On the use of the Sparse Grid techniques coupled with Polynomial Chaos

Pietro Marco Congedo, Remi Abgrall, Gianluca Geraci

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

On the use of the Sparse Grid techniques coupled with Polynomial Chaos

Remi Abgrall — Pietro Marco Congedo — Gianluca Geraci

N° 7579

March 23, 2011

Computational models and simulation

ISSN 0249-6399 ISRN INRIA/RR--7579--FR+ENG

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE

INRIA

centre de recherche
BORDEAUX - SUD OUEST

On the use of the Sparse Grid techniques coupled with Polynomial Chaos

Remi Abgrall , Pietro Marco Congedo , Gianluca Geraci

Theme : Computational models and simulation
Applied Mathematics, Computation and Simulation
Équipe-Projet Bacchus

Rapport de recherche n° 7579 — March 23, 2011 — 55 pages

Abstract: In this work we want to explore potentialities and deficiencies of Sparse Grid techniques coupled with Polynomial Chaos for multi dimensional (up to fifteen) stochastic problems. We used the sparse grid technique to compute the multi dimensional integrals needed to evaluate the coefficients of the polynomial expansion. Aim of this work is to compare several Sparse Grid techniques in terms of computational cost and accuracy with respect to Monte Carlo reference solution. We considered two problems: an algebraic function widely used in literature to test stochastic numerical methods, namely g-function, with poor regularity properties and a stochastic numerical simulation of a monodimensional compressible nozzle, where geometry and operating conditions are functions of random variables. After a detailed study on error computations and on the influence of the probability density function, we investigated the possibility of reducing the number of random variables by means of ANOVA analysis.

Key-words: Polynomial Chaos, Sparse Grid, Uncertainty Quantification

Centre de recherche INRIA Bordeaux – Sud Ouest
Domaine Universitaire - 351, cours de la Libération 33405 Talence Cedex
Téléphone : +33 5 40 00 69 00

Couplage de techniques de grilles creuses avec la méthode du Chaos Polynomial

Résumé : Dans ce rapport, les potentialités et les problèmes des techniques de grilles creuses couplées avec la méthode de Chaos Polynomial ont été analysées pour des problèmes stochastiques multidimensionnelles (jusqu'à 15 paramètres). Les techniques des grilles creuses ont été utilisées pour calculer les intégrales multidimensionnelles, nécessaires au calcul des coefficients du développement polynomiale. Le but de ces travaux est de comparer des différentes techniques de grilles creuses en terme de coût de calcul et précision par rapport à la solution de référence calculée avec la méthode Monte Carlo. Nous avons considéré deux problèmes : i) une fonction algébrique très utilisée dans la littérature pour tester les méthodes numériques stochastiques, appelé g-fonction, qui se caractérise par des faibles propriétés de régularité, et ii) la simulation numérique stochastique d'une tuyère unidimensionnelle compressible, où la géométrie et les conditions opérationnelles sont des fonctions des variables aléatoires. Après avoir calculé l'erreur et l'influence des fonctions de densité de probabilité, on a prise en compte la possibilité de réduire le nombre de variables aléatoires à travers une analyse ANOVA.

Mots-clés : Chaos Polynomial, grilles creuses, quantification de l'incertitude

Contents

List of Figures

List of Tables

1 Introduction

The objective of numerical simulations is to predict physical events and provide a better understanding of engineering devices. In the last decades strong effort was devoted to the development of accurate numerical algorithms on parallel architectures. An open question concerns how to quantify the confidence level of numerical simulations by considering that the physical system is affected by several sources of uncertainty. To establish the quality of a numerical simulation Verification & Validation have been introduced [1]. Verification aims to quantify the errors associated to the numerical resolution of the given system of equations, while Validation aims to identify whether system of equation is a correct representation of physics. Uncertainties in Computational Fluid Dynamics are due to an incomplete knowledge of physics (epistemic) or measurement (aleatory) errors, so incorporating uncertainties in simulation is mandatory in order to have a predictive numerical simulation.

1.1 UQ - state of the art

UQ formulations can be: intrusive or non-intrusive. They are intrinsically different because in the intrusive approach the numerical method must be modified to account for uncertainties; on the contrary in the non-intrusive framework the solver code is used as a black box, only the responses in a determinate set of points (in the stochastical space) are required. In this work only the non-intrusive strategy was explored.

A short overview on the state of the art for the uncertainty quantification techniques is now presented. The most common used method is the Monte Carlo (MC) sampling. In this method independent realizations of random inputs based on their probability distribution are generated. Upon solving the deterministic realizations of the problem, one collects an ensemble of solutions. From this ensemble of deterministic solutions it is possible to compute all the statistical information. The MC is very straightforward to apply and it requires only realizations of deterministic code. The convergence is relatively slow ($\approx 1/\sqrt{N}$ where N is the number of realizations) and a high number of solutions is required. Techniques have been developed to accelerate the convergence of the MC, eg. Latin hypercube (LHS), quasi-MC (QMC), however additional restrictions are posed and their applicability is often limited. In this work the Monte Carlo method used is based on a quasi-MC strategy. Recentely a method called generalized Polynomial Chaos (gPC) has been developed as a generalization of the classical polynomial chaos theory [2]. In this method the stochastic solution is expressed as orthogonal polynomials of the input random parameters and different types of polynomials can be chosen to achieve the better convergence. It is a spectral representation in a random space and it exhibits a fast convergence when the solution depends smoothly on the random parameters. In this work we focused on the gPC method so a complete description of the method is furnished in the next section. Another strategy is the so-called Stochastic Collocation method. This method is non-intrusive and, differently from gPC, uses a interpolation Lagrangian basis instead of the polynomial basis (see [3] and [4] for more details). A variant of this method, called Probabilistic Collocation [5], that involves a chaos version of Lagrangian interpolation has been proposed in the last years. More recently an effort in analyzing problems with

