}} q_{(i,l)} \mathcal{L}_{(i,k)}(\boldsymbol{\xi}_{(i,l)}) \rho_{\boldsymbol{\xi}}(\boldsymbol{\xi}_{(i,l)}). \tag{4.29}$$

In Figure 4.1 these linear Lagrange polynomials have been drawn for the $2D$ case where the stochastic space has been decomposed into 4 triangles.

For the fine quadrature on which the weights $c_{(i,k)}$ will be computed, NC quadratures up to degree 8 have been implemented in $2D$ and up to degree 6 in $3D$. Figures 4.2 and 4.3 demonstrate these quadratures on a reference element. One can easily transform this reference element and the quadrature points on it to the element $\Xi_{(i)}$ in the parametric space, hence the linear Lagrange polynomials only need to be computed once on the fine quadrature points, thereafter one needs to compute for each element the transformation matrix to go from the reference element to the actual element $\Xi_{(i)}$ in the parameter space.

It should however be noted that on the $2D$ simplices, the Newton-Cotes quadrature weights of degrees 4, 6, 7 and 8 contain negative weights while on the $3D$ tetrahedra, quadratures of degrees 2, 4, 5 and 6 contain negative weights. These particular quadrature rules are not guaranteed to be stable. It is a well-known drawback of Newton-Cotes quadratures that the use of larger numbers of equally spaced points can produce ill-conditioned quadrature rules with negative weights. Do note that these higher degree quadratures are not needed to integrate the linear Lagrange polynomials but merely account for the complexity introduced by the local probability density functions. In section 4.5.1, where the results on test problems will be presented, it will be verified that the approximation error present in the quadrature weights becomes small enough not to impact the results as the number of elements increases.

The non-intrusive SSC method is used in our framework as a robust and efficient stochastic collocation approximation. It is based on an unstructured tessellation rather than relying on the more common tensor product structure and allows for anisotropic refinements. Moreover, it is capable of handling non-hypercube probability spaces [Witteveen and Iaccarino, 2012b]. Note that our approximation differs slightly form the SSC method proposed in [Witteveen, Loeven, and Bijl, 2009]. In [Witteveen, Loeven, and Bijl, 2009] a second degree Newton-Cotes is used resulting in a piecewise quadratic approximation of the stochastic response. Furthermore, in order to avoid unphysical oscillations near singularities, the original SSC method splits quadratic elements into smaller first degree elements when an extremum is detected that is not located at one of the quadrature

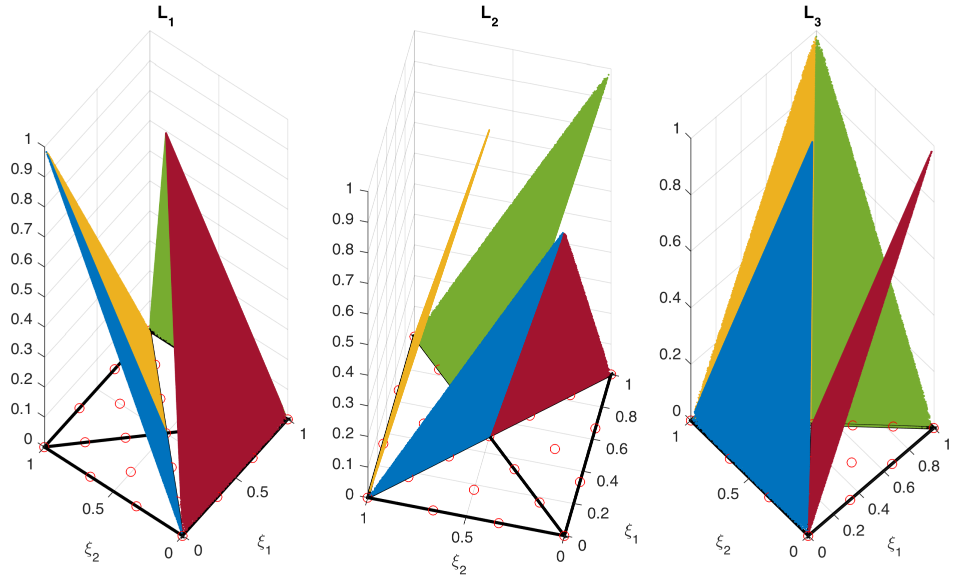


FIGURE 4.1: Lagrange basis of degree 1 plotted on four elements. On the left, the Lagrange basis function corresponding to the first vertex of each element is drawn in red (element 1), green (element 2), yellow (element 3) and blue (element 4). In the middle frame the Lagrange basis function corresponding to the second vertex is drawn (same colours as before) and on the right the Lagrange basis functions corresponding to the third vertex is presented (colours as before). The open red circles show the higher degree quadrature on which $\rho_{\boldsymbol{\xi}}$ will be evaluated in order to compute the integration weights corresponding to the three vertices of each element. In this example the fine quadrature is a third degree Newton-Cotes quadrature, but quadratures up to degree 8 have been implemented in 2D and up to degree 6 in 3D.

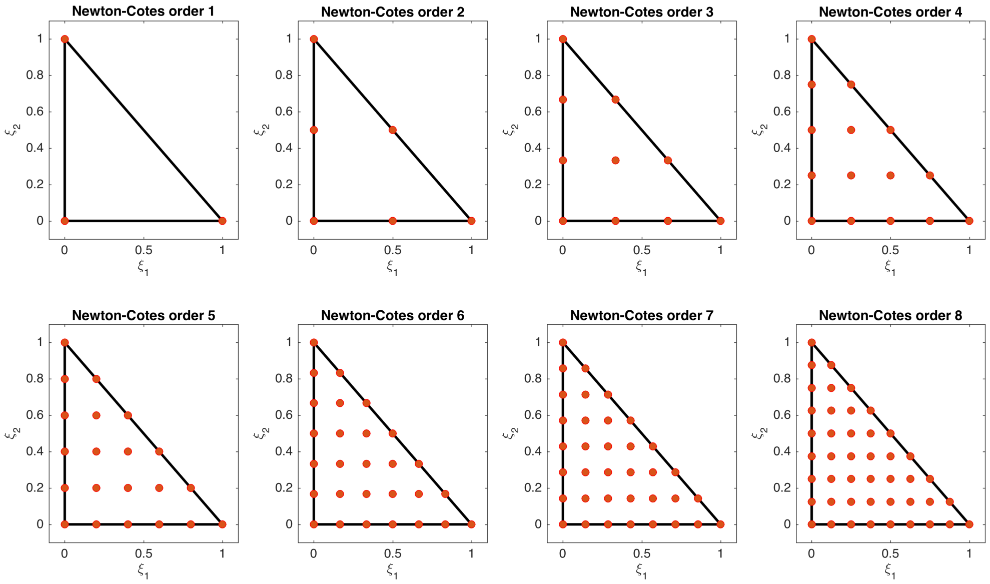


FIGURE 4.2: Newton-Cotes quadrature points on the 2D reference element.

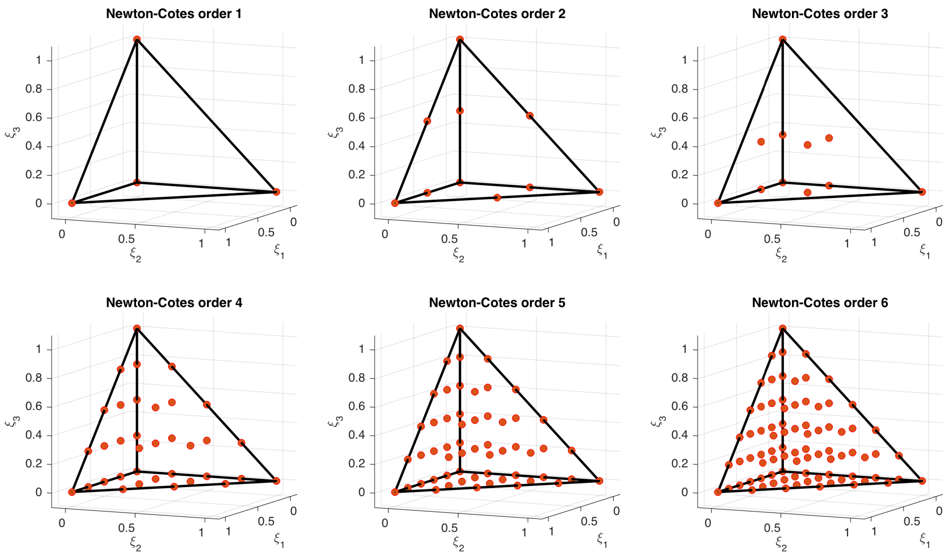


FIGURE 4.3: Newton-Cotes quadrature points on the 3D reference element.

points. In later publications other refinement criteria were investigated [Witteveen and Iaccarino, 2012a] as well as more elaborate schemes in order to obtain truly discontinuous representations [Witteveen and Iaccarino, 2013b; Witteveen and Iaccarino, 2013a].

We use only linear elements and rely only on $h-$adaptivity to control the error. The $h-$adaptivity used in this work is also very different from the one applied in the SSC method which splits existing elements according to a predefined pattern. In this work, we use a metric-based approach to drive anisotropic $h-$adaptivity, allowing for far greater flexibility.

4.5 Numerical Applications

In the previous sections, links between metric-based mesh adaptation in the stochastic space and deterministic goal-oriented mesh adaptation were pointed out. In the former, we have noticed how the probability density function plays a role similar to the adjoint solution in the latter. The deterministic version of this mesh adaptation technique has already been validated in [Loseille and Alauzet, 2011a; Loseille and Alauzet, 2011b]. In this section the effectiveness of metric-based mesh adaptation will be demonstrated on some test cases. In particular, we will emphasize the impact of the dimensionality and the parameter probability density functions on the effectiveness of the adaptive approximations of sensitive nonlinear functionals.

4.5.1 Validation of stochastic test functions

The first test function treated is one with multiple curved and straight discontinuities proposed by [Jakeman, Narayan, and Xiu, 2013]. A graphic representation is presented in

Figure 4.4 while its analytic form is given by

$$y(\boldsymbol{\xi}) = \begin{cases} f_1(\boldsymbol{\xi}) - 2 & \text{if } 3\xi_1 + 2\xi_2 \geq 0 \text{ and } -\xi_1 + 0.3\xi_2 < 0, \\ 2f_2(\boldsymbol{\xi}) & \text{if } 3\xi_1 + 2\xi_2 \geq 0 \text{ and } -\xi_1 + 0.3\xi_2 \geq 0, \\ 2f_1(\boldsymbol{\xi}) + 4 & \text{if } (\xi_1+1)^2 + (\xi_2+1)^2 < 0.95^2 \text{ and } D_\xi = 2, \\ f_1(\boldsymbol{\xi}) & \text{otherwise.} \end{cases} \tag{4.30}$$

Furthermore, f_1 and f_2 are given by

$$f_1(\boldsymbol{\xi}) = \exp\left(-\sum_{i=1}^{2} \xi_i^2\right) - \xi_1^3 - \xi_2^3,$$

$$f_2(\boldsymbol{\xi}) = 1 + f_1(\boldsymbol{\xi}) + \frac{1}{4D_{\boldsymbol{\xi}}} \sum_{i=2}^{D_{\boldsymbol{\xi}}} \xi_i^2.$$

The domain on which this function will be solved is $[-1, 1]^{D_{\boldsymbol{\xi}}}$.

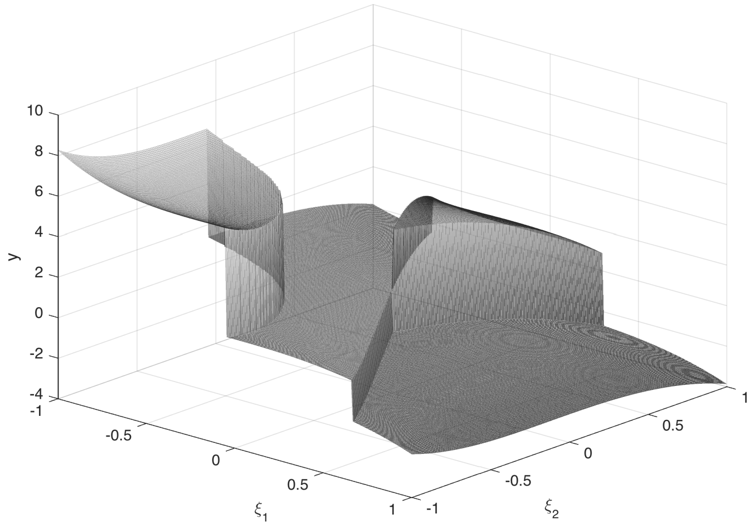


FIGURE 4.4: Test function with multiple discontinuities.

Test 1: uniform probability density function in two dimensions

Figure 4.5 shows the mesh at different mesh refinement steps on a 2D domain when both ξ_1 and ξ_2 follow a uniform probability distribution $\mathcal{U}_{[-1,1]}$.

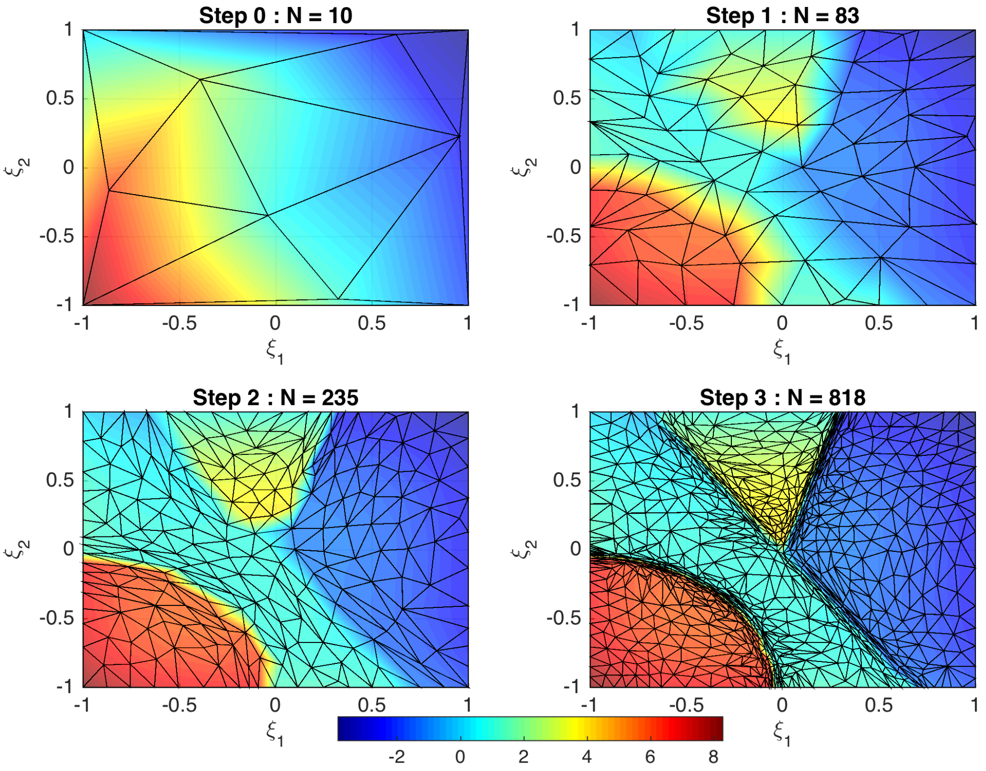
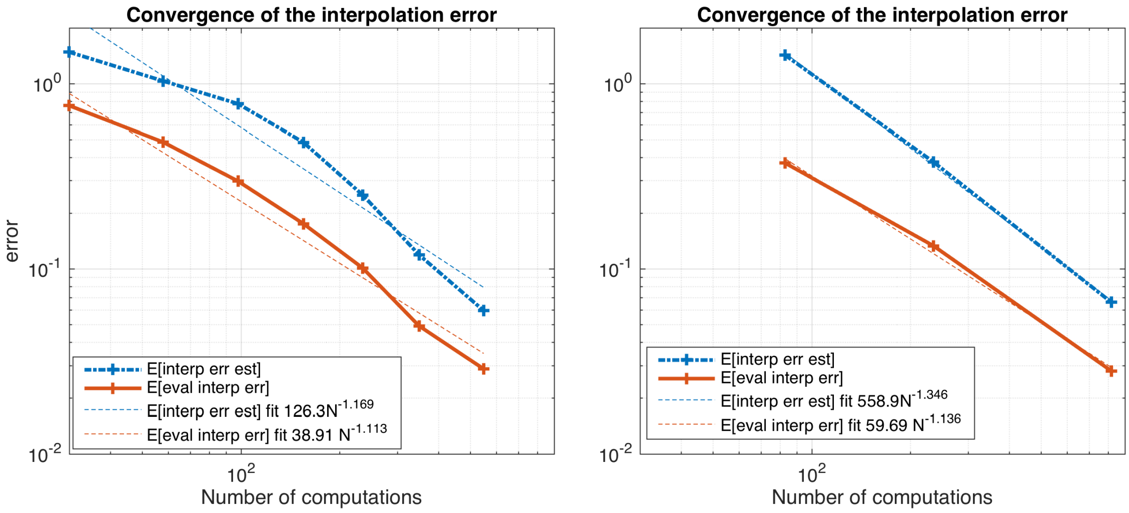


FIGURE 4.5: Approximated solution on the mesh at each refinement.

The initial mesh shown in Figure 4.5 (top left) contains 10 samples which were randomly drawn using Latin Hypercube Sampling (LHS) DoE following a uniform law. Three subsequent adaptation steps were done: first increasing from 10 to 83 samples, then to 235 samples and finally up to 818 samples. One can observe that the initial mesh contains very little information as to the whereabouts of the discontinuities, the first adapted mesh containing 83 samples starts showing a moderate amount of alignment with the discontinuities. It can be observed in the lower left quadrant of the top right figure that the mesh follows the arc-shaped discontinuity in a very coarse way and similarly, in the upper right quadrant, one can see the beginnings of mesh alignment with the discontinuity there. Mesh alignment becomes much clearer after the second adaptation step, in the mesh shown on the lower left in Figure 4.5. Sample point density is significantly higher in the vicinity of the discontinuities leading to a more accurate approximation in those regions. Finally, after the third adaptation step, a mesh with 818 samples is obtained where de discontinuities are clearly visible by the even higher sample point density, making a sharp resolution of the discontinuities possible. Note that no discontinuity capturing technique is used here, nor are there any parameters that need to be set; the results are obtained by running the metric-based mesh adaptation *out-of-the-box* on this test problem.



(A) 8 refinement steps (B) 3 refinement steps

FIGURE 4.6: Convergence of the interpolation error and the error estimate. The interpolation error is computed on a third degree Newton-Cotes quadrature, a sort of subgrid constructed in order to evaluate the real interpolation error.

Since this is an analytic test function, one can easily evaluate the expectation of the exact error committed by the response surface approximation. The convergence of the exact error and the error estimate are shown in Figure 4.10 for two separate cases. Both figures (A) and (B) started with the same initial mesh containing 10 samples. The difference is that in Fig. 4.10 (A) 8 small refinement steps were used (each time doubling the mesh complexity) while in Figure 4.10 (B) only 3 steps were taken (at each step the mesh complexity was increased by a factor of 5.5). This allows to evaluate the influence of the step size on the convergence of the error.

The theoretical error convergence is given by $\beta N^{\frac{2}{D_\xi}}$, which would mean that in the $2D$ case, one should observe a convergence rate $\mathcal{O}(N^{-1})$. It can be seen in Figure 4.10 that this convergence rate is surpassed in both cases. Furthermore it can be seen that the convergence rate is not significantly altered when the adaptation step size is significantly increased, the convergence constant of the error estimator does change significantly. The expectation of the *evaluated interpolation error* shown in Figure 4.10 is computed on a finer NC quadrature within each element. The convergence constant will both depend on the function that is being approximated and on the discrepancy between the continuous number of mesh vertices $\mathcal{N}$ and the number of vertices in the discrete mesh N. In the computation of the continuous interpolation error estimate, the continuous number of vertices is used, while in Figure 4.10 it is plotted as a function of the realized number of vertices in the discrete mesh N. For a detailed numerical validation between the continuous error estimate and the actual interpolation error, the reader is referred to [Loseille and Alauzet, 2011b].

Since the initial mesh is the Delaunay triangulation of randomly drawn sample points, it is a good idea to check the influence of this initial mesh on the error convergence properties. The test case with 3 refinement steps was run 50 times, each time with a different randomly drawn initial mesh. From each run results a convergence curve similar to Figure 4.6b, and for each of these a least squares fit to the model βN_ξ^κ is performed; the parameters are called β_{est} and κ_{est} for the interpolation error estimate and β_{eval} and κ_{eval} for the

$N_{\xi,0}$		μ	σ	$CoV = \frac{\sigma}{\lvert\mu\rvert}\cdot 100\%$
10	κ_{est}	-1.188	0.1221	10.27%
10	β_{est}	258.8	213.5	82.49%
10	κ_{eval}	-1.108	0.0563	5.09%
10	β_{eval}	51.05	17.66	34.58%
30	κ_{est}	-1.242	0.0881	7.10%
30	β_{est}	331.0	212.7	64.26%
30	κ_{eval}	-1.143	0.0644	5.63%
30	β_{eval}	67.37	29.17	43.30%

TABLE 4.1: A least squares fit of the error convergence to the model βN_ξ^κ is performed. For the convergence curve of the interpolation error estimate, the parameters are called β_{est} and κ_{est}, for the evaluated interpolation error they are called β_{eval} and κ_{eval} (interpolation errors were evaluated on a NC subgrid quadrature of degree 5). Both the average and standard deviation after 50 repetitions are given for the case when the initial random mesh is composed of $N_{\xi,0} = 10$ samples and for the case where $N_{\xi,0} = 30$.

evaluated interpolation error. The evaluation of the interpolation error was done on a NC subgrid of degree 5 in each element. In Table 4.1 the average and standard deviation as well as the Coefficient of Variation (CoV) for the parameters β and κ is given.

Additionally, the histogram of the convergence constant β and convergence exponent κ for all 50 runs is given in Figure 4.7.

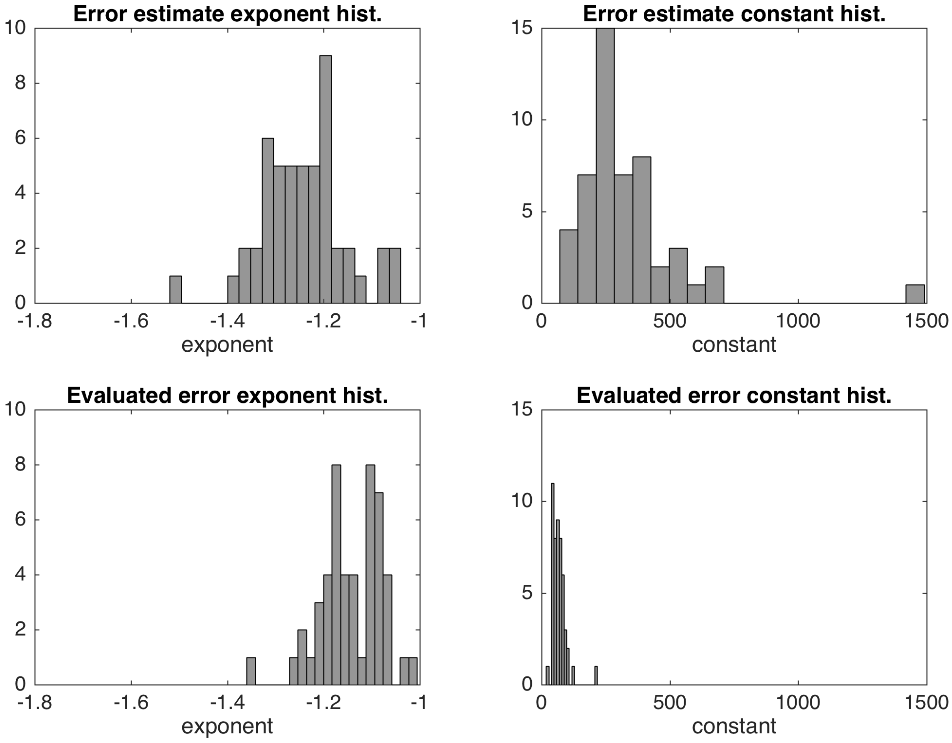


FIGURE 4.7: Histogram of the exponent κ and constant β for 50 runs of the test problem with $N_{\xi,0} = 30$. For each run the convergence of the error estimate and the actual evaluated error is fitted to βN_ξ^κ, the histograms show the distribution of the parameters β and κ.

While 50 trials may not be enough to have statistically significant conclusions on the influence of the initial mesh on the error convergence, it does provide important information to this end. First of all, it is reassuring to see that there are no trials where the convergence rate of error is slower than the theoretical $\kappa = -1$. The histogram of the exponent κ for both the evaluated interpolation error (κ_{eval}) and the interpolation error estimate (κ_{est}) have a similar, vaguely bell-shaped form, all be it centered around different values. The influence of the initial mesh on the convergence rate can be called modest, with the CoV of the order of 10%. It is interesting to note that the CoV of κ_{est} is higher than the one of κ_{eval}, with the former being double the latter in the case where $N_{\xi,0} = 10$.
The initial mesh has a much larger effect on the convergence constant β however. Their CoV is between 43% and 82%, but care should be taken with the interpretation of CoV and standard deviation in this case because, as can be seen from the histograms on the right in Figure 4.7, the distributions are skewed. As for the size of the initial mesh, $N_{\xi,0} = 10$ or $N_{\xi,0} = 30$, the convergence rate is slightly better in the case of $N_{\xi,0} = 30$.

Test 2: discontinuous probability density function in two dimensions

The same test function treated previously will now be used again, but in order to shown the effect of the underlying probability density function, instead of the uniform distribution used previously, now a discontinuous probability density function will be used:

$$\rho_\xi = \begin{cases} \frac{1-(2.6-\frac{1}{2}0.2^2\pi)0.005-(\frac{1}{2}0.2^2\pi)0.9}{1.4} \approx 0.66 & \text{if } \xi_2 \geq -0.3\xi_1 + 0.3, \\ 0.9 & \text{if } \xi_1^2 + (\xi_2+1)^2 \leq 0.2^2, \\ 0.005 & \text{otherwise.} \end{cases} \tag{4.31}$$

The method presented in this manuscript does not require an orthogonal basis to be found with respect to the probability density function. This is an advantage over the ME-gPC method where finding an orthogonal basis in each stochastic element locally orthogonal to the pdf can be a nuisance. This is a practical advantage that can be useful when random variable parametrizing the problem do not follow some well-known distribution but rather follow an arbitrary distribution obtained from experiments for example.

In Figure 4.8 the colour contours visualise this discontinuous probability density function while the mesh shown is the result of three subsequent adaptation steps. The intermediate meshes and the approximation on these meshes is shown in Figure 4.9.

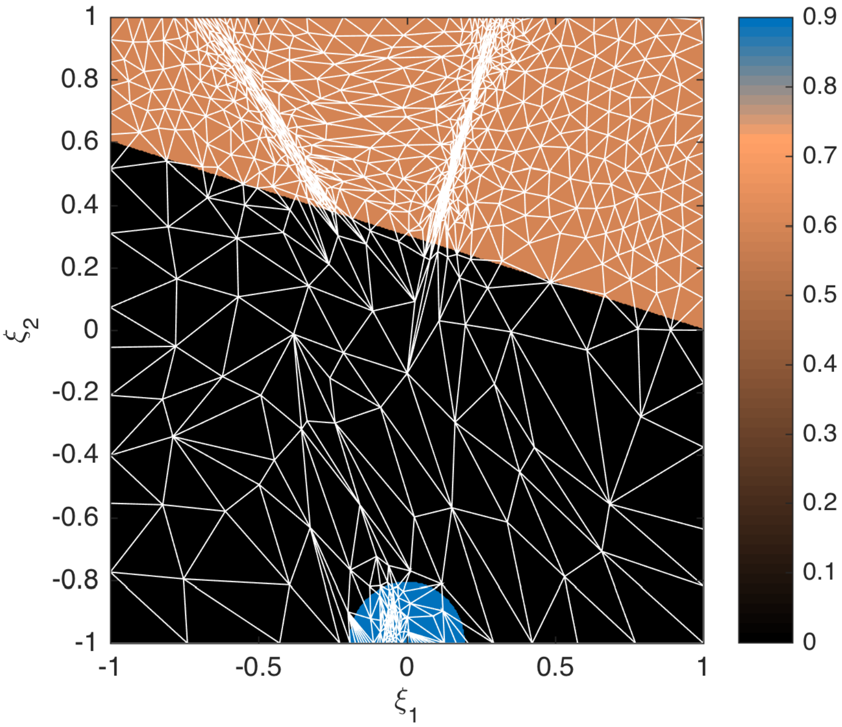


FIGURE 4.8: The mesh plotted is the one obtained at the end of three iteration steps and for which the interpolation error is shown in 4.10b. The color background represents the discontinuous probability density function.

The mesh shown in Figure 4.8 clearly shows how the probability distribution is taken into account in the mesh refinement. In regions with low event probability, even if there are discontinuities, few sample points will be placed. In regions with high event probability, more sample points will be placed and the sample point density is especially high in the vicinity of the discontinuities.

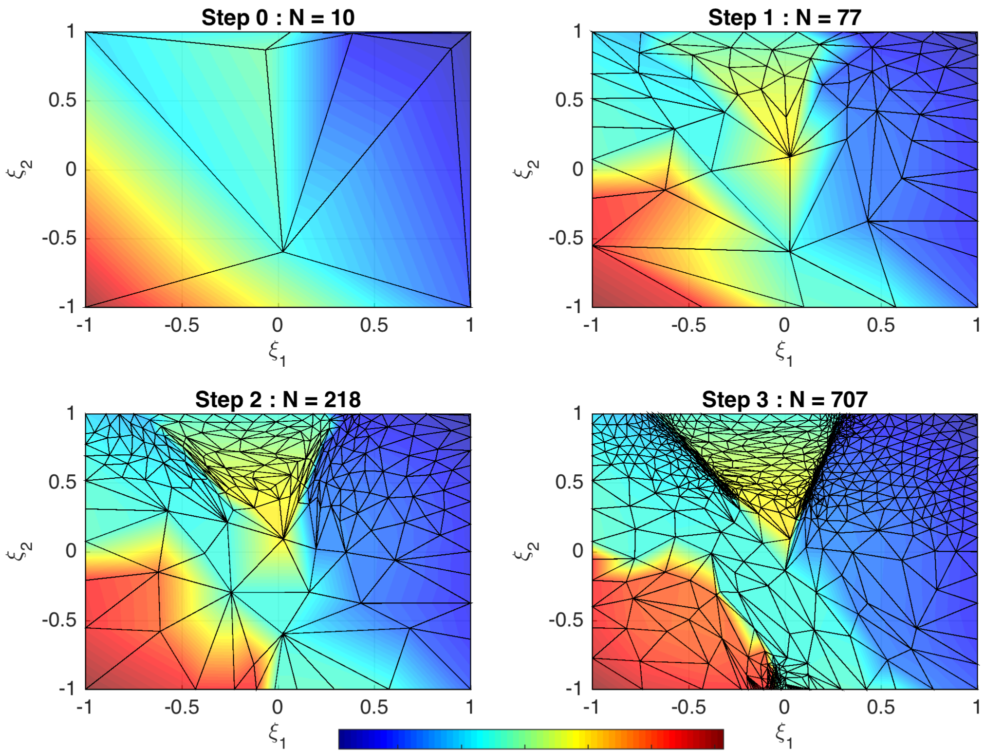
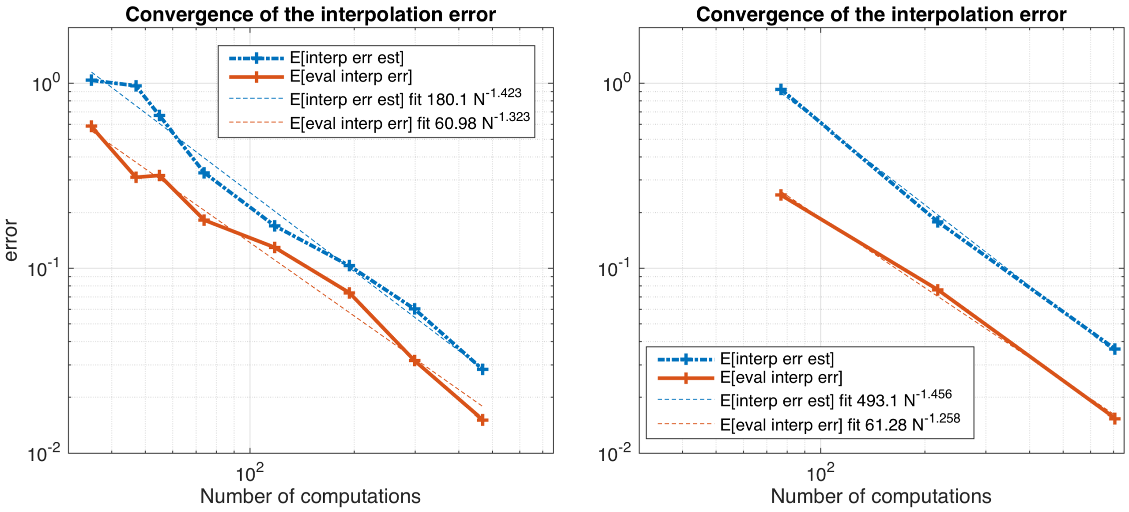


FIGURE 4.9: Approximated solution on the mesh at each refinement step where the discontinuous probability density function is taken into account.

In the evolution of the mesh adaptation process shown in Figure 4.9, it is interesting to note that the initial mesh contains no sample points in the high-probability half circle at the bottom of the domain. Still, the importance of this region is automatically detected.



(A) 8 refinement steps

(B) 3 refinement steps

FIGURE 4.10: Convergence of the interpolation error and the error estimate. The interpolation error is computed on a third degree Newton-Cotes quadrature, a sort of subgrid constructed in order to evaluate the real interpolation error.

$N_{\xi,0}$		μ	σ	$CoV = \frac{\sigma}{\lvert\mu\rvert} \cdot 100\%$
10	κ_{est}	-1.601	0.2112	13.19%
10	β_{est}	2.320	4.049	174.53%
10	κ_{eval}	-1.458	0.1387	9.515%
10	β_{eval}	3.277	2.885	88.06%
30	κ_{est}	-1.470	0.1340	9.11%
30	β_{est}	7.173	6.047	84.29%
30	κ_{eval}	-1.389	0.1114	8.02%
30	β_{eval}	2.023	1.474	72.86%

TABLE 4.2: Similar to Table 4.1 except that now the underlying pdf is the discontinuous pdf given by (4.31).

Similarly to the previous case, the adaptation process was executed with different step sizes. The resulting convergence of both the expectation of the actual interpolation error and the expectation of the error estimate are shown in Figure 4.10. The observations made for the case of uniform distribution are confirmed here: the convergence rate is at least what was predicted by theory and is not significantly affected when the step size is drastically increased. The convergence constant of the error estimate does increase when the step size is increased.

As was done with the previous test case, again here the numerical experiment with 3 refinement steps was repeated 50 times, each time with a different initial mesh with the samples drawn according to the discontinuous pdf. A first series of 50 runs was done whose initial meshes contained 10 vertices, and in a second series of 50 runs the initial meshes had 30 vertices. An overview of the mean, standard deviation and CoV of the convergence constant β and convergence exponent κ is given in Table 4.2 and the histograms of β_{est}, β_{eval}, κ_{est} and κ_{eval} for the case where $N_{\xi,0}$ is shown in Figure 4.11.

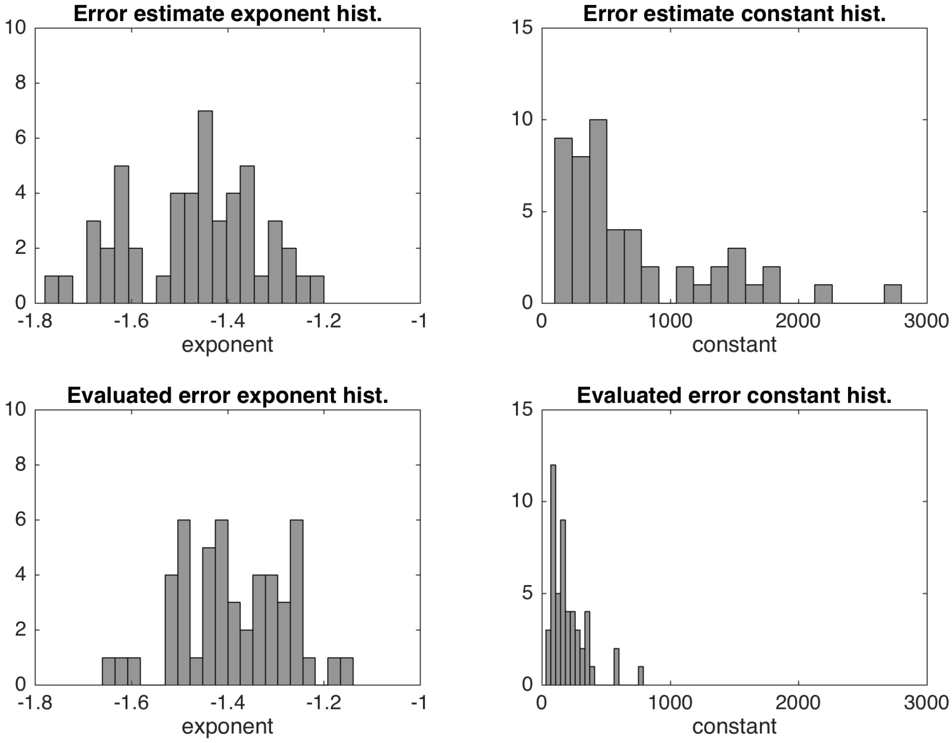


FIGURE 4.11: Histogram of the exponent κ and constant β for 50 runs of the test problem with $N_{\xi,0} = 30$; similar to Figure 4.7.

From both Table 4.2 and Figure 4.11 one comes to the same conclusions as for the case with uniform pdf: the convergence rate is only moderately influenced by the initial mesh while the effect on the convergence constant is much higher. As before, the influence of the initial mesh on the convergence rate of the evaluated interpolation error is lower than the influence on the convergence rate of the interpolation error estimate. Equally, the convergence rate is slightly higher when the initial mesh contains more samples.

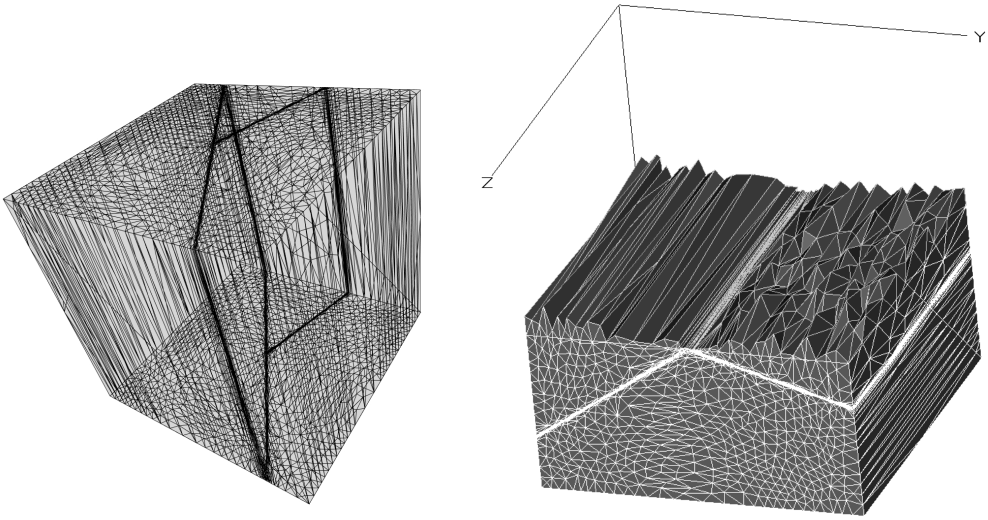
Accuracy of quadrature weights As an indication for the magnitude of the error contained in the approximation (4.29), we look at the sum of the quadrature weights of all elements. The total should equal exactly 1. For current test case (discontinuous response and discontinuous pdf) a table is presented below collecting the sum of the quadrature weights on the meshes shown in Figure 4.9 for different degrees of subgrid NC quadratures. From Table 4.3 one can see that the error committed in approximating the quadrature weights is significant on meshes with very few elements. However, with increasing quadrature degree and increasing N_ξ, this error quickly drops to $\mathcal{O}(10^{-3})$ or less.

Test 3: uniform distributions in three dimensions

The case of three uncertain variables is also examined for which a cutout of the $3D$ mesh is shown in Figure 4.12.

Subgrid NC deg.	$N_\xi = 10$	$N_\xi = 77$	$N_\xi = 218$	$N_\xi = 707$
1	1.209	1.195	1.024	0.996
2	0.589	0.986	0.984	1.007
3	1.199	0.984	1.003	1.002
4	0.965	0.980	1.004	1.002
5	1.012	0.999	1.001	0.999
6	0.856	1.002	0.993	1.000
7	0.965	1.002	1.006	1.000
8	1.073	1.003	1.000	1.000

TABLE 4.3: Overview of the sum of the quadrature weights for the discontinuous pdf (4.31) on the meshes in Figure 4.9. The subgrid NC quadrature degree used to compute the weights according to (4.29) is indicated in the column on the left.

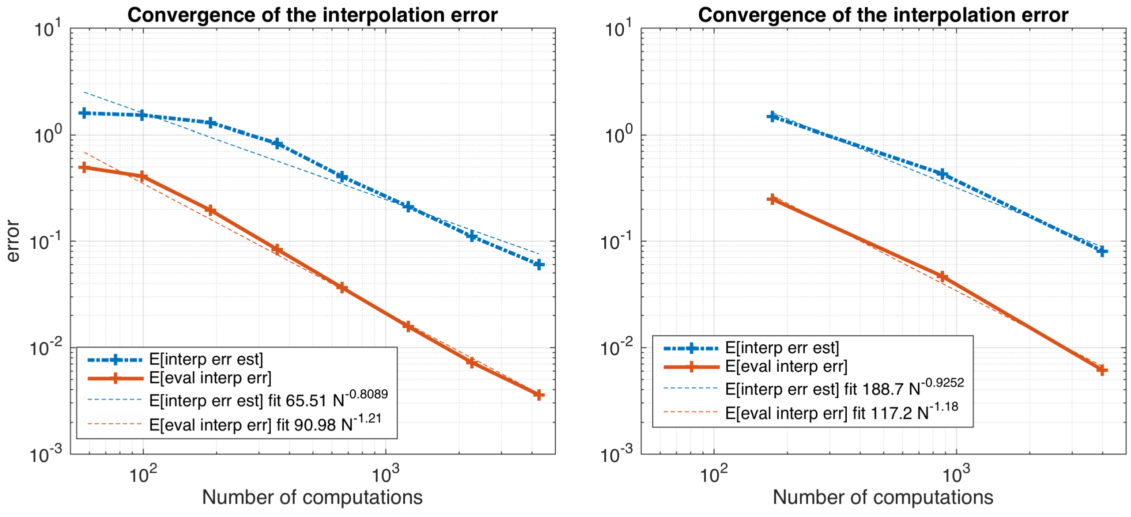


(A) Exterior of the $3D$ mesh.

(B) Cutout of the $3D$ mesh.

FIGURE 4.12: The adapted $3D$ mesh with 4290 samples.

As before, the error convergence is examined using different step sizes, the convergence plots obtained are shown in Figure 4.13.



(A) 8 refinement steps

(B) 3 refinement steps

FIGURE 4.13: Convergence of the interpolation error and the error estimate with three uncertain variables. The interpolation error is computed on a third degree Newton-Cotes quadrature, a sort of subgrid constructed in order to evaluate the real interpolation error.

The theoretical convergence should be of $\mathcal{O}(N^{-\frac{2}{3}}) \approx \mathcal{O}(N^{-0.67})$ and it can be observed that the convergence rate obtained surpasses this prediction. In contrast to the $2D$ case, the actual error decreases faster than the error estimate. This could be due to the fact that in the $3D$ version of this test function the solution is mostly discontinuous w.r.t. the first and second dimension but not in the third. It can be seen from (4.30) that the test function depends quadratically on ξ_3 whilst the dependence on ξ_1 and ξ_2 is discontinuous. As this somewhat reduces the effective dimensionality of the problem it seems that the evaluated expectation of the interpolation error diminishes quicker than the expectation of the interpolation error estimate.

4.5.2 Fluid mechanics stochastic piston problem

The metric-based mesh adaptation for the stochastic space is tested here on a classical application of fluid mechanics: the piston problem. This fundamental problem was revisited several times in the context of uncertainty quantification e.g. [Lin, Su, and Karniadakis, 2004; Zhang et al., 2013]. A particular version was well described by [Witteveen, Loeven, and Bijl, 2009]. In the following, we use the same notations as in that reference, the configuration of which is shown in Figure 4.14. It consists of a tube filled with air, assumed to be an ideal gas, and a piston pushing into the tube from the right. The flow domain is considered one-dimensional and the piston starts moving to the right at time $t = 0$ with constant velocity $u_{\text{piston}} > 0$. The initial conditions in the tube are given by the initial density ρ_{pre}, initial pressure p_{pre} and the initial velocity $u_{\text{pre}} = 0$.

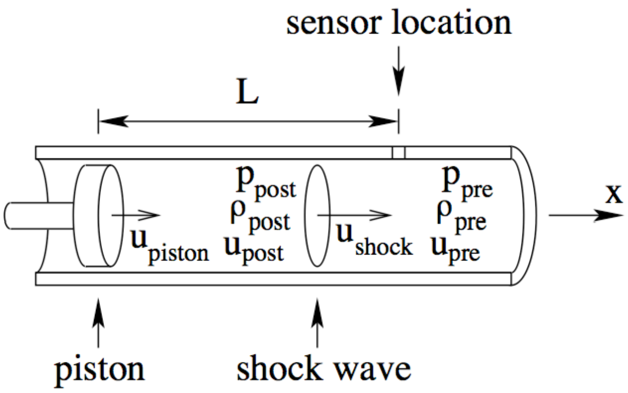


FIGURE 4.14: Setup of the piston problem (figure from [Witteveen, Loeven, and Bijl, 2009]).

As the piston moves to the right a shock runs into the gas with speed u_{shock}. The conditions behind the shock are given by ρ_{post}, p_{post} and $u_{\text{post}} = u_{\text{shock}}$. The effects of viscosity are neglected and so the pressure behind the shock can be obtained from the Rankine-Hugonoit equation

$$p_{\text{post}} - p_{\text{pre}} = \rho_{\text{pre}} c_{\text{pre}} (u_{\text{post}} - u_{\text{pre}}) \sqrt{1 + \frac{\gamma - 1}{2\gamma} \frac{p_{\text{post}} - p_{\text{pre}}}{p_{\text{pre}}}},$$

where the initial sound speed is given by $c_{\text{pre}} = \sqrt{\gamma p_{\text{pre}} / \rho_{\text{pre}}}$ and the specific heat ratio is chosen as $\gamma = 1.4$. The Mach number of the shock is obtained by the one-dimensional shock wave relations [Anderson, 2001]

$$M_{\text{shock}} = \sqrt{1 + \frac{\gamma + 1}{2\gamma} \left(\frac{p_{\text{post}}}{p_{\text{pre}}} - 1 \right)}.$$

The QoI is the instantaneous mass flow measured by the sensor positioned at a distance L from the initial position of the piston.

$$m(t) = \begin{cases} \rho_{\text{pre}} u_{\text{pre}} & \text{if } t < \frac{L}{u_{\text{shock}}}, \\ \rho_{\text{post}} u_{\text{post}} & \text{if } t > \frac{L}{u_{\text{shock}}}. \end{cases}$$

Two uncertain parameters

Identically to [Witteveen, Loeven, and Bijl, 2009], we consider first two uncertain parameters will be u_{piston} and p_{pre}, both being modeled as random variables following a lognormal distribution with mean $\mu_{u_{\text{piston}}} = \mu_{p_{\text{pre}}} = 1$ and $CoV = 10\%$ (the equations are nondimensionalized). The QoI is the mass flow $m(t)$ for $t = 0.5$ at the sensor which is located at $L = 1$.

The mesh adaptation process is visualized in Figure 4.15 where one can see how the curved discontinuous response surface is gradually resolved whilst taking into account the probability distribution ρ_ξ.

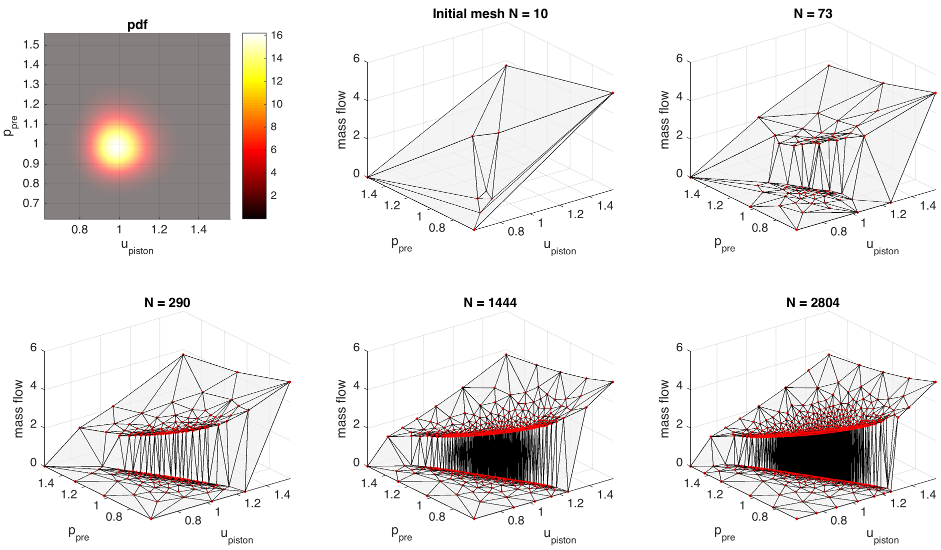


FIGURE 4.15: Adaptive refinement of the response surface. The joint probability density function of u_{piston} and p_{pre} is shown on the top left plot.

Mass flow low-order moments (mean and variance) convergence results are given in Figure 4.16. The errors in mean and variance are computed w.r.t. mean and variance obtained from Monte Carlo simulation (obtained with 10^7 samples). In the same figure the results obtained by Witteveen *et al.* [Witteveen, Loeven, and Bijl, 2009] are also included which allows for a quantitative comparison.

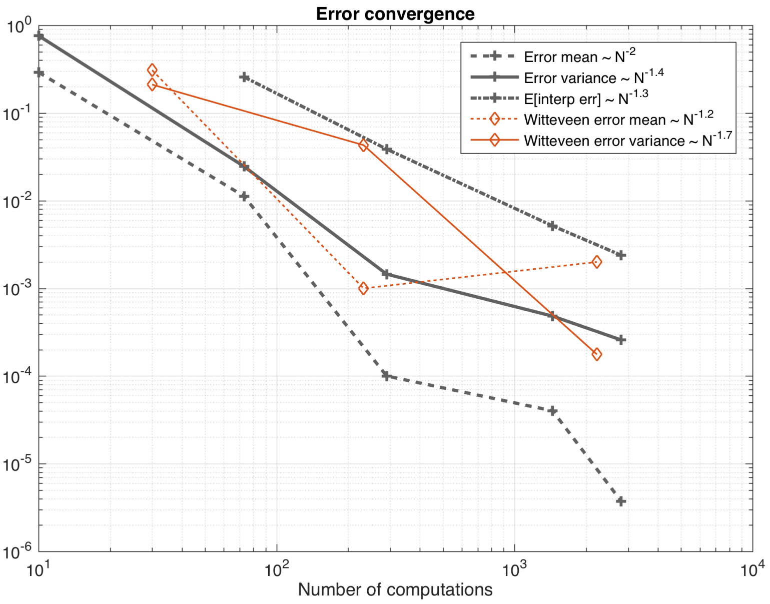
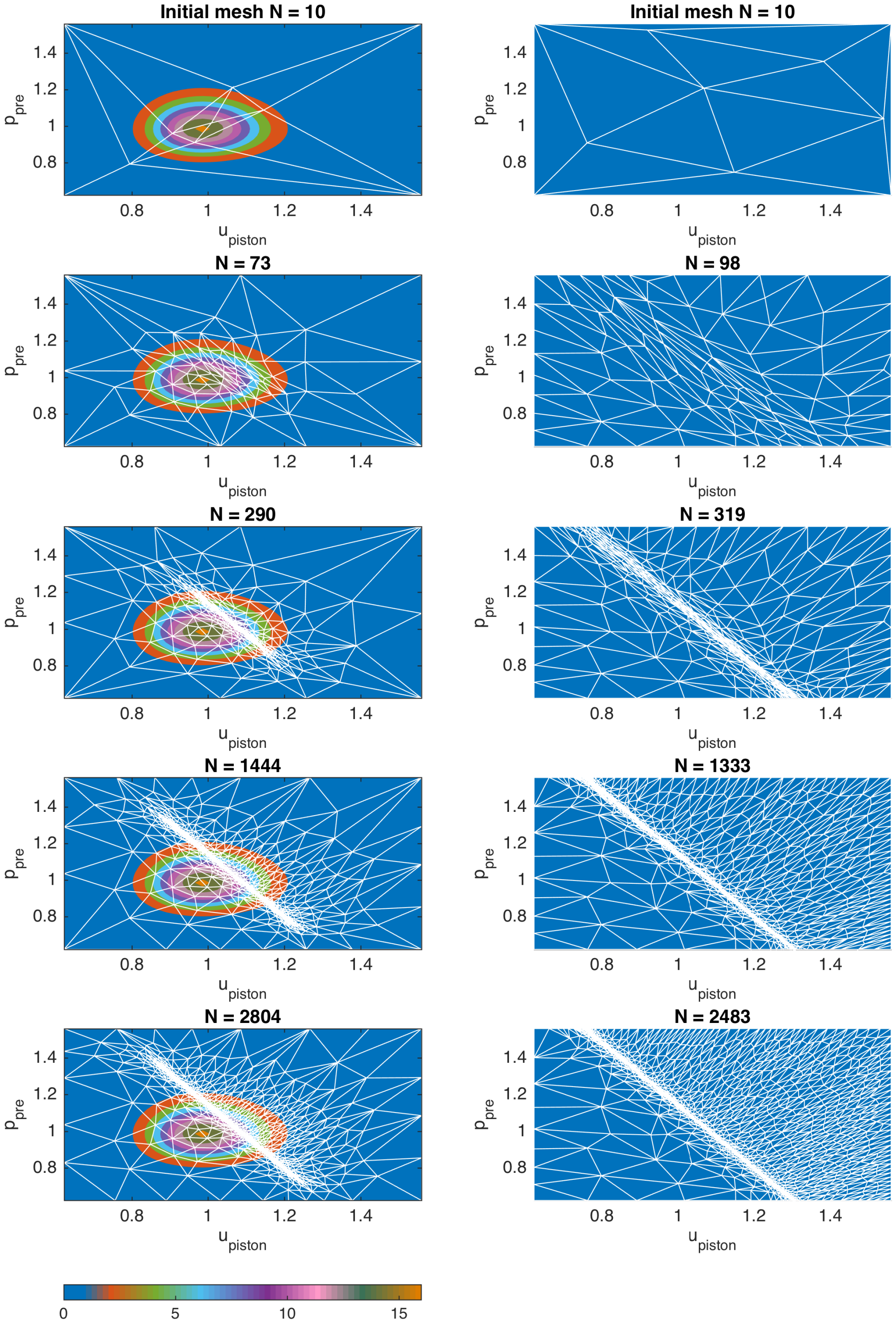


FIGURE 4.16: Piston problem $2D$: Convergence of the error of the mean, variance and the interpolation error $e_{\mathcal{M}^{opt}}$. The results obtained by [Witteveen, Loeven, and Bijl, 2009] are included for comparison. The legend entry *Error mean* signifies the error of $\mathbb{E}[QoI]$ ($|\mathbb{E}[QoI]_{MC} - \mathbb{E}[QoI_{surrogate}]|$), and analogously the entry *Error variance* the error of $\mathrm{Var}[QoI]$ while $\mathbb{E}$[interp err] is $\overline{\eta}$, the expectation of the interpolation error estimate.

In order to underscore how the adaptation process takes into account the probability distribution, Figure 4.17 compares the adaptation process when to random variables follow a lognormal distribution versus the case where the random variables would follow a uniform distribution.



Initial mesh N = 10

Initial mesh N = 10

N = 73

N = 98

N = 290

N = 319

N = 1444

N = 1333

N = 2804

N = 2483

FIGURE 4.17: On the left: mesh adaptation sequence when the two random variables follow a lognormal distribution with $\mu_{u_{\text{piston}}} = \mu_{p_{\text{pre}}} = 1$ and a coefficient of variation of $CoV = 10\%$; the underlying contour map is a representation of the probability density function. On the right: mesh adaptation sequence when the two random variables follow a uniform distribution.

Comparison with ME-gPC The ME-gPC method proposes to decompose the parametric space into hypercube elements; on each of these elements a local polynomial chaos expansion is then constructed and the global solution will be the combination of all these solution patches. Applying the ME-gPC method to the piston problem with two uncertain variables where both variables are uniformly distributed will allow for a comparison with the metric-based simplex-stochastic collocation method demonstrated above. As noted before, the decomposition into hypercube elements used by the ME-gPC method is not the most effective at capturing discontinuities that do not run along one of the principal axes. Secondly, the $hp-$refinement indicators are set by the user and, for optimal performance, these need to be tuned to the particular problem at hand.

The parameter controlling $h-$refinement, θ_1, is a limit on the weighted local variance decay rate (see [Wan and Karniadakis, 2005). Following the definitions from [Wan and Karniadakis, 2005] the parameters contriling the $hp-$adaptivity were chose as follows $\theta_1 = 10^{-8}$, $\theta_2 = 0.5$ and $\alpha = 0.5$; the resulting final domain decomposition is shown in Figure 4.18 and convergence curves are shown in Figure 4.19.

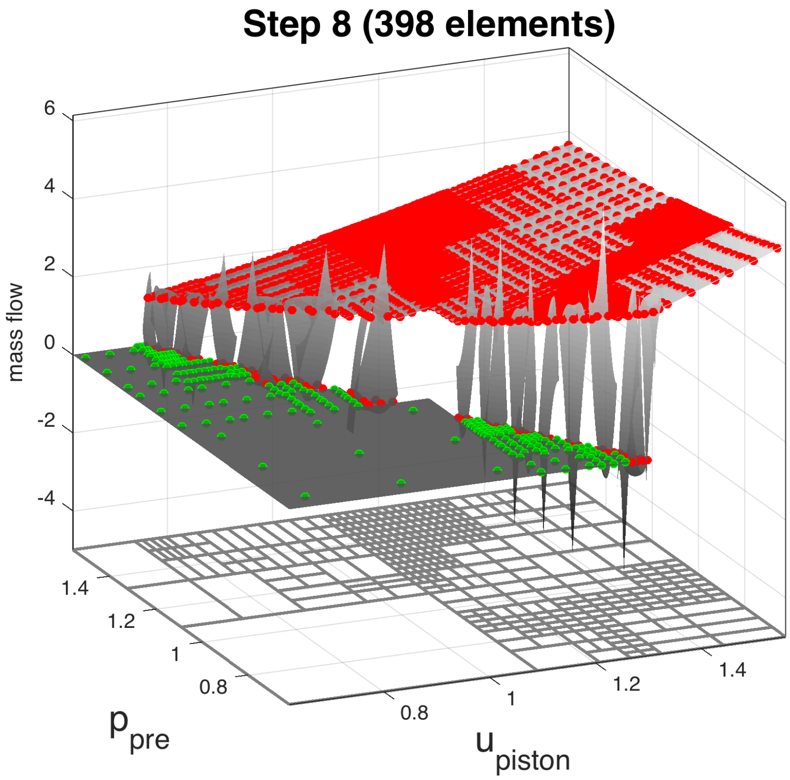


FIGURE 4.18: Piston problem $2D$: Response surface obtained with the ME-gPC method for the piston problem with two uncertain variables following a uniform probability distribution.

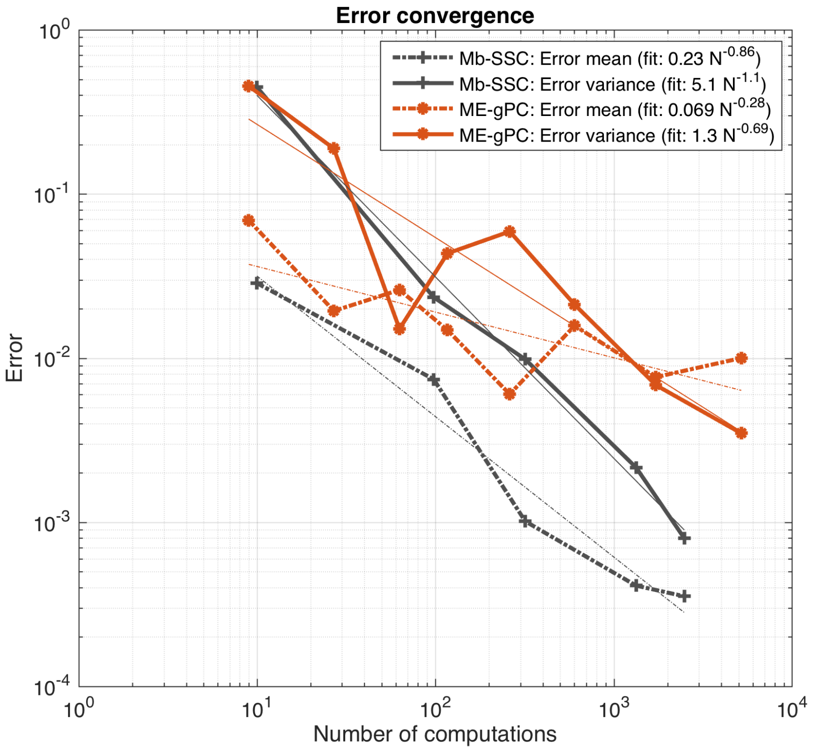


FIGURE 4.19: Comparison of convergence of metric-based SSC and ME-gPC for the piston problem with two uncertain variables both following a uniform probability distribution.

Three uncertain parameters

Similarly to [Witteveen, Loeven, and Bijl, 2009], a third lognormal random variable, the sensor position L, is added to the previous test case. In Figure 4.20, the convergence of the expectation of the interpolation error, the error in the mean and variance are shown. Plotted in the same figure are the results obtained by Witteveen et al.

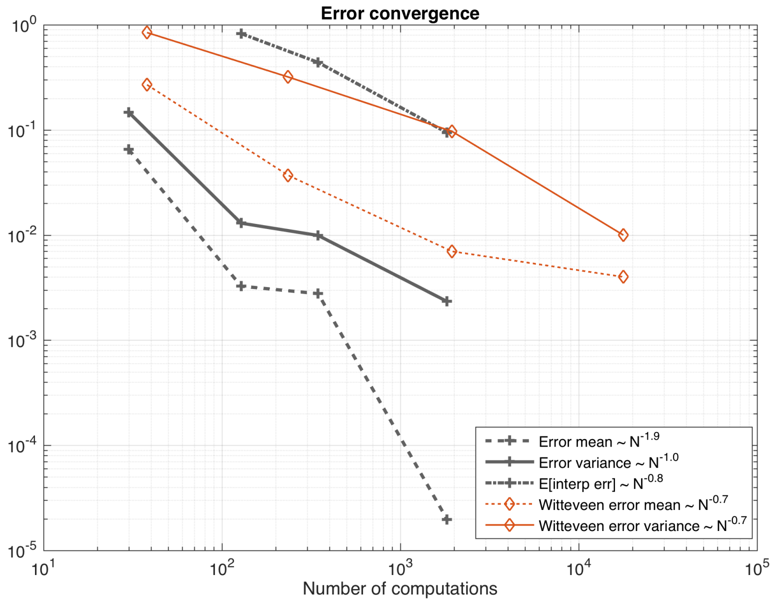


FIGURE 4.20: Piston problem $3D$: Convergence of the error of the mean, variance and the interpolation error $e_{\mathcal{M}^{opt}}$. The results obtained by [Witteveen, Loeven, and Bijl, 2009] are included for comparison.

With respect to the SSC results obtained by Witteveen et al., the metric-based mesh adaptation has a lower convergence constant and a slightly faster convergence of the error in the mean and variance.

4.6 Conclusions

From the results presented in this chapter, the following conclusions can be drawn:

- we show a *convergence in the mean* of the proposed metric-based approximation method for multivariate nonlinear *discontinuous* functionals. More specifically, the method is at least 2nd−order in the $\mathbf{L}_1$−norm, i.e. the mean error is such that: $\mathbb{E}[\eta] \sim \beta \times N^{-2\kappa/D_{\boldsymbol{\xi}}}$, with $(\beta > 0, \kappa \gtrsim 1)$ positive constants
- the convergence is *robust*: – for a given functional with a fixed number of random parameters, a change in the nature and/or the regularity of the probability measure of the parameters does not affect the convergence rate of this mean error, – the choice of the refinement step size does not greatly affect the convergence rate of the error, i.e. κ
- we have verified that the $\mathbf{L}_1$−error of the surrogate solution mean is always lower in magnitude than the mean error of the surrogate; while its convergence rate is significantly higher than the latter one
- we have noticed that the $\mathbf{L}_1$−error of the surrogate solution variance is larger than the error of the mean solution with a rate of convergence lower than the latter one, but higher than the convergence rate of the mean error
- we have investigated the influence of the intial DoE and found that the effect on the convergence rate is low with a CoV of the convergence rate around 10%.

For the piston problem, we have compared our results to the SSC results proposed by [Witteveen, Loeven, and Bijl, 2009], and we have noticed that:

- the proposed metric-based approximation method generates convergence rates – for the surrogate solution mean that are larger than the ones obtained with the SSC method and – for the surrogate solution variance that are of comparable order as the ones obtained with the SSC method
- convergence constants β are smaller with our method
- our convergence is more monotone than the SSC convergence that is based on local refinements.

Chapter 5

Metric-based adaptive error control in deterministic and stochastic spaces

5.1 Introduction

The SSC method and most other adaptive UQ methods mentioned in Chapter 4, make use of refinement criteria that are heuristics computed at a local level in the parametric space and these are often tailored to specific problems. While for some it has been shown that adaptation following the given criterion will lead to convergence, these criteria do not permit a straightforward comparison between the error committed in the continuous representation of the stochastic response and the error introduced in each sample by the deterministic solver approximation. We propagate the uncertain parameters though an approximate model (Ψ_h), which leads to errors in each sample and consequently to errors in the stochastic response surface. Potentially large computational gains in efficiency are to be realized if one were able to predict what level of sampling refinement is needed in each deterministic computation and what level of refinement is needed is the representation of the stochastic response in order to remain within a given total error level. By an extension of a dual-based *a posteriori* error estimate, the authors in [Bryant, Prudhomme, and Wildey, 2015] achieved this splitting of the error into deterministic and stochastic contributions in order to drive adaptivity in both spaces. In the parametric space, along with global (anisotropic) $p-$refinement, a Multi-Element style $h-$ and $h/p-$adaptivity methods are proposed which have the same deficiencies as the standard ME-gPC method: the refinement parameters driving $h-$ and $p-$adaptivity are chosen by the user and need to be tailored to the problem for optimal performance; secondly the use of hypercube elements does not permit efficient representation of discontinuities. The extension of *a posteriori* goal-oriented error estimates has previously been applied to problems with uncertain input parameters [Mathelin and Le Maître, 2007; Le Maître and Knio, 2010; Almeida and Oden, 2010; Butler, Dawson, and Wildey, 2011; Butler, Constantine, and Wildey, 2012] but, to the best of the author's knowledge, the work from Bryant et al. [Bryant, Prudhomme, and Wildey, 2015] is to date the only publication that demonstrates error splitting and automated adaptivity in both spaces. Other authors demonstrate estimates of the total error (the deterministic and stochastic contributions combined) [Mathelin and Le Maître, 2007], computable estimates of the error in each deterministic sample [Almeida and Oden, 2010], or estimates of the error of the stochastic response approximation [Butler, Dawson, and Wildey, 2011; Butler, Constantine, and Wildey, 2012] (although in [Butler, Dawson, and Wildey, 2011] adaptivity in the parametric space is not addressed).

Chapter partially based on Van Langenhove J., Lucor D., F. Alauzet and A. Belme. "Goal-oriented control of stochastic system approximations using metric-based anisotropic adaptation". (to be submitted to *Journal of Computational Physics*).

A splitting of the *total error* committed on the quantity of interest in contributions from stochastic and deterministic approximations has been proposed in Chapter 4. We have shown there how a Riemannian metric approach applies to the parameter space in order to control the stochastic interpolation error. This chapter builds upon the method presented in the previous chapter, but, whereas Chapter 4 was focused only on the adaptation of stochastic responses as demonstrated on analytical and semi-analytical test cases, in the present chapter, we will move to CFD applications *both* the deterministic and stochastic errors are controlled for internal and external flow simulations. An estimate of the deterministic error on each sample is computed using an *a priori* error estimate in the continuous framework of Riemannian metric space.

The structure of this chapter will be as follows: firstly the goal-oriented deterministic error estimate will be presented in Section 5.2. This will build upon the formulation and separation into a deterministic and stochastic error presented in Chapter 4. This is followed up by Section 5.3 in which the adaptive strategies for the coupled deterministic-stochastic refinement problem are proposed. Next, results obtained on several CFD test cases will be presented in Section 5.4, and finally, some conclusions will be summarized in Section 5.5.

5.2 Goal-oriented deterministic error estimate

Let us start with a brief recall of some elements introduced in Chapter 4. The abstract formulation of the model problem was given as:

$$\Psi(\boldsymbol{\xi}, w(\boldsymbol{\xi}, \mathbf{x})) = 0 \tag{5.1}$$

where Ψ is a state equation (for example a steady Navier-Stokes or Euler system) having $w(\boldsymbol{\xi}, \mathbf{x})$ for exact solution. In its discretized form, (5.1) becomes:

$$\Psi_{h_{\mathbf{x}}}(\boldsymbol{\xi}_{(i)}, w_{h_{\mathbf{x}}}(\boldsymbol{\xi}_{(i)}, \mathbf{x})) = 0, \tag{5.2}$$

with $w_{h_{\mathbf{x}}}$ the deterministic discrete solution associated to sample $\boldsymbol{\xi}_{(i)}$ and solved on a mesh $\mathcal{M}_{h_{\mathbf{x}}}$. Furthermore, the quantity of interest j was defined as a functional of the solution obtained for some fixed combination of the random parameters $\boldsymbol{\xi}_{(i)}$:

$$j_{h_{\boldsymbol{\xi}}}(\boldsymbol{\xi}) = J_{h_{\boldsymbol{\xi}}}\big(\boldsymbol{\xi}, w_{h_{\mathbf{x}}}(\boldsymbol{\xi}, \mathbf{x})\big), \tag{5.3}$$

where the subscript $h_{\boldsymbol{\xi}}$ indicates that the deterministic solution was computed via a discretization $\mathcal{M}_{h_{\boldsymbol{\xi}}}$. The separation of the error on the quantity of interest, can then be written as:

$$\delta j(\boldsymbol{\xi}) = \underbrace{J(\boldsymbol{\xi}, w) - J(\boldsymbol{\xi}, w_{h_{\mathbf{x}}})}_{\varepsilon(h_{\mathbf{x}}, h_{\boldsymbol{\xi}})} + \underbrace{J(\boldsymbol{\xi}, w_{h_{\mathbf{x}}}) - J_{h_{\boldsymbol{\xi}}}(\boldsymbol{\xi}, w_{h_{\mathbf{x}}})}_{\eta(h_{\mathbf{x}}, h_{\boldsymbol{\xi}})}. \tag{5.4}$$

In practice, δj being a random quantity, we will be interested in lowering the *average* total error, that we express as:

$$\overline{\delta j} \equiv \mathbb{E}\left[\delta j(\boldsymbol{\xi})\right] = \underbrace{\mathbb{E}\left[\varepsilon\right]}_{\overline{\varepsilon}} + \underbrace{\mathbb{E}\left[\eta\right]}_{\overline{\eta}} \tag{5.5}$$

A simple adaptive approach might be chosen in order to lower both contributions of the right hand side of previous equation, i.e.: 1. optimizing the set of spatial discretizations $\mathcal{H}_{h_{\mathbf{x}}}$ in order to diminish (or balance) the ε error contribution of the samples of a given

DoE and concurrently 2. update the DoE to lower $\mathbb{E}\left[\eta\right]$ for a given set of $\mathcal{H}_{h_{\mathbf{x}}}$.

The deterministic error contribution in (5.4), which appears each time the parameter set is given/fixed is discussed in this section. An *a priori* error estimation of this error is utilized, involving the computation of an adjoint problem. Only the main results are recalled and we refer to [Loseille, 2008; Loseille, Dervieux, and Alauzet, 2010] for a more detailed introduction. In this section, we have replaced $\cdot_{h_{\mathbf{x}}}$ by $\cdot_{h}$ for ease of notation and $\boldsymbol{\xi}$ refers to a fixed/given value unless mentioned otherwise.

First, the variational problem associated to state equations (5.1-5.2) on an appropriate Hilbert space of solutions $\mathcal{V}$ and respectively subspace $\mathcal{V}_h$ are introduced:

$$\text{Find } w(\boldsymbol{\xi}_{(i)},\cdot) \in \mathcal{V} \text{ such that } \forall \varphi \in \mathcal{V}, \quad \langle \Psi(w)\,,\, \varphi \rangle = 0, \tag{5.6}$$

with the associated discrete variational formulation:

$$\text{Find } w_h(\boldsymbol{\xi}_{(i)},\cdot) \in \mathcal{V}_h \text{ such that } \forall \varphi_h \in \mathcal{V}_h, \quad \langle \Psi_h(w_h)\,,\, \varphi_h \rangle = 0. \tag{5.7}$$

Furthermore, we assume some level of regularity for our QoI such that :

$$j(\boldsymbol{\xi}_{(i)}) \in \mathbb{R}\,;\; j = J(\boldsymbol{\xi}_{(i)}, w) = \langle g, w\rangle = \int_{\Omega_x} g\, w(\boldsymbol{x}, \boldsymbol{\xi}_{(i)})\mathrm{d}\boldsymbol{x} \tag{5.8}$$

where g represents the deterministic Jacobian of J. We introduce the *continuous adjoint* solution w^* of the following system:

$$w^*(\boldsymbol{\xi}_{(i)},\cdot) \in \mathcal{V}\,,\; \forall \psi \in \mathcal{V}\,,\; \left\langle \frac{\partial \Psi}{\partial w}(w)\psi, w^* \right\rangle = \langle g, \psi\rangle\,. \tag{5.9}$$

We assume that both state solution w and adjoint solution w^* are smooth enough, such that $w, w^* \in \mathcal{V} \cap \mathcal{C}^0$.
With $\mathcal{V}_h \subset \mathcal{V}$, the following error estimates for the unknown can be derived:

$$\langle \Psi_h(w), \varphi_h\rangle - \langle \Psi_h(w_h), \varphi_h\rangle = \langle \Psi_h(w), \varphi_h\rangle - \langle \Psi(w), \varphi_h\rangle = \langle (\Psi_h - \Psi)(w), \varphi_h\rangle. \tag{5.10}$$

The error committed on the functional, which we are trying to estimate, is given by:

$$\langle g, w\rangle - \langle g, w_h\rangle = \langle g, w - w_h\rangle$$

with w and w_h the respective solutions of (5.1-5.2).
The idea is then to define a local error estimation to be used as a guide for anisotropic mesh refinement. Interpolation errors are known to provide useful local information and in our case, e.g. directions and sizes for anisotropic mesh refinement. We introduce thus an interpolation operator: $\pi_h : \mathcal{V} \cap \mathcal{C}^0 \to \mathcal{V}_h$, which allows for a simple decomposition of the deterministic approximation error into two components:

$$J(\boldsymbol{\xi}_{(i)}, w) - J(\boldsymbol{\xi}_{(i)}, w_h) = \langle g, w - \pi_h w\rangle + \langle g, \pi_h w - w_h\rangle, \tag{5.11}$$

where the first term is the *interpolation error* and it can be shown (see [Loseille, Dervieux, and Alauzet, 2010]) that the second error term, called here the *implicit error*, can also be expressed in terms of interpolation errors. The following estimation is proposed for the deterministic error ε [Loseille, Dervieux, and Alauzet, 2010]:

$$\varepsilon(\boldsymbol{\xi}_{(i)}) = J(\boldsymbol{\xi}_{(i)}, w) - J(\boldsymbol{\xi}_{(i)}, w_h) \approx \Big\langle w^*, \Big(\Psi_h(\boldsymbol{\xi}_{(i)}, w) - \Psi(\boldsymbol{\xi}_{(i)}, w)\Big)\Big\rangle\,. \tag{5.12}$$

The *a priori* error estimate (5.12) is used here as refinement indicator for the deterministic mesh adaptation. More specifically, we build our mesh adaptation as an optimisation problem where we seek the optimal mesh that minimises (5.12) under the constraint of a given number of mesh nodes; the adjoint state w^* acting as a Lagrange multiplier associated to the equality constraint (5.1).

5.2.1 Application to nonlinear conservation laws: compressible Euler flow

The previously defined formal error analysis is illustrated next on the concrete example of the compressible Euler system. Note that we provide details of the discretization for fixed $\boldsymbol{\xi}$, thus for ease of notation, the dependence on $\boldsymbol{\xi}$ and $\mathbf{x}$ will not be explicitly written out.

The two-dimensional time-independent Euler equations are set in the computational domain $\Omega \subset \mathbb{R}^2$ with a boundary denoted Γ, and will be working in the function space$\mathcal{V} = \left[H^1(\Omega) \cap \mathcal{C}(\bar{\Omega})\right]^4$ (the set of measurable functions that are continuous and have a square integrable gradient). The reason for choosing this function space is because it will allow us to use the elementwise linear interpolation operator, an essential ingredient in the discretization. The variational formulation of the Euler equations is done in the function space $\mathcal{V}$ and reads:

$$\begin{aligned} &\text{Find } W \in \mathcal{V} \text{ such that } \forall \varphi \in \mathcal{V}, \quad \langle \Psi(W)\,,\, \varphi \rangle = 0 \\ \text{with } \langle \Psi(W)\,,\, \varphi \rangle &= \int_\Omega \varphi \nabla \cdot \mathcal{F}(W)\, \mathrm{d}\Omega \; - \int_\Gamma \varphi \hat{\mathcal{F}}(W).\mathbf{n}\, \mathrm{d}\Gamma \,. \end{aligned} \tag{5.13}$$

where $\mathbf{n}$ denotes the outward normal to Γ. Furthermore, W is used to denote the vector of conservative flux variables, the usual Euler flux is written as $\mathcal{F}(W) = (\mathcal{F}_1(W), \mathcal{F}_2(W))$, and $\hat{\mathcal{F}}$ is the boundary flux containing the boundary conditions. The contents of W and $\mathcal{F}$ are specified below:

$$W = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{pmatrix} \; ; \quad \mathcal{F}(W) = \begin{pmatrix} \rho \mathbf{u} \\ \rho \mathbf{u} u + p \mathbf{e}_x \\ \rho \mathbf{u} v + p \mathbf{e}_y \\ \rho \mathbf{u} H \end{pmatrix} \,. \tag{5.14}$$

In the above relation, ρ is the fluid density, p the thermodynamic pressure, E the total energy per unit mass, H is the total enthalpy given by $H = E + \frac{p}{\rho}$, and finally u and v are the Cartesian components of the velocity vector $\mathbf{u}$.

Discrete state system The Mixed-Element-Volume formulation [Cournède, Koobus, and Dervieux, 2006] is used as a semi-discrete model, and, as in [Loseille, Dervieux, and Alauzet, 2010] it is reformulated to a finite element variational formulation. The finite element partitioning of Ω in simplex elements denoted K is proposed here, with the set of elements being the mesh $\mathcal{H}$. Furthermore, the approximation space $\mathcal{V}_h$ is given by

$$V_h \;=\; \left\{ \varphi_h \in V \;\middle|\; \varphi_{h|K} \text{ is affine } \forall K \in \mathcal{H} \right\},$$

and Π_h is the $\mathcal{P}^1$ projection operator:

$$\Pi_h : \; V \rightarrow \; V_h \;\; \text{such that} \;\; \Pi_h \varphi(\mathbf{x}_i) = \varphi(\mathbf{x}_i), \; \forall\, \mathbf{x}_i \text{ vertex of } \mathcal{H}.$$

The weak discrete formulation can now be written as:

$$\begin{aligned}&\text{Find } W_h \in \mathcal{V}_h \text{ s.t. } \forall \varphi_h \in \mathcal{V}_h,\ \langle \Psi_h(W_h)\,,\,\varphi_h\rangle = 0,\\ &\text{with } \langle \Psi_h(W_h)\,,\,\varphi_h\rangle = \int_\Omega \varphi_h \nabla\cdot\mathcal{F}_h(W_h)\,\mathrm{d}\Omega - \int_\Gamma \varphi_h \hat{\mathcal{F}}_h(W_h).\mathbf{n}\,\mathrm{d}\Gamma + \int_\Omega \varphi_h\, D_h(W_h)\,\mathrm{d}\Omega,\end{aligned} \tag{5.15}$$

where $\mathcal{F}_h = \Pi_h\mathcal{F}$ and $\hat{\mathcal{F}}_h = \Pi_h\hat{\mathcal{F}}$. The term D_h in the above equation represents the numerical diffusion which accounts for the difference between the Galerkin central-difference approximation and the second-order Godunov approximation [Cournède, Koobus, and Dervieux, 2006]. In the presence of shocks, monotonicity limiters become first-order terms, but for smooth fields, the D_h terms will be a third order term with respect to the mesh size.

Deterministic error estimate applied to Euler model The deterministic error estimate applied to the Euler model is obtained by replacing in estimate (5.12) the operators Ψ and Ψ_h by their expressions given by Relations (5.13) and (5.15). As in [Loseille, Dervieux, and Alauzet, 2010], where it was observed that even for shocked flows, it is possible to neglect the viscosity term, here we follow the same guideline. In doing so, the following simplified error model is obtained:

$$\varepsilon \approx \int_\Omega W^*\,\nabla.\big(\mathcal{F}(W) - \Pi_h\mathcal{F}(W)\big)\,\mathrm{d}\Omega - \int_\Gamma W^*\,\big(\hat{\mathcal{F}}(W) - \Pi_h\hat{\mathcal{F}}(W))\big).\mathbf{n}\,\mathrm{d}\Gamma\ . \tag{5.16}$$

Integrating by parts leads to:

$$\varepsilon \approx \int_\Omega \nabla W^*\,\big(\mathcal{F}(W) - \Pi_h\mathcal{F}(W)\big)\,\mathrm{d}\Omega - \int_\Gamma W^*\,\big(\bar{\mathcal{F}}(W) - \Pi_h\bar{\mathcal{F}}(W))\big).\mathbf{n}\,\mathrm{d}\Gamma\,. \tag{5.17}$$

with $\bar{\mathcal{F}} = \hat{\mathcal{F}} - \mathcal{F}$. We observe that this estimate of δj is expressed in terms of interpolation errors of the Euler fluxes weighted by continuous functions W^* and ∇W^*.

In the error estimate (5.17) both positive and negative parts can (partially) cancel each other out on specific meshes. We prefer however not to rely on such an eventuality and rather choose to over-estimate the error by bounding all integrands by their absolute values:

$$\varepsilon \ \le\ \int_\Omega |\nabla W^*|\,|\mathcal{F}(W) - \Pi_h\mathcal{F}(W)|\,\mathrm{d}\Omega + \int_\Gamma |W^*|\,|(\bar{\mathcal{F}}(W) - \Pi_h\bar{\mathcal{F}}(W)).\mathbf{n}|\,\mathrm{d}\Gamma\ . \tag{5.18}$$

5.2.2 Continuous error model

The continuous framework of the Riemannian metric space presented in Chapter 3 will be used for the control of the deterministic error as well. We thus seek the optimal mesh with a given number of vertices that minimizes the deterministic error ε.

Continuous error model applied to Euler equations Working in the continuous framework enables us to write Estimate (5.18) in the following continuous form:

$$\begin{aligned}(g, W_h - W) \approx \mathbf{E}_{\mathbf{x}}(\mathbf{M}) \ &=\ \int_\Omega |\nabla W^*|\,|\mathcal{F}(W) - \pi_{\mathcal{M}}\mathcal{F}(W)|\,d\Omega\\ &+\ \int_\Gamma |W^*|\,|(\bar{\mathcal{F}}(W) - \pi_{\mathcal{M}}\bar{\mathcal{F}}(W)).\mathbf{n}|\,d\Gamma.\end{aligned} \tag{5.19}$$

We observe that the second term introduces a dependency of the error with respect to the boundary surface mesh. In the present work, we do not take this term into account and refer to [Loseille, Dervieux, and Alauzet, 2010] for a discussion on its importance. Then, introducing the continuous interpolation error, we can write the simplified error model as follows:

$$\begin{aligned} \mathbf{E}_{\mathbf{x}}(\mathbf{M}) &= \int_{\Omega} \text{trace}\left(\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})\,\mathbf{H}(\mathbf{x})\,\mathcal{M}^{-\frac{1}{2}}(\mathbf{x})\right)\,\mathrm{d}\Omega \\ \text{with} \quad \mathbf{H}(\mathbf{x}) &= \sum_{j=1}^{4}\left([\Delta x]_j(\mathbf{x}) + [\Delta y]_j(\mathbf{x})\right), \end{aligned} \tag{5.20}$$

in which

$$[\Delta x]_j(\mathbf{x}) = \left|\frac{\partial W_j^*}{\partial x}(\mathbf{x})\right| \cdot \left|H(\mathcal{F}_1(W_j))(\mathbf{x})\right| \tag{5.21}$$

$$[\Delta y]_j(\mathbf{x}) = \left|\frac{\partial W_j^*}{\partial y}(\mathbf{x})\right| \cdot \left|H(\mathcal{F}_2(W_j))(\mathbf{x})\right|. \tag{5.22}$$

Here, $H(\mathcal{F}_i(W_j))$ denotes the Hessian of the j^{th} component of the vector $\mathcal{F}_i(W)$.
The *deterministic mesh optimization problem* is formulated as:

$$\text{Find } \mathbf{M}_{\mathbf{x}}^{opt} = \underset{\mathbf{M}}{\text{argmin}}\ \mathbf{E}_{\mathbf{x}}(\mathbf{M}), \tag{5.23}$$

under the constraint of bounded mesh fineness:

$$\mathcal{C}(\mathbf{M}) = N_{\mathbf{x}}, \tag{5.24}$$

where $N_{\mathbf{x}}$ is a specified total number of nodes. The solution process is analogous to the one outlined in Section 4.3 and yields:

$$\mathcal{M}_{\mathbf{x}}^{opt} = N_{\mathbf{x}}^{\frac{2}{D_{\mathbf{x}}}} \left(\int_{\Omega_{\mathbf{x}}} \det(K|H_{\mathbf{x}}|)^{\frac{1}{2+D_{\mathbf{x}}}}\,\mathrm{d}\mathbf{x}\right)^{-\frac{2}{D_{\mathbf{x}}}} \det(K|H_{\mathbf{x}}|)^{-\frac{1}{2+D_{\mathbf{x}}}}\,|K H_{\mathbf{x}}| \tag{5.25}$$

where, for our Euler model, $K = \nabla W^*$ and $H_{\mathbf{x}} = H(\mathcal{F}(W))$.
The error estimate on this optimal metric will be:

$$\mathbf{E}_{\mathbf{x}}^{opt}(\mathbf{M}_{\mathbf{x}}^{opt}) = D_{\mathbf{x}} N_{\mathbf{x}}^{-\frac{2}{D_{\mathbf{x}}}} \underbrace{\left(\int_{\Omega} \det(\nabla W^* |\mathbf{H}(\mathbf{x})|)^{\frac{1}{2+D_{\mathbf{x}}}}\,\mathrm{d}\mathbf{x}\right)^{\frac{2+D_{\mathbf{x}}}{D_{\mathbf{x}}}}}_{\mathcal{K}_{\mathbf{x}}}, \tag{5.26}$$

The equivalent optimisation problem on the stochastic space was presented in Chapter 4. We thus have two optimisation problems to solve in the continuous framework of the Riemannian metric space for the deterministic and respectively stochastic problems.

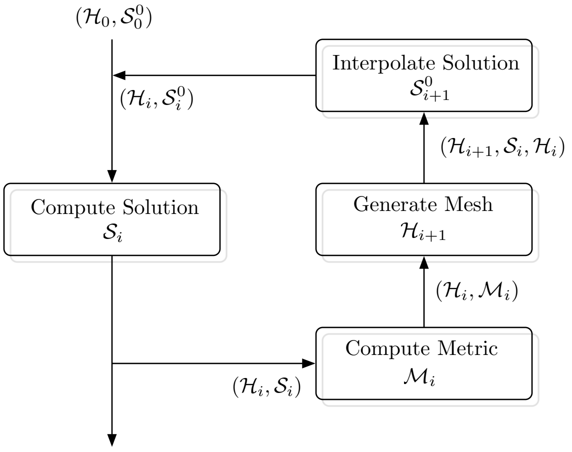
5.3 Adaptive strategies for total error control

In Chapter 3 the adaptation problem was formulated in the continuous framework of the Riemannian metric space, where the mesh refinement problem became a continuous optimization problem. Subsequently, in Chapter 4, this metric-based refinement was used to construct surrogate models in the stochastic space. Here, both the deterministic and the stochastic error will be controlled using this refinement technique. Keep in mind that, as indicated in Section 4.4, there are some differences between the refinement of the deterministic computational domain and the refinement of the mesh/DoE on the parametric domain.

In this section we first briefly recall the fixed-point mesh adaptation algorithm for the deterministic space. Next, we propose a coupled algorithmic approach for a more optimal control of *both* deterministic and stochastic error contributions.

5.3.1 Deterministic adaptive strategy

Regarding deterministic mesh adaptivity, we follow the work of [Alauzet, 2003; Belme, 2011; Loseille, 2008] and employ a fixed point algorithm as illustrated in Figure 5.1. Indeed, the mesh adaptation problem is a non-linear problem, and an iterative algorithm is well suited to converge the couple mesh-solution. The stopping criteria can be a targeted error lever or, as is usually proposed in practice, a maximum number of iterative loops. In general, 5 fixed-point iterations are enough to reach a satisfying level of convergence.



$(\mathcal{H}_0, \mathcal{S}_0^0)$
$(\mathcal{H}_i, \mathcal{S}_i^0)$
Interpolate Solution $\mathcal{S}_{i+1}^0$
$(\mathcal{H}_{i+1}, \mathcal{S}_i, \mathcal{H}_i)$
Compute Solution $\mathcal{S}_i$
Generate Mesh $\mathcal{H}_{i+1}$
$(\mathcal{H}_i, \mathcal{M}_i)$
Compute Metric $\mathcal{M}_i$
$(\mathcal{H}_i, \mathcal{S}_i)$

FIGURE 5.1: Schematic illustrating the mesh adaptation process. The couple (mesh, solution) is denoted $(\mathcal{H}_i, \mathcal{S}_i)$ where the subscript i denotes the fixed-point iteration number; $\mathcal{S}_0^0$ is the initial solution on mesh $\mathcal{H}_0$ whereas $\mathcal{S}_i^0$ is the initial solution interpolated on the mesh $\mathcal{H}_i$. Each discrete mesh $\mathcal{H}_i$ is based on the continuous metric $\mathcal{M}_i$.

5.3.2 Optimal adaptive strategy

We have seen up until now the adaptive strategies and algorithms when solving the optimisation problems 5.23 or 4.12. However, as part of the motivation and novelty of this paper we wish to: (a) quantify the deterministic error at each sample of the parameters space and compute a local (sample-wise) optimization deterministic problem with a required complexity $\mathcal{C}(\mathbf{M})$ and to (b) be able to choose which error, stochastic or deterministic, dominates a computation and thus to solve the corresponding problem.
There are several options for the first issue (a). Indeed, the two sources of errors: stochastic and deterministic, are strongly coupled. Suppose we compute our QoI for a sample $\boldsymbol{\xi}_{(i)}$ on a uniform mesh. This mesh is usually not fit to our QoI but to the studied problem in general. Thus, very often, large deterministic errors can propagate to the parameter space. This is even more pronounced when dealing with problems involving shocks. Very often, the true solution is unreachable. However, we are now able to build an adapted

mesh to best observe our QoI and thus reduce the propagation of the deterministic error to the stochastic space by simply correcting a computation. Moreover, the deterministic error on each sample does not necessarily contribute to the overall problem with the same level of error. Ideally, one would want the mean deterministic error to be lower than or equal to some target error $\bar{\varepsilon}^t$, best chosen to be comparable to the surrogate model error $\bar{\eta}$. Secondly, one would want the variance of the error contained in all the sample to be as close as possible to zero. This ensures that all the samples used to construct the surrogate model of the QoI are of equal accuracy. Since an interpolation method is used for the surrogate model, having a large error in a few samples can be detrimental to the quality of the surrogate model. A trivial solution is to require that the deterministic error in each sample be equal to $\bar{\varepsilon}^t$. This results in the expected deterministic error to be equal to $\bar{\varepsilon}^t$ and its variance to be (as close as possible to) zero.

Following the error model (5.26) the required complexity (or number of nodes $N^t_{\mathbf{x}}$) for this case can be computed as:

$$N^t_{\mathbf{x}} = \left(\frac{D_{\mathbf{x}} \mathcal{K}_{\mathbf{x}}}{\bar{\varepsilon}^t} \right)^{\frac{D_{\mathbf{x}}}{2}} . \tag{5.27}$$

using the notation $\mathcal{K}_{\mathbf{x}}$ introduced in (5.26).

The detailed optimisation algorithm is presented in Algorithm 2 hereafter. This optimisation strategy can be coupled with Algorithm 1 in order to address issue (b) and control both stochastic and deterministic errors. Indeed, one can now choose which errors dominates our computation by comparing $\bar{\varepsilon}$ and $\bar{\eta}$. The required complexity (or number of samples $N^t_{\boldsymbol{\xi}}$) for the stochastic problem will be computed following a similar approach from (4.14):

Algorithm 2 Sample-wise control of the deterministic error contribution over the parametric domain

```
Compute mean deterministic error over parameter space $\bar{\varepsilon} = \mathbb{E}[\varepsilon(\boldsymbol{\xi})] = \int_{\Xi} \varepsilon(\boldsymbol{\xi}) \rho_{\xi} \mathrm{d}\boldsymbol{\xi}$.
for $i = 1$ to $N_{\boldsymbol{\xi}}$ do
    if $\varepsilon(\boldsymbol{\xi}_{(i)}) > \bar{\varepsilon}^t$ then ( $\varepsilon(\boldsymbol{\xi}_{(i)})$ is local deterministic error at $\boldsymbol{\xi}_{(i)}$)
        Compute required complexity $\mathcal{C}^t_{(i)}$ from 5.27 to reduce deterministic error.
        Solve optimisation problem (5.23) associated to this computed complexity $\mathcal{C}^t_{(i)}$.
        Update $j(\boldsymbol{\xi}_{(i)})$ value to sample point $\boldsymbol{\xi}_{(i)}$ in the parametric domain.
    else
        Keep $j(\boldsymbol{\xi}_{(i)})$ value.
    end if
end for
```

$$N^t_{\boldsymbol{\xi}} = \left(\frac{D_{\boldsymbol{\xi}} \mathcal{K}_{\boldsymbol{\xi}}}{\bar{\eta}^t} \right)^{\frac{D_{\boldsymbol{\xi}}}{2}} , \tag{5.28}$$

where $\mathcal{K}_{\boldsymbol{\xi}}$ was introduced in (4.14). The coupled adaptation strategy is detailed below in Algorithm 3.

Algorithm 3 Total error control strategy

```
Generate and compute (or read) initial samples $\{\boldsymbol{\xi}\}_0$ on uniform or adapted initial deterministic mesh $\mathcal{H}_{\boldsymbol{x},0}$.
Form initial stochastic mesh $\mathcal{H}_{\xi,0}$ by a Delaunay triangulation (or load initial existing mesh).
Compute the stochastic error $\bar{\eta}_0$ and the mean deterministic error $\bar{\varepsilon}_0$.
Set maximal number of iteration cycles $it_{MAX}$ and set total target error value $\bar{\delta j}^t$.
while $l < it_{MAX}$ OR $\bar{\delta j} \leq \bar{\delta j}^t$ do
    if $\bar{\varepsilon}_l > \bar{\eta}_l$ then
        Adapt deterministic computations following Algorithm 2 with $\bar{\varepsilon}^t = \bar{\eta}_l$.
    else
        Adapt in parametric domain following Algorithm 1.
    end if
    $l = l + 1$
end while
```

5.4 Numerical applications

5.4.1 Supersonic inlet/isolator

The interest in and difficulties of scramjet propulsion were underscored in Chapter 2 where a test case involving a scramjet engine was treated. Here we treat the inlet and isolator of a dual-mode ramjet/scramjet engine. At low Mach numbers, this engine operates as a ramjet while at high supersonic to hypersonic Mach numbers it operates in scramjet mode. We will look at the internal flow of the inlet/isolator configuration depicted in Figure 5.2. The tapered part on the left (upstream) is the inlet and the straight duct, called the isolator, connects the inlet to the combustion chamber. The capacity of the inlet/isolator to provide an increase in static pressure before the combustor directly impacts the engine performance. The static pressure gain depends on the geometry of the inlet/isolator and the flow profile. Experimental investigations for this type of configuration can be found in [Wagner et al., 2007; Wagner et al., 2009], the interested reader is referred to these works for a more in-depth treatment of the physics involved.

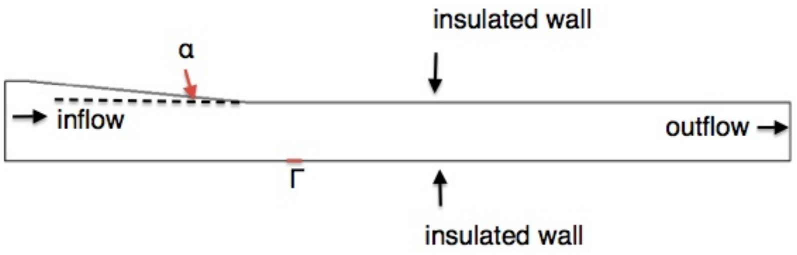
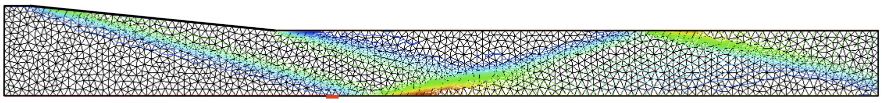
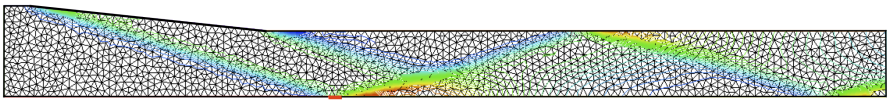


FIGURE 5.2: Inlet problem: geometry configuration with an illustration of the targeted area Γ.

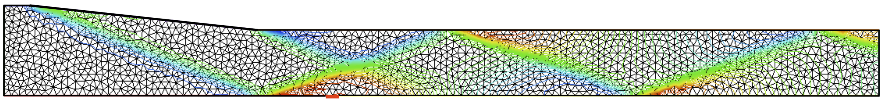
In this numerical experiment, the flow is modeled by the compressible Euler equations, cf. Section 5.2.1, meaning that some important phenomena such as shock-boundary layer interactions will not be treated as viscosity is neglected. Furthermore, the QoI is related to the pressure signature on the lower surface wall, namely we define it as the integrated pressure coefficient over a short segment: $j = \int_\Gamma \frac{p-p_\infty}{\frac{1}{2}\rho_\infty ||\boldsymbol{u}_\infty||^2}\, d\boldsymbol{x}$, where Γ is a small region of size $5.08mm$, located on the lower wall at $x = 116.08mm$ from the entrance, slightly downstream on the end of the inlet ramp, cf. Figure 5.2. It is thus interesting to analyze how the shocked flow and pressure distribution are impacted by operational and geometrical uncertainties: i.e. changes in the free stream Mach number M_∞ and in the ramp angle α. A change in the ramp angle will also affect its length, i.e. the boundary of the domain being modified. Palacios *et al.* [Palacios et al., 2012] analyzed this stochastic system with an identical configuration and obtained a discontinuous response surface in the QoI. The authors were interested in developing an adaptive deterministic mesh associated to one nominal condition, in order to obtain a representative reference value of the QoI for the entire variability range. Here, we have chosen uniformly distributed parameters with large variabilities: i.e. $\alpha \in \mathcal{U}_{[5.6;6.1]}$ (in degrees) and $M_\infty \in \mathcal{U}_{[3.5;5.5]}$. This is a challenging numerical problem since singularities arise in both physical and parameter spaces. Indeed, we can see from the study of some pressure fields associated to several conditions, as illustrated in Figure 5.3, how variations in the M_∞ and (to some extent) in the ramp angle α affect the shock train and pressure values along the computational domain and more specifically in the targeted zone Γ. One may also guess that depending on the combination between the flow speed and the angle of attack, the small area under interest may experience some pressure discontinuities (or not), strongly impeding on the numerical errors of the approximations. The solution fields shown in Figure 5.3 were computed on uniform meshes with about 1500 vertices. This mesh is not at all adapted to the flow field (notice the smeared out shocks), let alone to the specific quantity of interest.



(A) $M = 5.5$, $\alpha = 5.6$



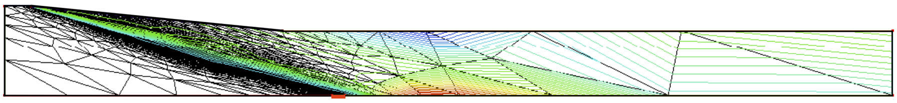
(B) $M = 4.78$, $\alpha = 5.94$



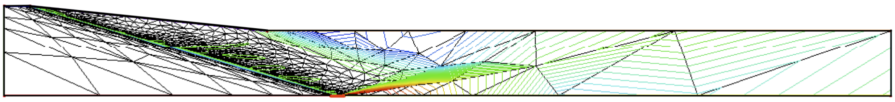
(C) $M = 3.5$, $\alpha = 6.1$

FIGURE 5.3: Overview of iso-density lines for some selected parameter sets on initial, non-adapted meshes.

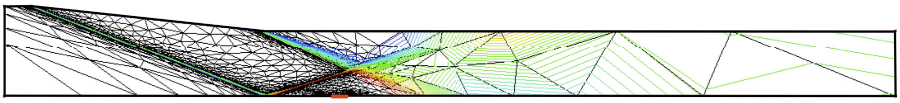
We first need to make sure we control the discretization error for a given flow speed and geometry. While it is obvious that a good “shock-capturing" method is needed for this problem, depending on the available computing resources, it is in general not recommended to refine all shocks present in the domain. Our proposed adaptive method based on optimal control of both stochastic and deterministic errors is a sound approach to make the right selection for refinements. Thanks to the adjoint-based method, the mesh is efficiently adapted only in the regions with large impact for our QoI. Representative examples of goal-based adapted meshes for various ramp angles and Mach numbers are displayed in Figure 5.4 (the target complexity there was $\mathcal{C}_{\mathbf{x}} = 1000$). With a closer look at the meshes, cf. Figure 5.5, we observe that the remeshing effort is solely focused in the regions impacting the QoI.



(A) $M = 5.5$, $\alpha = 5.6$



(B) $M = 4.78$, $\alpha = 5.94$

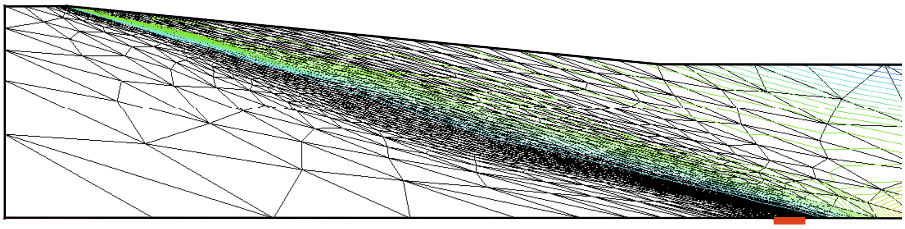


(C) $M = 3.5$, $\alpha = 6.1$

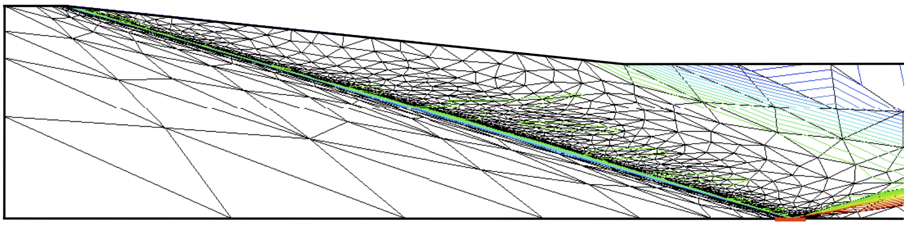
FIGURE 5.4: Overview of iso-density lines for the same parameter sets as shown in Figure 5.3, but this time a goal-based mesh adaptation was preformed on each mesh each mesh with a mesh complexity of $\mathcal{C} = 1000$.

The shock waves are now resolved much more accurately and one can now clearly see that Figures 5.4a, 5.4b and 5.4c represent three qualitatively distinct cases :

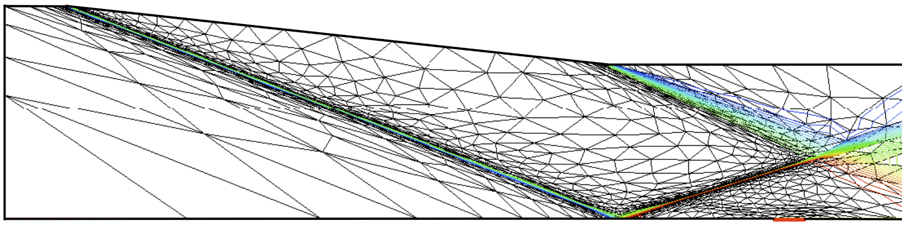
- Figure 5.4a and 5.5a: compression shock hits the lower wall downstream of the pressure sensor. The pressure at the sensor is not influenced resulting in $C_p = 0$. Even with a low mesh complexity, the error will be small.
- Figure 5.4b and 5.5b: compression shock hits the lower wall on the sensor. The pressure coefficient on the sensor will heavily depend on a accurate resolution of the shock for which appropriate mesh adaptation will be needed.
- Figure 5.4c and 5.5c: compression shock hits the lower wall upstream of the pressure sensor. Not only does one need to resolve the shock accurately enough to obtain an accurate solution field downstream of the shock where the pressure sensor is located, but one also needs to resolve the interaction between the reflected compression shock and the expansion fan. As can be seen in Figure 5.5c.



(A) $M = 5.5$, $\alpha = 5.6$, shock impinges on the wall downstream of the sensor.



(B) $M = 4.78$, $\alpha = 5.94$, shock impinges on the wall on the sensor.



(C) $M = 3.5$, $\alpha = 6.1$, shock impinges on the wall upstream of the sensor.

FIGURE 5.5: Zoom of the results shown before in Figure 5.4, showing the position where the shock hits the lower wall relative to the pressure sensor.

The resulting response surface is, as can be expected, discontinuous. In Figure 5.6, this response surface is visualized and the parameter sets corresponding to the solution fields discussed above are indicated.

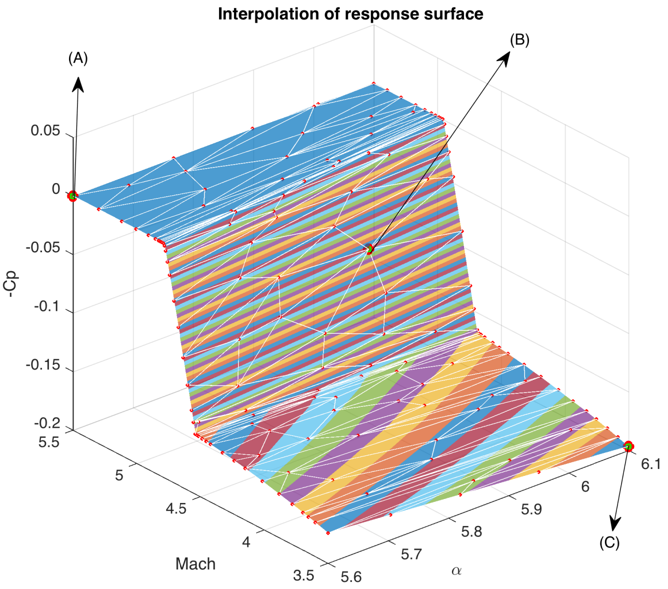
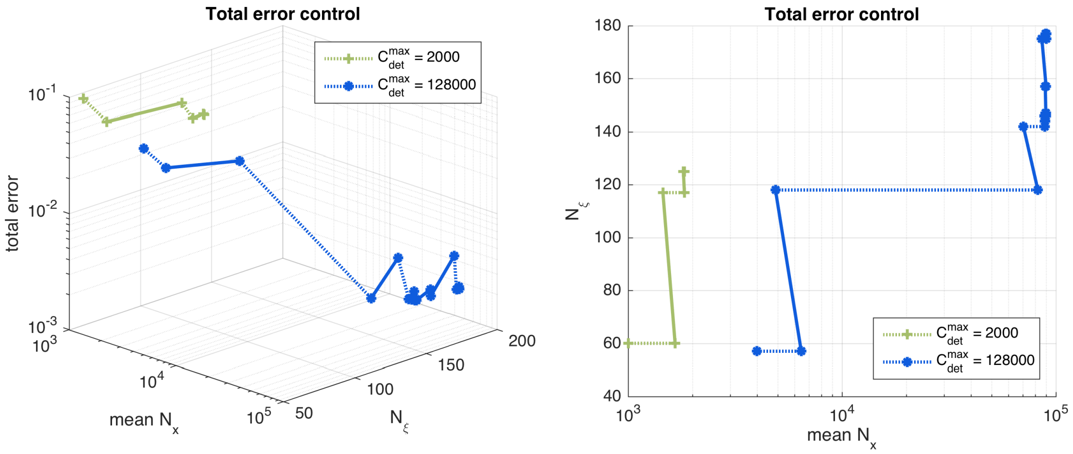


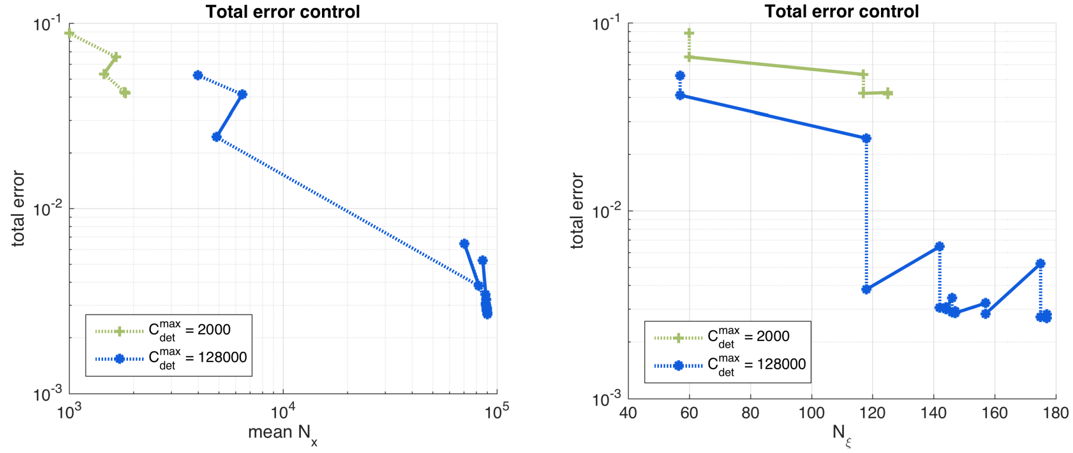
FIGURE 5.6: Response surface of the pressure coefficient on the sensor.

We have applied our optimal adaptive strategy described in Section 5.3.2 to control *both* stochastic and deterministic errors. Two computations were done, the first where $\mathcal{C}_{\mathbf{x}}^{max} = 2000$ and a second with $\mathcal{C}_{\mathbf{x}}^{max} = 128000$. The evolution of the total error for both cases is plotted in Figure 5.7 while the statistical moments and separated errors are given in Table 5.1 for the case where $\mathcal{C}_{\mathbf{x}}^{max} = 2000$ and in Table 5.2 for the case where $\mathcal{C}_{\mathbf{x}}^{max} = 128000$. The results in the tables and error convergence plots show the final results after each refinement step, which contains multiple fixed-point iterations.



(A) Evolution of the total error $\overline{\varepsilon}+\overline{\eta}$.

(B) Error plot shown from above.



(C) Error plot seen from the side: total error in function of $\overline{N}_{\mathbf{x}}$

(D) Error plot seen from the side: total error in function of N_{ξ}

FIGURE 5.7: Error control by adaptive refinement (multiple views of the same plot). The total error is controlled by either refining the mesh complexity each deterministic computation using Algorithm 2 (dotted lines) or by increasing the mesh complexity in the parameter space following Algorithm 1 (full lines). In the first computation the maximal mesh complexity in each deterministic computation was set to $\mathcal{C}_{\mathbf{x}}^{max}=2000$ while in a second computation it was set to 128000.

Setting a maximal complexity $\mathcal{C}_{\mathbf{x}}^{max}$ is not required, this limitation was used here for the practical purpose of avoiding a cost exceeding our computational resources.

From Figure 5.7 one can see that the total error for the case with $\mathcal{C}_{\mathbf{x}}^{max}=2000$ stagnates at a much higher level than for the case $\mathcal{C}_{\mathbf{x}}^{max}=128000$. This is cause by the fact that initially, the deterministic error dominates and the constraint $\mathcal{C}_{\mathbf{x}}^{max}=2000$ impedes $\overline{\varepsilon}$ from descending below $\overline{\eta}$. Step 0 in Table 5.1 corresponds to the first point on the green curve in Figure 5.7 with the coordinates $(N_{\boldsymbol{\xi}}, \text{mean } N_{\mathbf{x}})=(30,1000)$ which corresponds to the initial mesh (a Delaunay triangulation of 30 randomly drawn points). On this initial mesh, the stochastic error estimate is not yet available so by default a refinement step of the stochastic mesh is done. At that point, called step 1, one can compare $\overline{\varepsilon}$ and $\overline{\eta}$; since $\overline{\varepsilon}$ dominates Algorithm 2 is called resulting in an increase of the average deterministic complexity from 1000 to 1657. Still, however, the deterministic error dominates (see step 2 in Table 5.1) and the constraint on the deterministic complexity makes it impossible to significantly reduce $\overline{\varepsilon}$ further. In order to show the reader the existence of the plateau caused

Step nr.	N_{ξ}	mean $N_{\mathbf{x}}$	$\mathbb{E}[-Cp]$	Var$[-Cp]$	$\bar{\varepsilon}$	$\bar{\eta}$
0	30	1000	-0.10572691	0.00636237	0.0591	
1	60	1000	-0.10545861	0.00626342	0.0641	0.0242
2	60	1657	-0.10545130	0.00626541	0.0415	0.0242
3	117	1454	-0.10451489	0.00624180	0.0508	0.0022
4	117	1832	-0.10452315	0.00624091	0.0398	0.0022
5	125	1811	-0.10450733	0.00624493	0.0407	0.0018
6	125	1836	-0.10450885	0.00624503	0.0400	0.0018

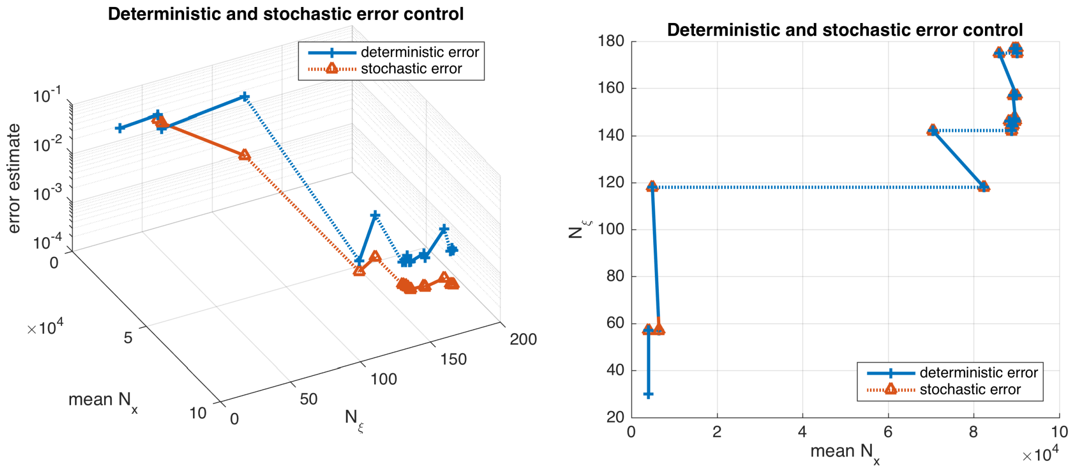
TABLE 5.1: Inlet statistics and errors with $\mathcal{C}_x^{max} = 2000$.

Step nr.	N_{ξ}	mean $N_{\mathbf{x}}$	$\mathbb{E}[-Cp]$	Var$[-Cp]$	$\bar{\varepsilon}$	$\bar{\eta}$
0	30	4000	-0.10584943	0.00633727	0.0253	
1	57	4000	-0.10521272	0.00622762	0.0286	0.0236
2	57	6446	-0.10521254	0.00622689	0.0175	0.0236
3	118	4885	-0.10443541	0.00626330	0.0229	0.0014
4	118	82373	-0.10443116	0.00626273	0.0024	0.0014
5	142	70471	-0.10441939	0.00626304	0.0057	0.0008
6	142	88849	-0.10441761	0.00626286	0.0023	0.0008
7	144	88708	-0.10441652	0.00626417	0.0023	0.0007
8	144	89294	-0.10441650	0.00626417	0.0023	0.0007
⋮	⋮	⋮	⋮	⋮	⋮	⋮
18	177	89351	-0.10441013	0.00626303	0.0024	0.0004
19	177	90096	-0.10440860	0.00626287	0.0022	0.0004

TABLE 5.2: Inlet statistics and errors with $\mathcal{C}_x^{max} = 128000$.

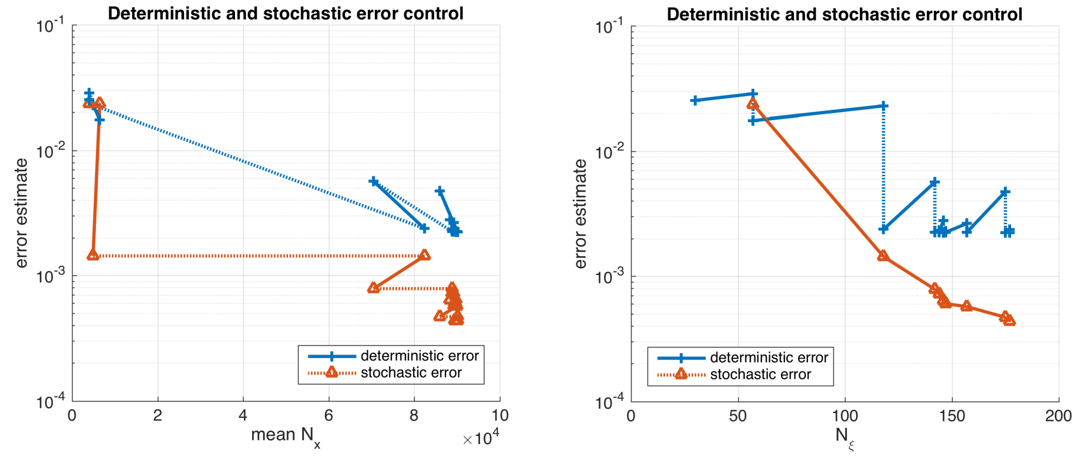
by this constraint, from this point on forced refinements alternating between the stochastic and deterministic space were mandated. During each stochastic refinement step, new samples are added and for each new sample the deterministic complexity $\mathcal{C}_{\mathbf{x}} = 1000$ is set by default. Then follows a deterministic refinement step that will, if necessary, refine these computations.

For the case where $\mathcal{C}_{\mathbf{x}}^{max} = 128000$ the error decomposed into $\bar{\varepsilon}$ and $\bar{\eta}$ is plotted in Figure 5.8, there one can see that $\mathcal{C}_{\mathbf{x}}^{max} = 128000$ does permit the deterministic error to dive below the stochastic error. As for the case with $\mathcal{C}_{\mathbf{x}}^{max}$, the initial stochastic mesh is made up of 30 samples but this time the default deterministic complexity is set to 4000. The error estimate ε contained in each sample is given in the deterministic error map Figure 5.9, notice the link between the error map and the features of the actual response surface (Figure 5.6). This is even more pronounced in Figure 5.10 where the error map of step 1 is shown. In the part of the parametric domain where the Mach number - inlet angle couple makes the shock impinge upon the sensor, the error has its maximum. When the shock falls behind the sensor, ans the Cp is zero, the error is very small, in the order of 10^{-6}, and where the shock hits the wall upstream of the sensor, but not on the sensor, the error lies in between.



(A) Evolution of the deterministic error $\overline{\varepsilon}$ and the stochastic error $\overline{\eta}$.

(B) Error plot shown from above.



(C) $\overline{\varepsilon}(\overline{N}_{\mathbf{x}})$ and $\overline{\eta}(\overline{N}_{\mathbf{x}})$

(D) $\overline{\varepsilon}(N_{\boldsymbol{\xi}})$ and $\overline{\eta}(N_{\boldsymbol{\xi}})$

FIGURE 5.8: Behaviour of $\overline{\varepsilon}$ and $\overline{\eta}$ for the supersonic inlet test case with $\mathcal{C}_{\mathbf{x}}^{max} = 128000$. Dotted lines indicate deterministic refinement following Algorithm 2 while full lines represent refinement in the parametric space following Algorithm 1.

The error and complexity maps of of steps 0 and 1 (Figures 5.9 and 5.10) that most new samples are added in the region of the parametric domain where the first oblique shock hits the lower wall on the sensor. The deterministic error in these samples is also maximal in this region. In order to compute to within the target accuracy the pressure coefficient on the sensor for these samples, mesh refinement will be needed. At the same time, in the parametric domain, this region corresponds to a discontinuity which, in order to be resolved to within target accuracy, needs mesh refinement in the stochastic space.

After an initial refinement in the stochastic space, comparing $\overline{\varepsilon}$ and $\overline{\eta}$ (step 1 in Table 5.2) shows that the deterministic error dominates and Algorithm 2 ensures that the necessary computations are refined such that $\overline{\varepsilon}$ becomes smaller than or equal to the deterministic target error $\overline{\varepsilon}^t$. Comparing Figures 5.10 and 5.11 one can see how Algorithm 2 has increased the complexity the most for the samples with the highest deterministic error estimate. During this refinement step the mean deterministic complexity increases from 4000 to 6446 while the number of samples $N_{\boldsymbol{\xi}}$ stays the same. The expectation and variance of the stochastic response however may change from step 1 to 2 as now the solution field in the refined deterministic computations is more accurate thus the QoI, the pressure coefficient on the sensor, at these sample points may change.

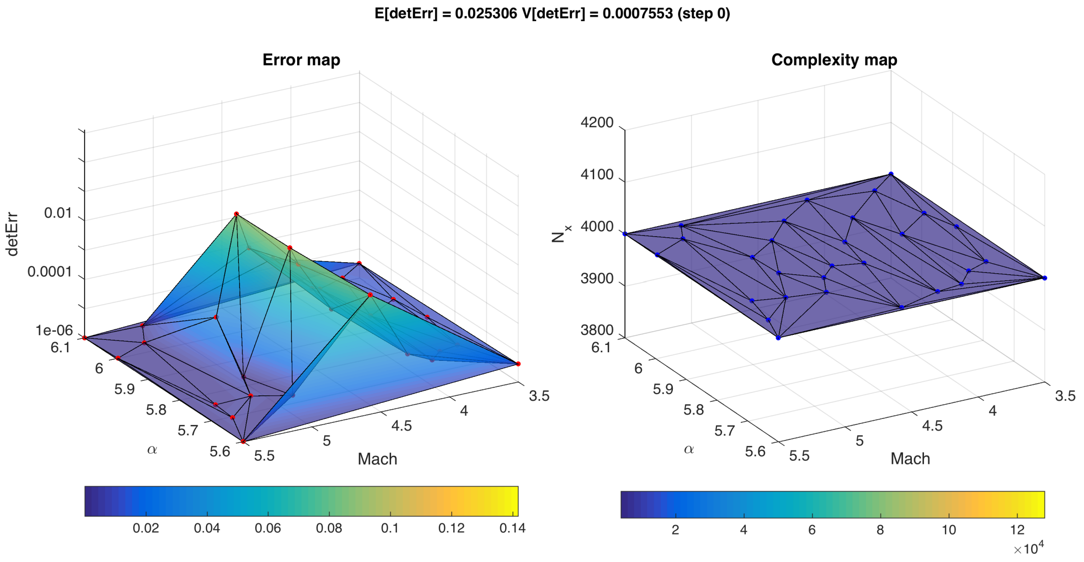


FIGURE 5.9: Inlet step 0: Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space.

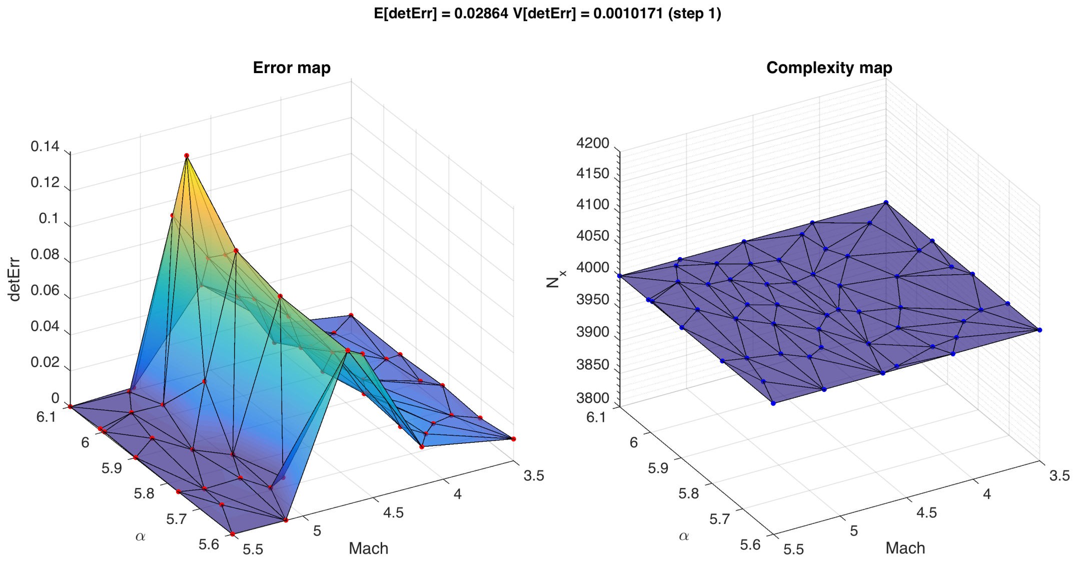


FIGURE 5.10: Inlet step 1: Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space.

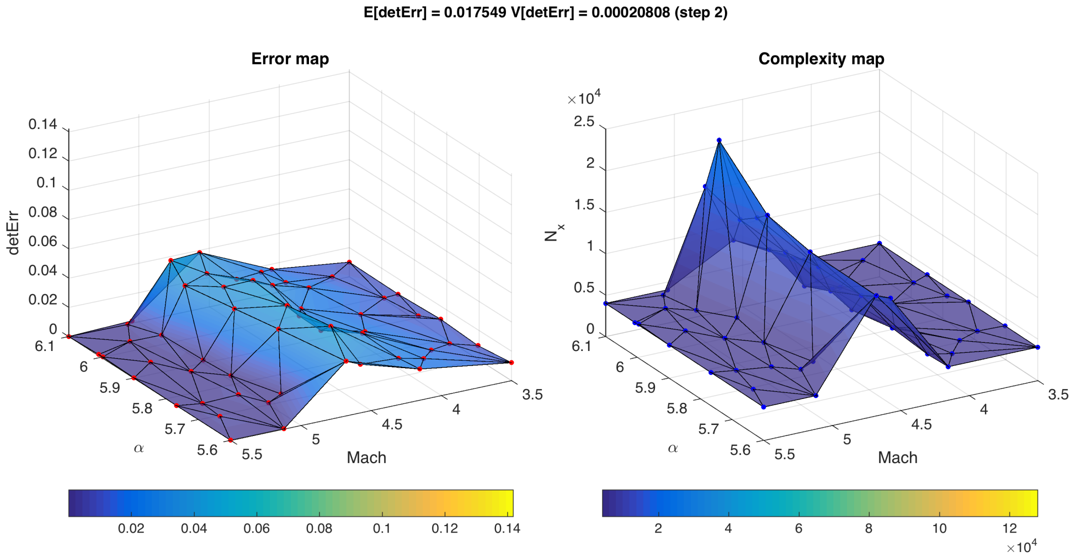


FIGURE 5.11: Inlet step 2: Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space.

After completion of step 2, the dominating error is $\overline{\eta}$. The next refinement step will thus be a stochastic refinement step where new samples are added using metric-based mesh refinement.

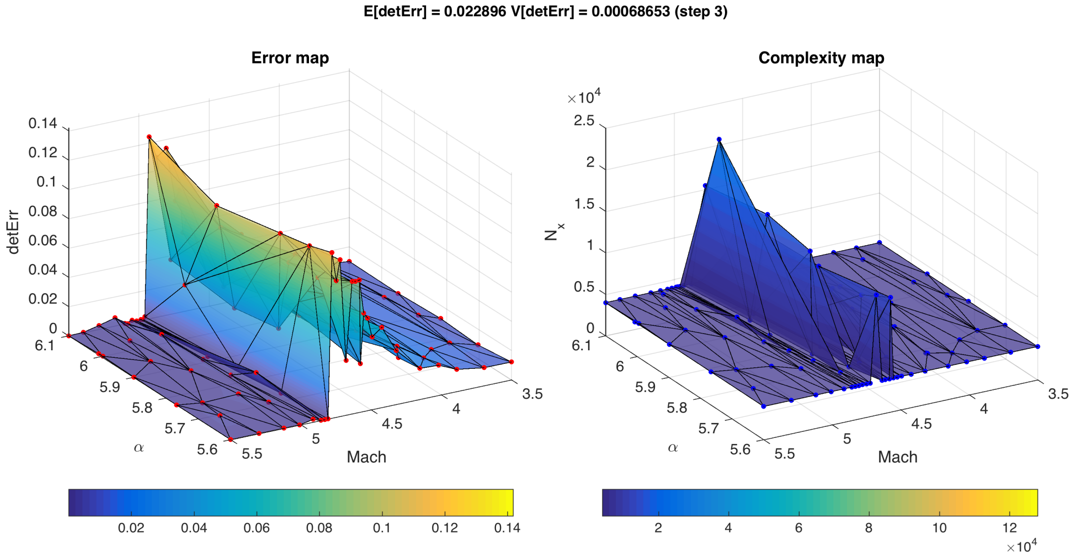


FIGURE 5.12: Inlet step 3: Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space.

The introduction of new samples during step 3 has lowered $\overline{\eta}$ from 0.0236 to 0.0014 (see Table 5.2) but it has also lead to an increase of $\overline{\varepsilon}$ from 0.0175 to 0.0229 due to the fact that all new samples have been computed using the default complexity $C_{\boldsymbol{\xi}} = 4000$. With the deterministic error now again dominating, a deterministic refinement step is started with the goal of reaching a target error $\overline{\varepsilon}^t$ that is below the current $\overline{\eta}$.

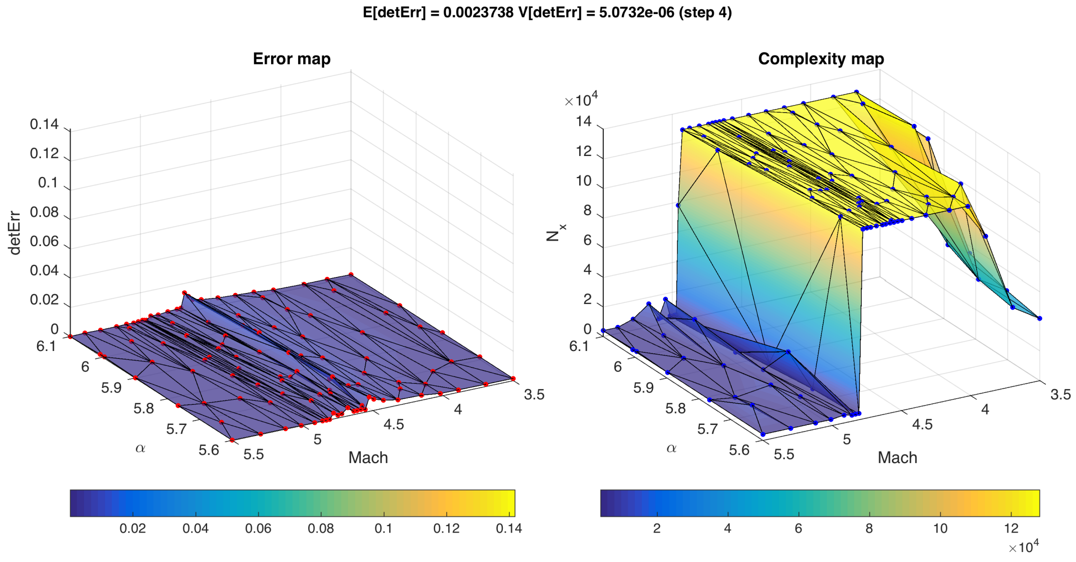


FIGURE 5.13: Inlet step 4: Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space.

This goal, however, is not reached. At the end of step 4 the deterministic error $\overline{\varepsilon}$ has been reduced to 0.0024, but the constraint on the maximum deterministic complexity impedes it from getting below the current stochastic error $\overline{\eta} = 0.0014$. In Figure 5.13 the error and complexity maps at step 4 is given, where the latter shows a plateau of of all the computations that were pushed to the maximal allowed deterministic complexity.
From this point on, forced alternate refinement was used. The result is that each time a few samples are added at default complexity, the deterministic error rises, and in the subsequent deterministic refinement steps, these new samples are pushed to a higher complexity in order to reduce $\overline{\varepsilon}$ as much as possible. However, as is clear form looking at Figure 5.8d, $\overline{\varepsilon}$ reaches a plateau, due to the constraint on the maximal complexity, while $\overline{\eta}$ does continue do descend. With the dominating deterministic error stagnating, the fact that $\overline{\eta}$ still goes down, means that the total error will still diminish but will never get below the current deterministic error (see Figure 5.7d). This alternating refinement was forced to continue up to step 19.

An overview of the adaptation process described above can be found in Figure 5.14 where results at the final sub-iteration of adaptation step 0 (i.e. at the initial stage), step 2 and final step 19 are shown. As was previously observed, we see that the integrated C_p response (cf. second row) in the parametric space is quite regular except for two sharp oblique transition regions of varying M_∞ and α, corresponding to the impingement of the inlet first compression shock onto the region of interest Γ. The metric-based stochastic adaptation is able to sense these regions of poor smoothness. It gradually and anisotropically refines the approximation by adding more samples in these areas.

5.4.2 Scramjet internal flow

The coupled control of both deterministic and stochastic errors is now tested on the scramjet problem introduced in Chapter 2. The geometry is recalled in Figure 5.15, and the QoI

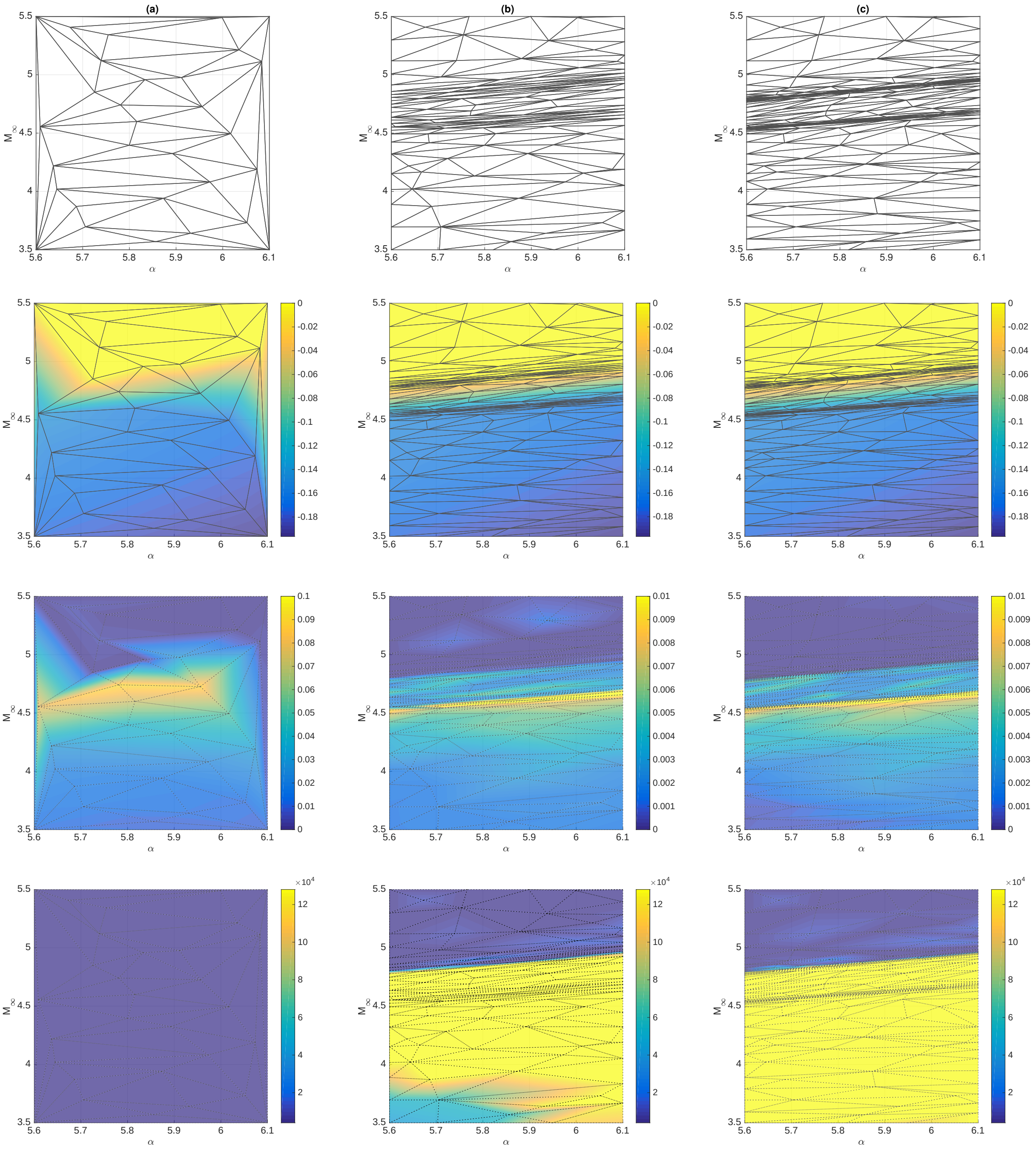


FIGURE 5.14: Inlet problem: some results of the coupled adaptation strategy for error control. Results are displayed at the end of step 0 (column (a)), step 2 (b) and step 19 (c). Adapted stochastic partitions (first row) and isocontour maps are shown in the parametric space: the surface response of the approximated QoI $j(M_\infty, \alpha)$ (2^{nd} row), – spatial discretization error maps $\varepsilon(M_\infty, \alpha)$ (3^{rd} row) and – corresponding optimal spatial complexity map $\mathcal{C}_{\boldsymbol{x}}(M_\infty, \alpha)$ (bottom row).

is the pressure signature on Γ:

$$j(w) = \int_\Gamma \left(\frac{p - p_\infty}{p_\infty} \right)^2 dy, \tag{5.29}$$

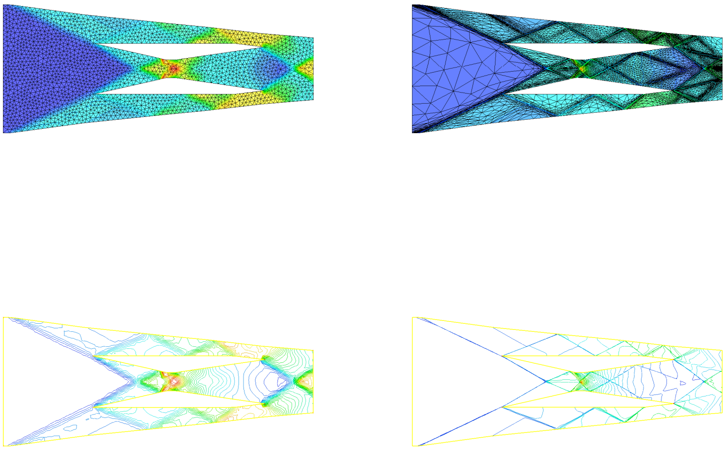


FIGURE 5.16: Scramjet: Density field solution for $M_\infty = 3$ and $\alpha = 0$: uniform mesh (left image) vs. goal-based adapted mesh (right image)

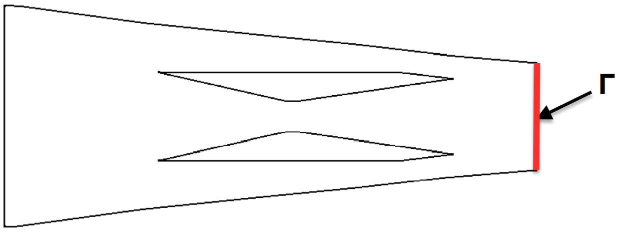


FIGURE 5.15: Scramjet geometry; the QoI is integrated on Γ.

The internal flow in the scramjet is modeled by the Euler equations, viscosity is neglected in this study, and there are two uncertain parameters: the angle of attack α and the free stream Mach number M_∞. As before, both are assumed to follow a uniform distribution with $\alpha \sim \mathcal{U}_{[0,6]}$ degrees and $M_\infty \sim \mathcal{U}_{[2.4,4]}$ respectively. The effect of the goal-based deterministic refinement is demonstrated in Figure 5.16 where density field is shown for a uniform mesh on the left side and an adapted mesh on the right for one configuration (M_∞, α). In contrast to the previous treatment given to this test case, using a robust LOO-weighted Least Squares method to construct a continuous representation of the response surface where the approximation errors contained in each deterministic computation were seen as noise, here this approximation error is computed for each deterministic sample and the expectation of those errors $\overline{\varepsilon}$ and compared to the expected error on the parametric domain $\overline{\eta}$. The coupled refinement strategy outlined in Algorithm 3 will ensure that the refinement will take place in the space where the error is dominant.

Three response surfaces with increasing number of vertices are shown in Figure 5.17, one can compare these response surfaces to the ones obtained in Figure 2.12 in Chapter 2.

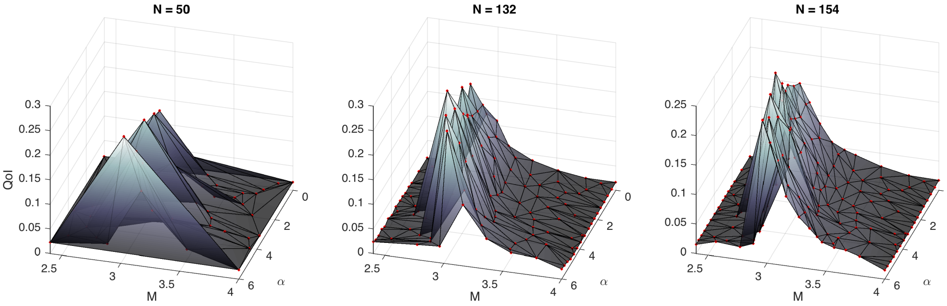
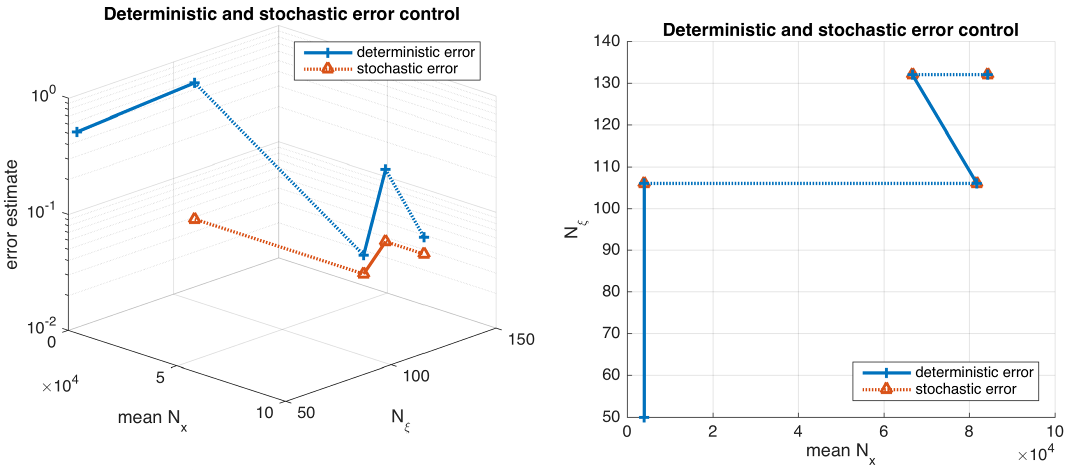


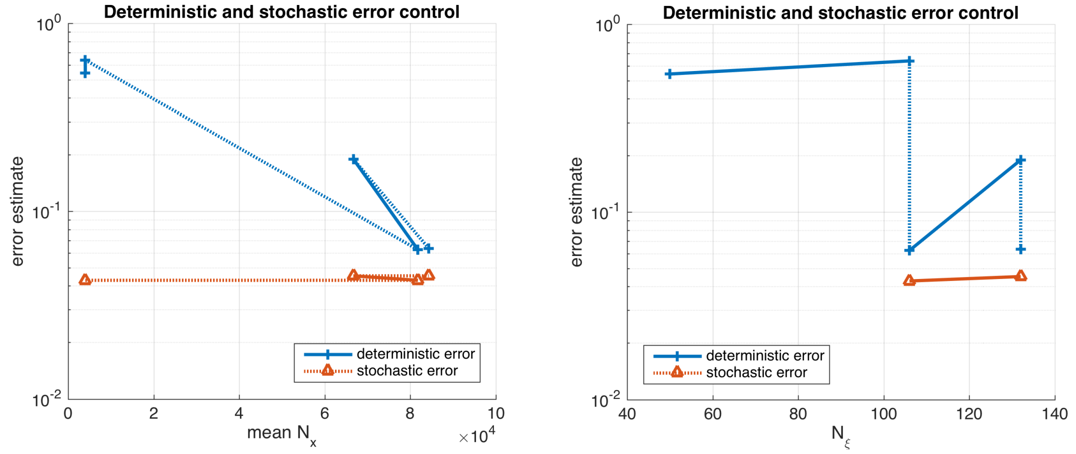
FIGURE 5.17: Response surfaces of the QoI for the scramjet case. On the left, the Delaunay triangulation on the randomly drawn samples; in the middle the response surface corresponding to the first data point in Figure 5.18; and on the right the final mesh corresponding to the last data point in aforementioned figures.

In Figure 5.18 the evolution of the deterministic error $\overline{\varepsilon}$ and the stochastic error $\overline{\eta}$ are shown in function of the number of samples $N_{\boldsymbol{\xi}}$ and the mean deterministic mesh complexity. The full lines in the plot indicate a adaptation of the stochastic response surface and the dotted lines indicate that Algorithm 2 was used for a refinement of the deterministic computations. The subplots in Figure 5.18 each show a different view of the same plot.



(A) Evolution of the deterministic error $\overline{\varepsilon}$ and the stochastic error $\overline{\eta}$.

(B) Error plot shown from above.



(C) $\overline{\varepsilon}(\overline{N}_{\mathbf{x}})$ and $\overline{\eta}(\overline{N}_{\mathbf{x}})$

(D) $\overline{\varepsilon}(N_{\boldsymbol{\xi}})$ and $\overline{\eta}(N_{\boldsymbol{\xi}})$

FIGURE 5.18: Behaviour of $\overline{\varepsilon}$ and $\overline{\eta}$ for the scramjet test case with $\mathcal{C}_{\mathbf{x}}^{max} = 128000$. Dotted lines indicate deterministic refinement following Algorithm 2 while full lines represent refinement in the parametric space following Algorithm 1.

The deterministic error line has five markers, while the stochastic error line only four; this is due to the fact that on the initial stochastic mesh no stochastic error estimate is available. This initial mesh (called step 0) is the Delaunay triangulation of 50 samples drawn according to the pdf of the two random variables α and M_∞. At least one fixed-point iteration of the stochastic mesh adaptation procedure is needed to generate an error estimate $\overline{\eta}$. At this point we find ourselves in Figure 5.18b at the point $(\text{mean}N_{\mathbf{x}}, N_{\boldsymbol{\xi}}) = (4000, 106)$ (step 1). It can be seen in Figure 5.18a that at these coordinates, the deterministic error $\overline{\varepsilon}$ dominates. Algorithm 3 then orders a deterministic refinement to be done and calls Algorithm 2, the result of which is visualised by the dotted lines starting from the coordinates $(\text{mean}N_{\mathbf{x}}, N_{\boldsymbol{\xi}}) = (1000, 106)$ and ending at $(81767, 106)$ (step 2). At the end of this adaptation cycle, the mean mesh complexity of the deterministic computations has been increased from 4000 to 81767, and the deterministic error estimate $\overline{\varepsilon}$ has decreased from 0.64 to 0.062 the number of samples $N_{\boldsymbol{\xi}}$ has remained the same as did $\overline{\eta}$. Now, at step 2, the deterministic error still dominates, but further refining the deterministic computations no longer reduces $\overline{\varepsilon}$ because of the limitation $\mathcal{C}_{\mathbf{x}}^{max} = 128000$. In Figures 5.19, 5.21 and 5.22 a map of the complexity of each CFD computation $\mathcal{C}_{\mathbf{x}}(\boldsymbol{\xi})$ and its corresponding error estimate $\varepsilon(\boldsymbol{\xi})$ is shown on the parametric domain.

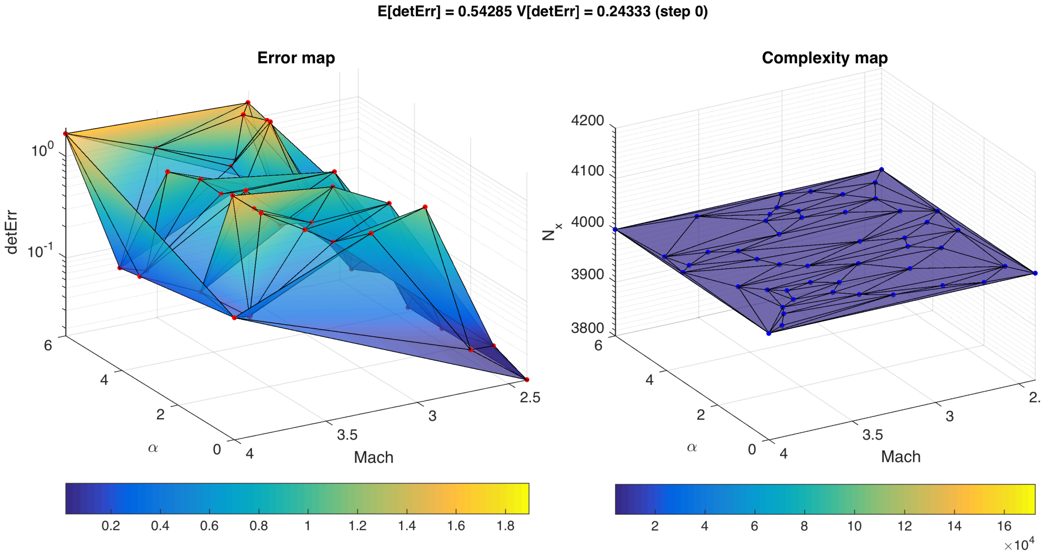


FIGURE 5.19: Map showing the error on each deterministic computation and the mesh complexity of those computations for the scramjet case on the initial Delaunay triangulation on the parameter space.

The initial complexity for all deterministic computations was set at 4000, as can be seen on the right in Figure 5.19. While the complexity in each of the deterministic computations is the same, the error is not. The angle of attack and free stream Mach number have an important effect on the flow solution, as the random variables change, the number of shocks, their strength and interactions all change. The result is that for some computations, especially those where the free stream Mach number is low and, a complexity of only 4000 will be enough to achieve an error of the order of 10^{-1} while for other computations, a much higher complexity will be required. In the former case, at free stream Mach numbers in the approximate range $[2.4, 2.7]$, the complex system of interacting shocks as shown in Figure 5.16 is not yet in place and mesh complexities of 4000 suffice to achieve the target error. As an illustration of the effect of the goal-based adaptation, Figure 5.20 shows, for $M_\infty = 3$ and $\alpha = 0$, how the refinement reveals much more detailed features of the flow solution. Without the mesh refinement, many of these intricate features remain unresolved.

From step 0 (Figure 5.19) to step 1 (Figure 5.21) we see the effect one fixed-point iteration of the adaptation of the mesh on the parametric space.The complexity for each deterministic computations remains at 4000, but the number of samples increases.

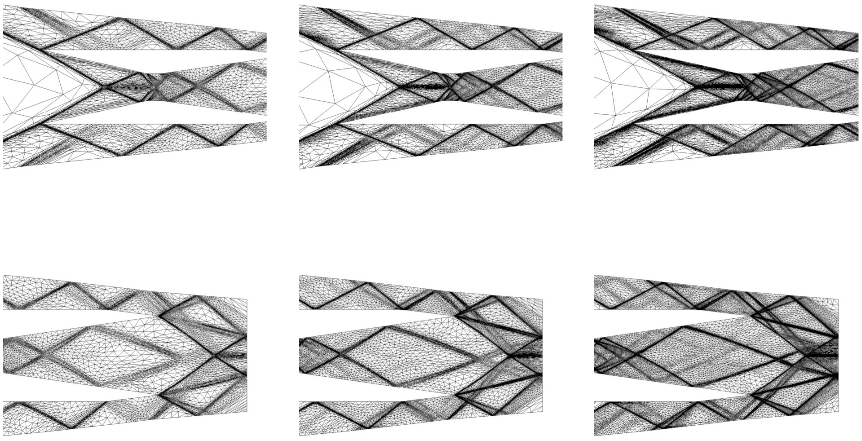


FIGURE 5.20: Scramjet: Deterministic adapted meshes at $M_\infty = 3$. By line: zoom in the middle of the geometry vs. zoom at the outflow, for deterministic complexity $\mathcal{C}_{\boldsymbol{x}} = 8000$, 16000 and 32000.

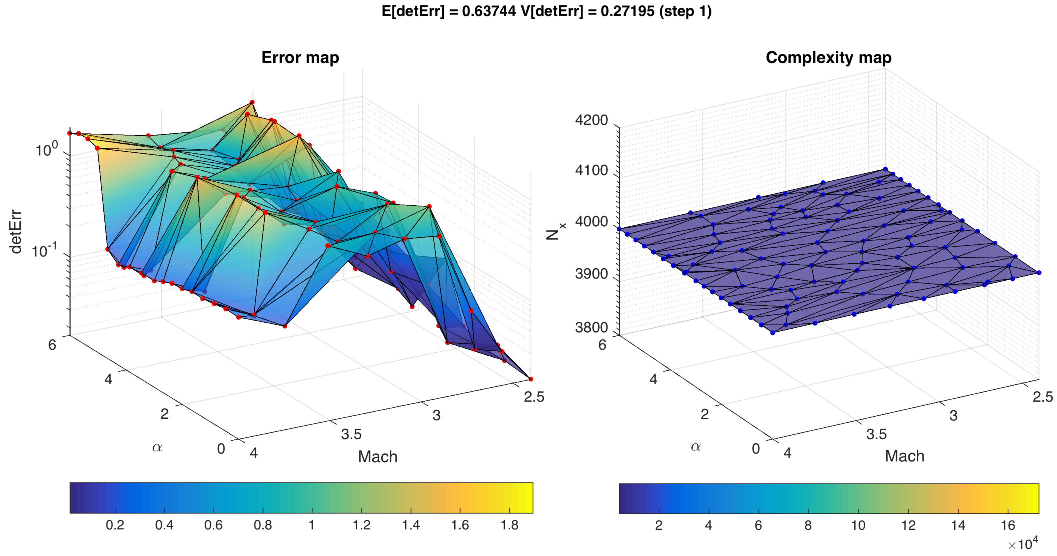


FIGURE 5.21: Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space for the scramjet case.

Going from step 1 (error map shown in Figure 5.21) to step 2 (error map given by Figure 5.22) one can see the effect of the deterministic refinement cycle. Following Algorithm 2, for each sample (which corresponds to a CFD computation), unless the deterministic error contained in that sample is already below the global deterministic target error, an individual target error is set, based on which the target complexity for that sample is computed. One can see on the left in Figure 5.22 that the error has dropped by an order of magnitude when compared to Figure 5.21. At the same time, the complexity map on the right in Figure 5.22 shows two plateaus: a small plateau for $N_{\mathbf{x}} \approx 4000$ at the low end of the M_∞ range and a larger plateau with $N_{\mathbf{x}} \approx 128000$ towards the higher Mach numbers. The plateau at $N_{\mathbf{x}} \approx 4000$ are the computations for which no goal-based mesh refinement was needed; the plateau at $N_{\mathbf{x}} \approx 128000$ represents all the computations that were given the maximum complexity allowed.

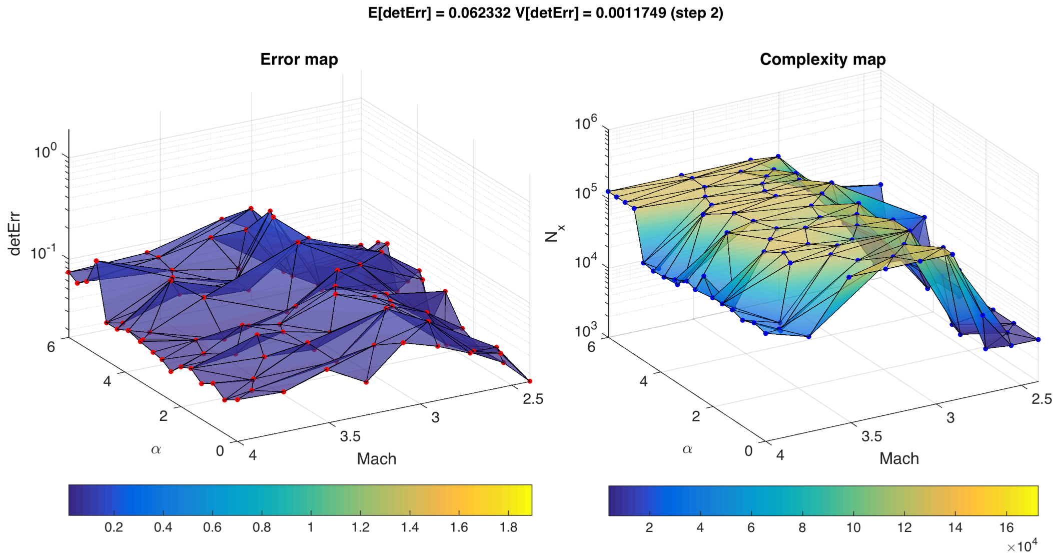


FIGURE 5.22: Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space for the scramjet case.

The errors and complexities shown in Figure 5.22 are the result of 4 iterations of Algorithm 2. Due to the constraint set on the maximum mesh complexity allowed, the deterministic error $\overline{\varepsilon}$ has not been reduced beyond the stochastic error $\overline{\eta}$. It is still the deterministic error that dominates, and in order to reduce the total error in the most effective way, one needs to refine the deterministic computations. Normally the refinement procedure would end here, but in order to show the effect of stochastic refinement the computation was forced to execute a refinement of the stochastic mesh. In that procedure, the new samples introduced are given the initial mesh complexity of 4000, which causes the error and complexity maps shown in Figure 5.23 to become somewhat spiked.

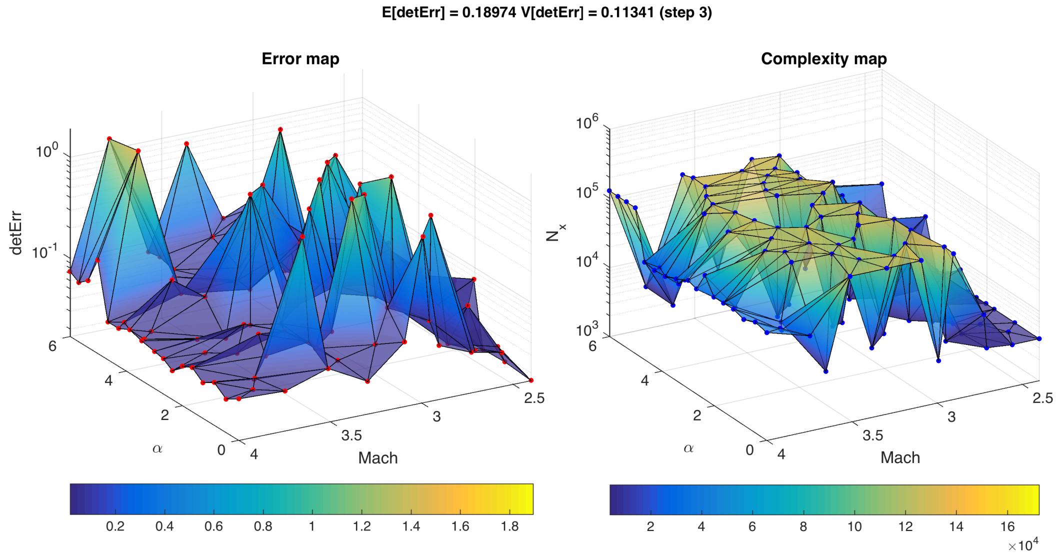


FIGURE 5.23: Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space for the scramjet case.

As the stochastic refinement step has introduced new computations at a low complexity, the mean complexity has now decreased and the deterministic error has increased. Another deterministic refinement step will again increase the mean deterministic mesh complexity and reduce $\overline{\varepsilon}$ to the same order of magnitude as was achieved in step 2. However, due to the limitation on $\mathcal{C}_{\mathbf{x}}^{max}$ is is impossible to reduce $\overline{\varepsilon}$ below $\overline{\eta}$.

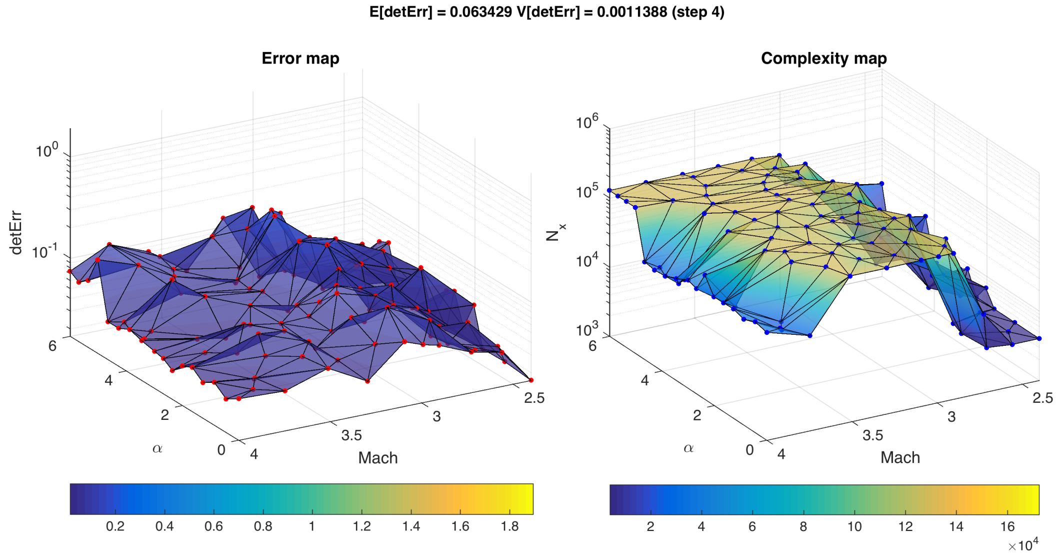


FIGURE 5.24: Map showing the error on each deterministic computation and the mesh complexity of those computations on the parameter space for the scramjet case.

An overview of the mean and variance of the response surface throughout this coupled refinement procedure is summarized in Table 5.3; the table also includes $\overline{\varepsilon}$, clearly showing how an increase in the mean $N_{\mathbf{x}}$ impacts the deterministic error estimate.

Step nr.	$N_{\boldsymbol{\xi}}$	mean $N_{\mathbf{x}}$	μ_j	σ_j^2	$\overline{\varepsilon}$	$\overline{\eta}$
0	50	4000	0.0491	0.0033	0.5429	
1	106	4000	0.0514	0.0029	0.6374	0.0428
2	106	81767	0.0533	0.0035	0.0623	0.0428
3	132	66667	0.0561	0.0042	0.1897	0.0426
4	132	84300	0.0547	0.0040	0.0634	0.0426

TABLE 5.3: Scramjet statistics and errors.

5.4.3 NACA0012

The example focusses on a NACA0012 geometry, the flow is inviscid with the inflow Mach number and the angle of attack as uncertain parameters; both follow a lognormal distributed such that $\mu_{M_\infty} = 0.8$ with a coefficient of variation $CoV_{M_\infty} = 1\%$ and $\mu_\alpha = 1.5$ degrees with $CoV_\alpha = 10\%$. The scalar quantity of interest $j(\boldsymbol{\xi})$ is the value of the Mach number at $x = 0.65c$ with c the chord length on the upper surface of the airfoil. The geometry and the location of the pressure sensor are shown in Figure 5.25.

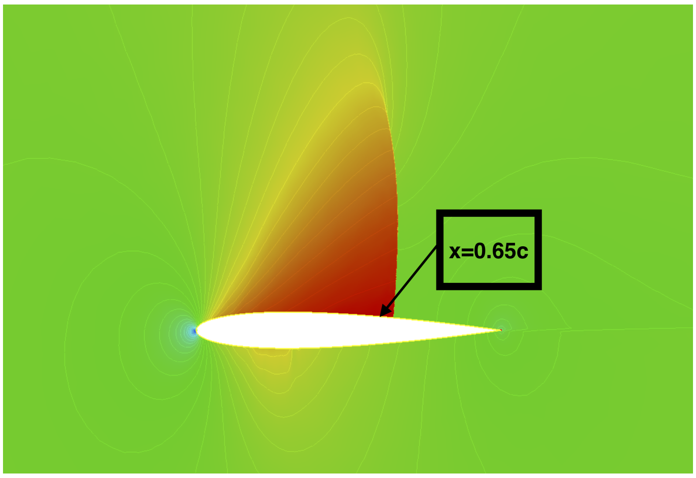
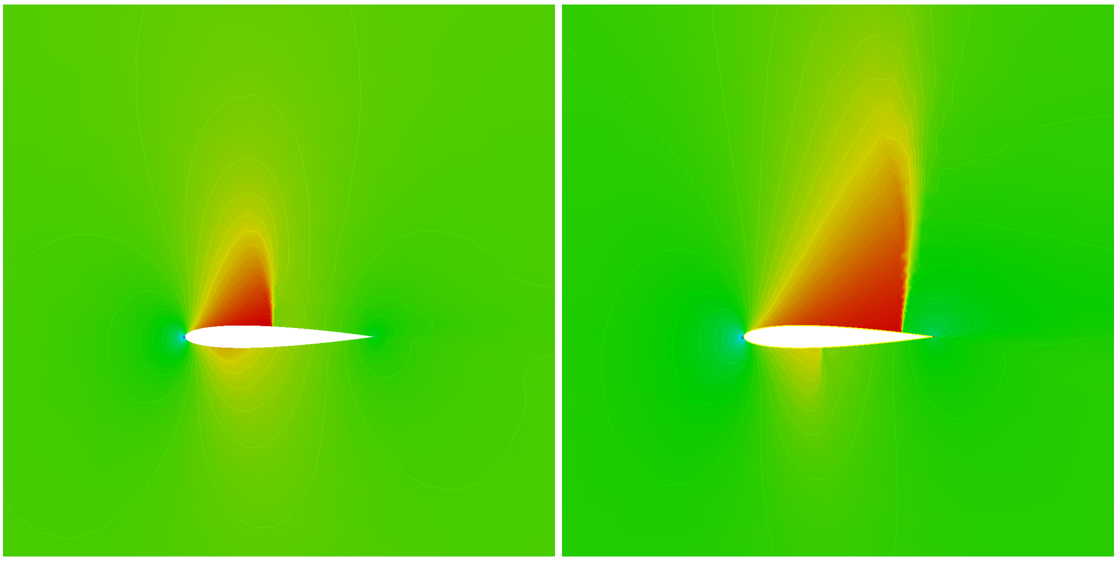


FIGURE 5.25: NACA0012 with pressure sensor.

In this transonic regime, the solution field is highly sensitive to small changes in the flow conditions or to changes in the geometry. This is demonstrated in Figures 5.26a and 5.26b where the Mach field for two different samples in the parameter space are shown. Mach field values range from subsonic to supersonic, with a shock forming on the upper surface of the airfoil. The position and strength of the shock are very sensitive to the random variables; when the shock traverses the pressure sensor, a sudden change in the QoI produces a discontinuity in the response surface.



(A) $M_\infty = 0.77$ and $\alpha = 1$

(B) $M_\infty = 0.83$ and $\alpha = 2$

FIGURE 5.26: NACA0012: Mach field for $M_\infty = 0.77$ and $\alpha = 1$ (A), vs. Mach field for $M_\infty = 0.83$ and $\alpha = 2$ (B).

A sequence of response surfaces along with the pdf of the random variables shows how this discontinuity is gradually resolved whilst taking into account the pdf in the placement of new samples.

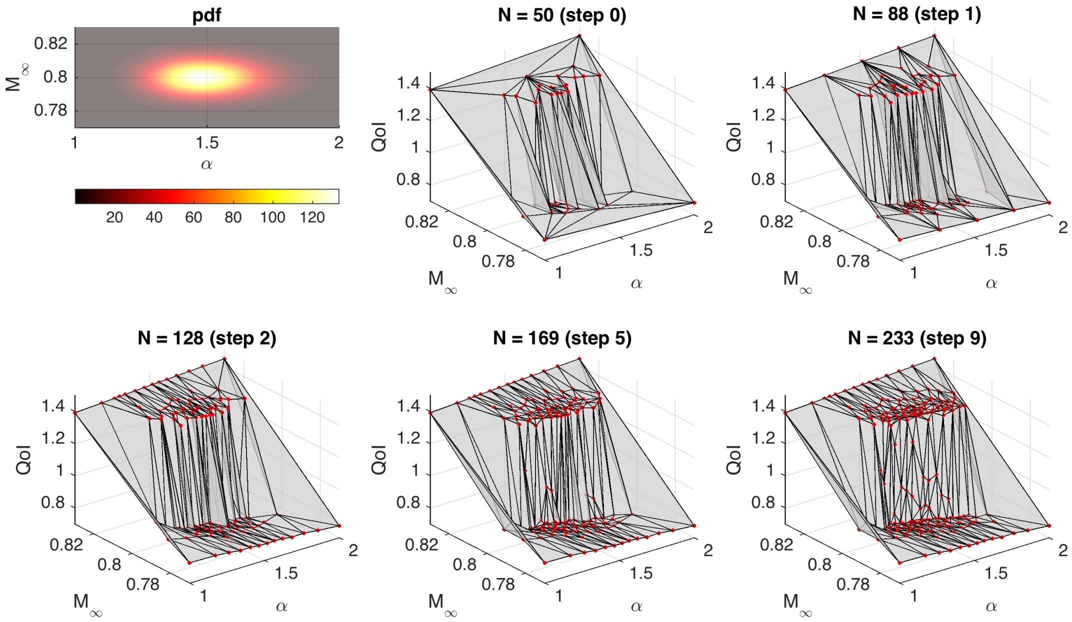


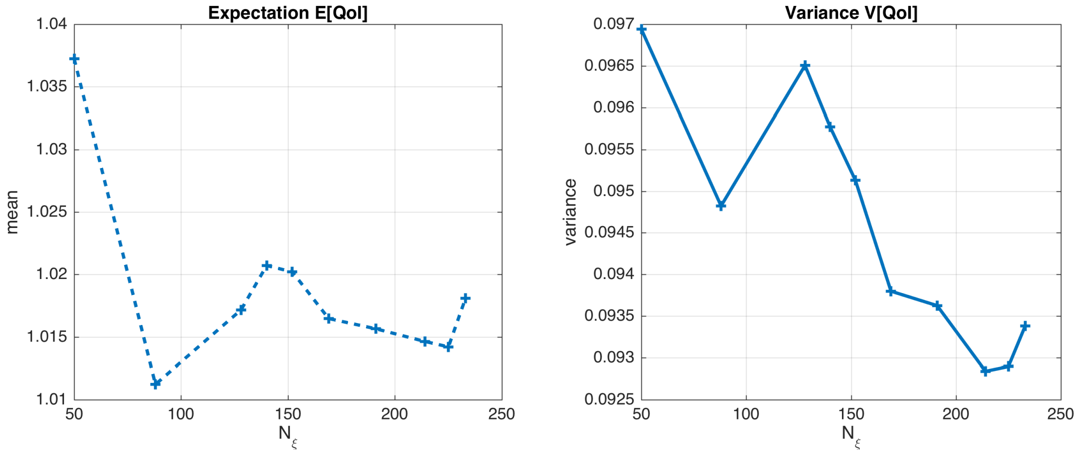
FIGURE 5.27: NACA0012 with pressure sensor.

For this test case, the initial stochastic DoE contains 50 samples, and the initial complexity for the deterministic computations is 4000. From the start, the deterministic error contribution is of the order of $\mathcal{O}(10^{-3})$ while the stochastic error contribution is much higher at $\mathcal{O}(10^{-1}$. The maximal number of refinement steps was (arbitrarily) set at 9, and different error contributions can be observed in Table 5.4.

Step nr.	$N_{\boldsymbol{\xi}}$	mean $N_{\mathbf{x}}$	$\overline{\varepsilon}$	$\overline{\eta}$
⋮	⋮	⋮	⋮	⋮
3	140	3993	$0.119{\cdot}10^{-3}$	0.279
4	152	3993	$0.118{\cdot}10^{-3}$	0.262
5	169	3993	$0.118{\cdot}10^{-3}$	0.157
6	191	3993	$0.118{\cdot}10^{-3}$	0.153
7	214	3993	$0.118{\cdot}10^{-3}$	0.144
8	225	3993	$0.118{\cdot}10^{-3}$	0.130
9	233	3993	$0.118{\cdot}10^{-3}$	0.119

TABLE 5.4: NACA0012: Evolution of deterministic and stochastic error components.

The evolution of the expectation and variance of the QoI is plotted in Figure 5.28 where a convergence up to an accuracy of 10^{-2} becomes apparent.



(A) Expectation (B) Variance

FIGURE 5.28: NACA0012: Evolution of the expectation and variance during refinement of the stochastic mesh.

Since the stochastic refinement is unable to reduce $\overline{\eta}$ below $\overline{\varepsilon}$, no refinement extra refinement is needed on the deterministic computations; all refinement steps are done in the stochastic space. The new samples are added at a default target complexity of 4000, hence the mean $N_{\mathbf{x}}$ remains stable at 3993 throughout all the refinement steps. When $\overline{\eta}$ in function of $N_{\boldsymbol{\xi}}$ is fitted to the model $\beta N_{\boldsymbol{\xi}}^{\kappa}$ this yields for the convergence constant $\beta = 200$ and for the convergence rate $\kappa = -1.4$. These findings are in line with what was reported in Chapter 4.

5.5 Conclusions

In this chapter, we have proposed a coupled control of both stochastic and deterministic errors using the Riemannian metric framework useful when dealing with anisotropic behaviour. An *a priori* deterministic error control for the QoI is described, involving the computation of an adjoint state. This approach allows to drive goal-based adaptivity in the deterministic space where the remeshing effort is solely concentrated in the zones impacting the QoI. An adaptive algorithm has been proposed to chose which error dominates and to adapt in the appropriate space, the physical (deterministic) or parametric (stochastic) space, in order to reduce the total error. The proposed approach has been applied for

CFD problems modeled by the Euler system, for internal (inlet and scramjet problem) and external flows (around a NACA0012 airfoil). Preliminary results have shown that we are capable of:

- capturing the anisotropic behaviour in both deterministic and stochastic parametric space,
- detection the dominating error, and further reduction of this error by solving optimisation problems with an increased number of nodes/samples constraint,
- reducing the total error on the QoI given a computational budget,
- reducing the variance of the deterministic error contained in the samples over the stochastic space,
- reaching some numerical convergence of the statistical moments of the stochastic response.

Chapter 6

Conclusions and Perspectives

We have addressed in this thesis several open questions regarding robust adaptive uncertainty quantification. More precisely, we have considered two situations when propagating uncertainties through an approximate model: (a) no information regarding the sample (deterministic) error is available and (b) the case where estimates of this error can be computed and used to reduce the overall error.

In the first case, this model error appears as noise and outliers in the samples. This noise however does not necessarily follow a particular distribution making the filtering of this noise cumbersome in practice. In Chapter 2 an approach was proposed that quantified the confidence in each sample using an $\mathbf{L}_2-$based cross-validation of a generalized Polynomial Chaos (gPC) approximation. This information was then used as preconditioning weights in a second step where the final surrogate model was constructed by a preconditioned $\mathbf{L}_1-$regularized gPC approximation. This approach showed to be effective in automatically filtering the effect of outliers and noise not following any particular regularity assumption.

In this first part of the thesis we explored gPC formulations ranging from pseudospectral projection, Least Squares regression, and interpolation. All of these formulations used a DoE that is a tensorization of quadrature rules. For some applications this zero-variability sampling can be beneficial as it brings reliability to the recovery procedure. The downsides are that it is best suited for low to moderate dimensional problems and that it does not take the anisotropic features of the response into account.

With a focus on parametrized complex flow simulations leading to output responses with low regularity and discontinuities, an extension of metric-based adaptation to the stochastic space was proposed in Chapter 4. In this context, the choice for linear simplex elements in the parametric space is a legitimate one. It will preclude Gibbs phenomena in the approximation of the response and furthermore, the use of this meshing approach will allow for the automatic capturing of any discontinuities leading to an efficient approximation of these low regularity response surfaces. This was demonstrated on several stochastic test problems and on the fluid mechanical piston problem where the proposed method compared favourably to the existing Simplex-Stochastic Collocation and Multi-Element gPC methods.

The effects of the error in each deterministic simulation were not considered in Chapter 4 but were taken into consideration in Chapter 5. There, both the deterministic and stochastic contributions were controlled using the Riemannian metric framework. In the deterministic computations on the physical space, an *a priori* goal-oriented error estimate was used while on the stochastic space the error was controlled though an estimate of the interpolation error. This approach was demonstrated to be effective in reducing the total error in selected compressible flow test problems.

Since the original metric-based method is limited to $3D$ domains, the stochastic extension presented here will also be limited to three dimensional parametric domains. In any case, uncertainty quantification methods based on a Delaunay triangulation of the parametric space are not suited for high dimensional problems. In the context of the SSC method, it has been reported [Witteveen and Iaccarino, 2012b; Edeling, Dwight, and Cinnella, 2016] that this becomes impractical from $5-$dimensional parameter spaces onwards. Since, like the SSC method, the first step of the method proposed in Chapters 4 and 5 will be a Delaunay triangulation of the parameter space, the same limitations will apply here even if the metric-based mesh adaptation method were extended beyond 3 dimensions.

Faced with discontinuities in the stochastic response, an accurate approximation will never be cheap. One can therefore imagine that an approximation of such a response in a very high-dimensional parametric space will be computationally infeasible no matter what approximation method is used. The methods proposed in this thesis do not address the curse of dimensionality, nonetheless the low-dimensional approach can be justified by the fact that many engineering problems have a low effective dimensionality of the parameter space and the realization that accurately approximating high-dimensional discontinuous stochastic response surfaces will be too computationally expensive whatever method used.

6.1 Perspectives

With the results presented in this thesis in mind, several perspectives can be formulated.

Concerning the approach proposed in Chapter 2, potential perspectives for future work involve: – a different way of evaluating the preconditioning weights that would be even less sensitive to data outliers. With this aim, one may argue that non-weighted LASSO–type algorithms could be deployed upfront as we have noticed they often performed better in terms of robustness than standard $\mathbf{L}_2-$projections. Once combined with the second step above, this approach could then be generalized in the form of an adaptive formulation where the weights would be iteratively refined in conjunction with the surrogate model level of complexity. In this case, it would be reminiscent of an iteratively reweighted least squares technique; or – introduce a (re)weighted norm in the $\mathbf{L}_1-$minimization which is known to produce better compressive performance. This information could be provided by the spectrum of the low-order model, selected and validated in the initial step.

A downside of the linear simplex approximation used in Chapters 4 and 5 is the lack of $p-$refinement capability. A higher order interpolation error estimate could be applied for both stochastic and deterministic problems in smooth regions, allowing thus to lower the computational budget (while maintaining the desired accuracy). This would however require developments of new error estimations and associated adaptive tools.

An alternative method of achieving some form of $p-$adaptivity without a need for a higher order interpolation error estimate, would be to apply the Riemannian metric approach as presented in this manuscript, letting it determine the best sample locations. Then as a post-processing step, higher order surrogate models could be constructed using this unstructured sample set.

Furthermore, the way in which the complexity for new samples was determined in Chapter 5 could be improved. The simulations corresponding to these new samples were done on meshes with some predefined default complexity target. By looking at the target

complexities chosen for the samples neighbouring the new sample, faster convergence of the deterministic component of the total error could be achieved.
Lastly, the *curse of dimensionality* was not addressed in this theses. The UQ methods proposed in this thesis could however become an ingredient in a more encompassing approach that isolates the lower dimensional subspaces where their strengths can be exploited.

Bibliography

Alauzet, F. (2003). "Adaptation de maillage anisotrope en trois dimensions. Application aux simulations instationnaires en Mécanique des Fluides". (in French). PhD thesis. Montpellier, France: Université Montpellier II.

Alauzet, Frédéric and Adrien Loseille (2009). "Metrix user guide. error estimates and mesh control for anisotropic mesh adaptation". PhD thesis. INRIA.

— (2016a). "A decade of progress on anisotropic mesh adaptation for computational fluid dynamics". In: *Computer-Aided Design* 72, pp. 13–39.

— (2016b). "A decade of progress on anisotropic mesh adaptation for computational fluid dynamics". In: *Computer-Aided Design* 72. 23rd International Meshing Roundtable Special Issue: Advances in Mesh Generation, pp. 13 –39. ISSN: 0010-4485. DOI: `http://dx.doi.org/10.1016/j.cad.2015.09.005`. URL: `//www.sciencedirect.com/science/article/pii/S0010448515001517`.

Almeida, Regina C and J Tinsley Oden (2010). "Solution verification, goal-oriented adaptive methods for stochastic advection–diffusion problems". In: *Computer Methods in Applied Mechanics and Engineering* 199.37, pp. 2472–2486.

Anderson, J.D. (2001). *Fundamentals of Aerodynamics*. Aeronautical and Aerospace Engineering Series. McGraw-Hill. ISBN: 9780072373356.

Babuška, Ivo, Fabio Nobile, and Raul Tempone (2007). "A stochastic collocation method for elliptic partial differential equations with random input data". In: *SIAM Journal on Numerical Analysis* 45.3, pp. 1005–1034.

Becker, Stephen, Jérôme Bobin, and Emmanuel J Candès (2011). "NESTA: a fast and accurate first-order method for sparse recovery". In: *SIAM Journal on Imaging Sciences* 4.1, pp. 1–39.

Belme, A. (2011). "Aérodynamique instationnaire et méthode adjointe". (in French). PhD thesis. Sophia Antipolis, France: Université de Nice Sophia Antipolis.

Berger, M. (2003). *A Panoramic View of Riemannian Geometry*. Springer Berlin Heidelberg. ISBN: 9783642182457.

Berger, M., M. Cole, and S. Levy (1987a). *Geometry I*. Universitext. Springer Berlin Heidelberg. ISBN: 978-3-540-11658-5. DOI: `10.1007/978-3-540-93815-6`.

— (1987b). *Geometry I*. Universitext. Springer Berlin Heidelberg. ISBN: 978-3-540-17015-0. DOI: `10.1007/978-3-540-93816-3`.

Berveiller, Marc, Bruno Sudret, and Maurice Lemaire (2006). "Stochastic finite element: a non intrusive approach by regression". In: *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique* 15.1-3, pp. 81–92.

Bijl, Hester et al., eds. (2013). *Uncertainty quantification in Computational Fluid Dynamics*. Vol. 92. Lecture Notes in Computational Science and Engineering. Springer.

Blatman, Géraud and Bruno Sudret (2011). "Adaptive sparse polynomial chaos expansion based on least angle regression". In: *Journal of Computational Physics* 230.6, pp. 2345–2367.

Bottasso, Carlo L (2004). "Anisotropic mesh adaption by metric-driven optimization". In: *International Journal for Numerical Methods in Engineering* 60.3, pp. 597–639.

Breiman, Leo and Philip Spector (1992). "Submodel selection and evaluation in regression. The X-random case". In: *International statistical review/revue internationale de Statistique*, pp. 291–319.

Bryant, CM, S Prudhomme, and T Wildey (2015). "Error decomposition and adaptivity for response surface approximations from PDEs with parametric uncertainty". In: *SIAM/ASA Journal on Uncertainty Quantification* 3.1, pp. 1020–1045.

Butler, T, P Constantine, and Tim Wildey (2012). "A posteriori error analysis of parameterized linear systems using spectral methods". In: *SIAM Journal on Matrix Analysis and Applications* 33.1, pp. 195–209.

Butler, T, Clint Dawson, and Tim Wildey (2011). "A posteriori error analysis of stochastic differential equations using polynomial chaos expansions". In: *SIAM Journal on Scientific Computing* 33.3, pp. 1267–1291.

Cameron, Robert H and William T Martin (1947). "The orthogonal development of non-linear functionals in series of Fourier-Hermite functionals". In: *Annals of Mathematics*, pp. 385–392.

Candès, E. J., J. K. Romberg, and T. Tao (2004). "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information". In: *Information Theory, IEEE Transaction* 52-2, pp. 489–509.

Candès, E. J, J. K. Romberg, and T. Tao (2006). "Stable signal recovery from incomplete and inaccurate measurements". In: *Communications on pure and applied mathematics* 59.8, pp. 1207–1223.

Candès, E. J. J and T. Tao (2005). "Decoding by linear programming". In: *Information Theory, IEEE Transactions on* 51.12, pp. 4203–4215.

Chkifa, A. et al. (2015). "Discrete least squares polynomial approximation with random evaluations – application to parametric and stochastic elliptic PDEs". In: *ESAIM: M2AN* 49.3, pp. 815–837.

Choi, Seung-Kyum et al. (2004). "Polynomial Chaos expansion with Latin Hypercube Sampling for estimating response variability". In: *AIAA journal* 42.6, pp. 1191–1198.

Cohen, Albert, Mark A Davenport, and Dany Leviatan (2013). "On the stability and accuracy of least squares approximations". In: *Foundations of computational mathematics* 13.5, pp. 819–834.

Compere, Gaëtan et al. (2010). "A mesh adaptation framework for dealing with large deforming meshes". In: *International journal for numerical methods in engineering* 82.7, pp. 843–867.

Conrad, Patrick R and Youssef M Marzouk (2013). "Adaptive Smolyak pseudospectral approximations". In: *SIAM Journal on Scientific Computing* 35.6, A2643–A2670.

Coupez, Thierry (2000). "Génération de maillage et adaptation de maillage par optimisation locale". In: *Revue européenne des éléments finis* 9.4, pp. 403–423.

Cournède, Paul-Henry, Bruno Koobus, and Alain Dervieux (2006). "Positivity statements for a Mixed-Element-Volume scheme on fixed and moving grids". In: *European Journal of Computational Mechanics/Revue Européenne de Mécanique Numérique* 15.7-8, pp. 767–798.

Curran, Edward T. (2001). "Scramjet Engines: The First Forty Years". In: *Journal of Propulsion and Power* 17.6, pp. 1138–1148. DOI: `10.2514/2.5875`. URL: `http://dx.doi.org/10.2514/2.5875`.

Diaz, MJ Castro et al. (1997). "Anisotropic unstructured mesh adaptation for flows simulations". In: *Internat. J. Numer. Methods Fluids* 25, pp. 475–491.

Dobrzynski, Cécile and Pascal Frey (2008). "Anisotropic Delaunay mesh adaptation for unsteady simulations". In: *Proceedings of the 17th international Meshing Roundtable*, pp. 177–194.

Donoho, David L (2006). "Compressed sensing". In: *Information Theory, IEEE Transactions on* 52.4, pp. 1289–1306.

Donoho, David L, Michael Elad, and Vladimir N Temlyakov (2006). "Stable recovery of sparse overcomplete representations in the presence of noise". In: *Information Theory, IEEE Transactions on* 52.1, pp. 6–18.

Doostan, Alireza and Houman Owhadi (2011). "A non-adapted sparse approximation of PDEs with stochastic inputs". In: *Journal of Computational Physics* 230.8, pp. 3015–3034.

Edeling, WN, Richard P Dwight, and Pasquale Cinnella (2016). "Simplex-stochastic collocation method with improved scalability". In: *Journal of Computational Physics* 310, pp. 301–328.

Efron, Bradley et al. (2004). "Least angle regression". In: *The Annals of statistics* 32.2, pp. 407–499.

Fidkowski, K.J. and D.L. Darmofal (2011). "Review of output-based error estimation and mesh adaptation in computational fluid dynamics". In: *AIAA journal* 49.4, pp. 673–694.

Foo, Jasmine and George Em Karniadakis (2010). "Multi-element probabilistic collocation method in high dimensions". In: *Journal of Computational Physics* 229.5, pp. 1536–1557.

Foo, Jasmine, Xiaoliang Wan, and George Em Karniadakis (2008). "The multi-element probabilistic collocation method (ME-PCM): Error analysis and applications". In: *Journal of Computational Physics* 227.22, pp. 9572–9595.

Frey, Pascal (2001). "Yams a fully automatic adaptive isotropic surface remeshing procedure". PhD thesis. INRIA.

Frey, Pascal Jean and Paul-Louis George (2007). *Mesh generation: application to finite elements*. ISTE.

Fuchs, Jean Jacques (2005). "Recovery of exact sparse representations in the presence of bounded noise". In: *Information Theory, IEEE Transactions on* 51.10, pp. 3601–3608.

George, PL (2002). *Gamanic3d, Adaptive anisotropic tetrahedral mesh generator*. Tech. rep. Technical Report, INRIA.

Ghanem, Roger G and Pol D Spanos (2003). *Stochastic finite elements: a spectral approach*. Courier Corporation.

Green, P. J. (1984). "Iteratively Reweighted Least Squares for Maximum Likelihood Estimation, and some Robust and Resistant Alternatives". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 46.2, pp. 149–192.

Hampton, Jerrad and Alireza Doostan (2015a). "Coherence motivated sampling and convergence analysis of least squares Polynomial Chaos regression". In: *Computer Methods in Applied Mechanics and Engineering* 290, pp. 73–97.

— (2015b). "Compressive sampling of Polynomial Chaos expansions: convergence analysis and sampling strategies". In: *Journal of Computational Physics* 280, pp. 363–386.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The elements of statistical learning*. 2nd. Springer series in statistics. Springer.

Hecht, F and B Mohammadi (1997). "Mesh adaption by metric control for multi-scale phenomena and turbulence". In: *35th Aerospace Sciences Meeting and Exhibit*, p. 859.

Hecht, Frédéric. "BAMG: Bidimensional Anisotropic Mesh Generator". In:

Huber, P. J. and E. M. Ronchetti (2009). *Robust statistics*. 2nd. Probability and statistics. Wiley.

Jakeman, John D, Michael S Eldred, and Khachik Sargsyan (2015). "Enhancing ℓ_1-minimization estimates of polynomial chaos expansions using basis selection". In: *Journal of Computational Physics* 289, pp. 18–34.

Jakeman, John D, Akil Narayan, and Dongbin Xiu (2013). "Minimal multi-element stochastic collocation for uncertainty quantification of discontinuous functions". In: *Journal of Computational Physics* 242, pp. 790–808.

James O. Street, Raymond J. Carroll and David Ruppert (1988). “A Note on Computing Robust Regression Estimates Via Iteratively Reweighted Least Squares”. In: *The American Statistician* 42.2, pp. 152–154.

Jones, William, Eric Nielsen, and Michael Park (2006). “Validation of 3D adjoint based error estimation and mesh adaptation for sonic boom prediction”. In: *44th AIAA Aerospace Sciences Meeting and Exhibit*, p. 1150.

Laug, Patrick and Houman Borouchaki (2003). “BL2D-V2: mailleur bidimensionnel adaptatif”. PhD thesis. INRIA.

Le Maître, Olivier P and Omar M Knio (2010). *Spectral Methods for Uncertainty Quantification*. Springer.

Le Maıtre, OP et al. (2004a). “Multi-resolution analysis of Wiener-type uncertainty propagation schemes”. In: *Journal of Computational Physics* 197.2, pp. 502–531.

Le Maıtre, OP et al. (2004b). “Uncertainty propagation using Wiener–Haar expansions”. In: *Journal of computational Physics* 197.1, pp. 28–57.

Li, Xiangrong, Mark S Shephard, and Mark W Beall (2005). “3D anisotropic mesh adaptation by mesh modification”. In: *Computer methods in applied mechanics and engineering* 194.48, pp. 4915–4950.

Lin, G., C.-H. Su, and G. E. Karniadakis (2004). “The Stochastic Piston Problem”. In: *PNAS* 101.45, pp. 15840–15845.

Loseille, Adrien (2008). “Adaptation de maillage anisotrope 3D multi-échelles et ciblée à une fonctionnelle pour la mécanique des fluides. Application à la prédiction haute-fidélité du bang sonique”. (in French). PhD thesis. Paris, France: INRIA.

Loseille, Adrien and Frédéric Alauzet (2011a). “Continuous mesh framework part I: well-posed continuous interpolation error”. In: *SIAM Journal on Numerical Analysis* 49.1, pp. 38–60.

— (2011b). “Continuous mesh framework part II: validations and applications”. In: *SIAM Journal on Numerical Analysis* 49.1, pp. 61–86.

Loseille, Adrien, Alain Dervieux, and Frédéric Alauzet (2010). “Fully anisotropic goal-oriented mesh adaptation for 3D steady Euler equations”. In: *Journal of computational physics* 229.8, pp. 2866–2897.

Loseille, Adrien and Rainald Löhner (2010). “Anisotropic adaptive simulations in aerodynamics”. In: *AIAA Paper* 169, pp. 6–2010.

Mathelin, L. and K. A. Gallivan (2012). “A compressed sensing approach for partial differential equations with random input data”. In: *Communications in Computational Physics* 12.4, pp. 919–954.

Mathelin, Lionel and Olivier Le Maître (2007). “Dual-based a posteriori error estimate for stochastic finite element methods”. In: *Communications in Applied Mathematics and Computational Science* 2.1, pp. 83–115.

Michal, Todd and Joshua Krakos (2012). “Anisotropic mesh adaptation through edge primitive operations”. In: *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, p. 159.

Migliorati, Giovanni, Fabio Nobile, and Raúl Tempone (2015). “Convergence estimates in probability and in expectation for discrete least squares with noisy evaluations at random points”. In: *Journal of Multivariate Analysis* 142, pp. 167 –182.

Migliorati, Giovanni et al. (2014). “Analysis of Discrete ℓ_2 Projection on Polynomial Spaces with Random Evaluations”. In: *Foundations of Computational Mathematics* 14.3, pp. 419–456.

Molinaro, Annette M, Richard Simon, and Ruth M Pfeiffer (2005). “Prediction error estimation: a comparison of resampling methods”. In: *Bioinformatics* 21.15, pp. 3301–3307.

Needell, Deanna and Joel A Tropp (2009). “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples”. In: *Applied and Computational Harmonic Analysis* 26.3, pp. 301–321.

Pain, CC et al. (2001). “Tetrahedral mesh optimisation and adaptivity for steady-state and transient finite element calculations”. In: *Computer Methods in Applied Mechanics and Engineering* 190.29, pp. 3771–3796.

Palacios, Francisco et al. (2012). “Robust grid adaptation for efficient uncertainty quantification”. In: *AIAA journal* 50.7, pp. 1538–1546.

Peng, Ji, Jerrad Hampton, and Alireza Doostan (2014). “A weighted ℓ_1-minimization approach for sparse polynomial chaos expansions”. In: *Journal of Computational Physics* 267, pp. 92–111.

Peter, J., M. Nguyen-Dinh, and P. Trontin (2012). “Goal oriented mesh adaptation using total derivative of aerodynamic functions with respect to mesh coordinates – With applications to Euler flows”. In: *Computers & Fluids* 66, pp. 194 –214.

Poëtte, G., A. Birolleau, and D. Lucor (2015). “Iterative Polynomial Approximation Adapting to Arbitrary Probability Distribution”. In: *SIAM J. Numerical Analysis* 53.3, pp. 1559–1584.

Rauhut, Holger and Rachel Ward (2012). “Sparse Legendre expansions via ℓ_1-minimization”. In: *Journal of approximation theory* 164.5, pp. 517–533.

Resmini, A., J. Peter, and D. Lucor (2015). “Sparse grids-based stochastic approximations with applications to aerodynamics sensitivity analysis”. In: *Int. J. Numer. Meth. Engng* 10.1002/nme.5005.

Sargsyan, Khachik et al. (2014). “Dimensionality reduction for complex models via Bayesian compressive sensing”. In: *International Journal for Uncertainty Quantification* 4.1.

Shen, J. and L.-L. Wang (2010). “Sparse Spectral Approximations of High-Dimensional Problems Based on Hyperbolic Cross”. In: *SIAM Journal on Numerical Analysis* 48.3, pp. 1087–1109.

Smola, Alex J and Bernhard Schölkopf (2004). “A tutorial on support vector regression”. In: *Statistics and computing* 14.3, pp. 199–222.

Tam, A et al. (2000). “Anisotropic mesh adaptation for 3D flows on structured and unstructured grids”. In: *Computer Methods in Applied Mechanics and Engineering* 189.4, pp. 1205–1230.

Tang, Gary and Gianluca Iaccarino (2014). “Subsampled Gauss quadrature nodes for estimating Polynomial Chaos expansions”. In: *SIAM/ASA Journal on Uncertainty Quantification* 2.1, pp. 423–443.

Tang, Tao and Tao Zhou (2014). “On Discrete Least-Squares Projection in Unbounded Domain with Random Evaluations and its Application to Parametric Uncertainty Quantification”. In: *SIAM Journal on Scientific Computing* 36.5, A2272–A2295.

Tatang, M. et al. (1997). “An efficient method for parametric uncertainty analysis of numerical geophysical models”. In: *Journal of Geophysical Research* 102, pp. 21925–21932.

Tibshirani, R. (1996). “Regression shrinkage and selection via the lasso”. In: *Journal of the Royal Statistical Society* 58-1, pp. 267–288.

Tropp, Joel (2006). “Just relax: Convex programming methods for identifying sparse signals in noise”. In: *Information Theory, IEEE Transactions on* 52.3, pp. 1030–1051.

Vallet, Marie-Gabrielle (1992). “Génération de maillages éléments finis anisotropes et adaptatifs”. PhD thesis.

Van Den Berg, Ewout and Michael P Friedlander (2008). “Probing the Pareto frontier for basis pursuit solutions”. In: *SIAM Journal on Scientific Computing* 31.2, pp. 890–912.

Venditti, D.A. and D.L. Darmofal (2003). “Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows”. In: *J. Comp. Phys.* 187.1, pp. 22–46.

Wagner, JL et al. (2007). "An experimental investigation of supersonic inlet unstart". In: *AIAA Paper* 4352, p. 2007.

Wagner, JL et al. (2009). "Experimental investigation of unstart in an inlet/isolator model in Mach 5 flow". In: *AIAA journal* 47.6, pp. 1528–1542.

Wan, Xiaoliang and George Em Karniadakis (2005). "An adaptive multi-element generalized polynomial chaos method for stochastic differential equations". In: *Journal of Computational Physics* 209.2, pp. 617–642.

— (2006). "Multi-element generalized polynomial chaos for arbitrary probability measures". In: *SIAM Journal on Scientific Computing* 28.3, pp. 901–928.

Ward, Rachel (2009). "Compressed sensing with cross-validation". In: *Information Theory, IEEE Transactions on* 55.12, pp. 5773–5782.

Wiener, Norbert (1938). "The Homogeneous Chaos". In: *American Journal of Mathematics*, pp. 897–936.

Williams, Christopher KI and Carl Edward Rasmussen (2006). "Gaussian processes for machine learning". In: *the MIT Press* 2.3, p. 4.

Witteveen, Jeroen, Karthik Duraisamy, and Gianluca Iaccarino (2011). "Uncertainty quantification and error estimation in scramjet simulation". In: *17th AIAA International Space Planes and Hypersonic Systems and Technologies Conference*. American Institute of Aeronautics and Astronautics. DOI: `doi:10.2514/6.2011-2283`. URL: `http://dx.doi.org/10.2514/6.2011-2283`.

Witteveen, Jeroen AS and Gianluca Iaccarino (2010). "Simplex elements stochastic collocation in higher-dimensional probability spaces". In: *12th AIAA Non-Deterministic Approaches Conference, Orlando, Florida, AIAA*. Vol. 2924, pp. 29–30.

— (2012a). "Refinement criteria for simplex stochastic collocation with local extremum diminishing robustness". In: *SIAM Journal on Scientific Computing* 34.3, A1522–A1543.

— (2012b). "Simplex stochastic collocation with random sampling and extrapolation for nonhypercube probability spaces". In: *SIAM Journal on Scientific Computing* 34.2, A814–A838.

— (2013a). "Simplex stochastic collocation with ENO-type stencil selection for robust uncertainty quantification". In: *Journal of Computational Physics* 239, pp. 1–21.

— (2013b). "Subcell resolution in simplex stochastic collocation for spatial discontinuities". In: *Journal of Computational Physics* 251, pp. 17–52.

Witteveen, Jeroen AS, Alex Loeven, and Hester Bijl (2009). "An adaptive stochastic finite elements approach based on Newton–Cotes quadrature in simplex elements". In: *Computers & Fluids* 38.6, pp. 1270–1288.

Xiu, Dongbin and Jan S. Hesthaven (2005a). "High-order collocation methods for differential equations with random inputs". In: *J. Sci. Comput.* 27.3, pp. 1118–1139.

Xiu, Dongbin and Jan S Hesthaven (2005b). "High-order collocation methods for differential equations with random inputs". In: *SIAM Journal on Scientific Computing* 27.3, pp. 1118–1139.

Xiu, Dongbin and George Em Karniadakis (2002). "The Wiener–Askey polynomial chaos for stochastic differential equations". In: *SIAM journal on scientific computing* 24.2, pp. 619–644.

Yan, Liang, Ling Guo, and Dongbin Xiu (2012). "Stochastic collocation algorithms using ℓ_1-minimization". In: *International Journal for Uncertainty Quantification* 2.3.

Yang, Xiu and George Em Karniadakis (2013). "Reweighted ℓ_1 minimization method for stochastic elliptic differential equations". In: *Journal of Computational Physics* 248, pp. 87–108.

Zhang, Z. et al. (2013). "Numerical solution of the Stratonovich- and Ito–Euler equations: Application to the stochastic piston problem". In: *Journal of Computational Physics* 236, pp. 15 –27. ISSN: 0021-9991. DOI: `http://dx.doi.org/10.1016/j.jcp.2012.`

11.017. URL: `http://www.sciencedirect.com/science/article/pii/S002199911200695X`.

Zhou, T., A. Narayan, and D. Xiu (2015). "Weighted discrete least-squares polynomial approximation using randomized quadratures". In: *J. Comp. Phys.* 298, pp. 787–800.

Zhou, Tianhe and Danfu Han (2008). "A weighted least squares method for scattered data fitting". In: *Journal of Computational and Applied Mathematics* 217.1, pp. 56–63.

Zienkiewicz, Olgierd Cecil and Jian Zhong Zhu (1992a). "The superconvergent patch recovery and a posteriori error estimates. Part 1: The recovery technique". In: *International Journal for Numerical Methods in Engineering* 33.7, pp. 1331–1364.

— (1992b). "The superconvergent patch recovery and a posteriori error estimates. Part 2: Error estimates and adaptivity". In: *International Journal for Numerical Methods in Engineering* 33.7, pp. 1365–1382.