a higher number of uncertainties led to adaptive methodologies to tackle in an efficient manner the high dimensionality. We can quote Blatman and Sudret [6], Witteveen et al. [7], the multi-element method of Foo and Karniadakis [8], Ma and Zabaras [9] and Agarwall [10].

In this work all the results presented were obtained by the NISP library developed in the OPUS project (see also [11] for some details).

The report is organized as follows. In section §2 the mathematical background of the problem is furnished. In section §3 the Polynomial Chaos is described with an emphasis on the integration problem §3.3 and §3.4. The ANOVA technique is briefly recalled in §4. Numerical results are furnished in section §5 for the g-function and in section §6 for the nozzle problem. Some results on the possibility of a random variables reduction are presented in §7. Conclusions and future perspective work are discussed in section §8.

Part I
Theoretical framework

2 Problem setting

Consider the following problem for an output of interest $u(\boldsymbol{x}, t, \boldsymbol{\xi}(\boldsymbol{\omega}))$:

$$\mathcal{L}(\boldsymbol{x}, t, \boldsymbol{\xi}(\boldsymbol{\omega}); u(\boldsymbol{x}, t, \boldsymbol{\xi}(\boldsymbol{\omega}))) = \mathcal{S}(\boldsymbol{x}, t, \boldsymbol{\xi}(\boldsymbol{\omega})), \tag{1}$$

where the operator $\mathcal{L}$ can be either an algebraic or a differential operator (in this case we need appropriate initial and boundary conditions). The operator $\mathcal{L}$ and the source term $\mathcal{S}$ are defined on the domain $D \times T \times \Xi$, where $\boldsymbol{x} \in D \subset \mathbb{R}^{n_d}$, with $n_d \in \{1, 2, 3\}$, and $t \in T$ are the spatial and temporal dimensions. Randomness is introduced in (1) and its initial and boundary conditions in term of d second order random parameters $\boldsymbol{\xi}(\boldsymbol{\omega}) = \{\xi_1(\omega_1), \ldots, \xi_d(\omega_d)\} \in \Xi$ with parameter space $\Xi \subset \mathbb{R}^d$. The symbol $\boldsymbol{\omega} = \{\omega_1, \ldots, \omega_d\} \in \Omega \subset \mathbb{R}$ denotes realizations in a complete probability space $(\Omega, \mathcal{F}, P)$. Here Ω is the set of outcomes, $\mathcal{F} \subset 2^\Omega$ is the σ-algebra of events and $P : \mathcal{F} \to [0, 1]$ is a probability measure. In our case the random variables $\boldsymbol{\omega}$ are by definition standard uniformly $\mathcal{U}(0, 1)$ or Gaussian $\mathcal{N}(\mu = 0, \sigma = 1)$ distributed. Random parameters $\boldsymbol{\xi}(\boldsymbol{\omega})$ can have any arbitrary probability density function $p(\boldsymbol{\xi}(\boldsymbol{\omega}))$, in this way $p(\boldsymbol{\xi}(\boldsymbol{\omega})) > 0$ for all $\boldsymbol{\xi}(\boldsymbol{\omega}) \in \Xi$ and $p(\boldsymbol{\xi}(\boldsymbol{\omega})) = 0$ for all $\boldsymbol{\xi}(\boldsymbol{\omega}) \notin \Xi$; we can now drop the argument $\boldsymbol{\omega}$ for brevity. The probability density function $p(\boldsymbol{\xi}(\boldsymbol{\omega}))$ is defined as a joint probability density function from the independent probability function of each variable: $p(\boldsymbol{\xi}(\boldsymbol{\omega})) = \prod_{i=1}^{d} p_i(\xi_i)$. This assumption allows to an independent polynomial representation for every direction in the probabilistic space with the possibility to recover the multidimensional representation by tensorization. In the present work the test cases, presented in part II, are algebraic, steady equations and, if we drop the spatial argument $\boldsymbol{x}$ for simplicity, we can write

$$\mathcal{L}(\boldsymbol{\xi}; u(\boldsymbol{\xi})) = 0 \tag{2}$$

then the aim is to find the statistical moments of the solution $u(\boldsymbol{\xi})$.

3 PC expansion technique

In this section the Polynomial Chaos technique is presented in the framework of a non intrusive approach. This technique consists in several steps. First a sampling method is chosen to generate a discrete parameter space $\boldsymbol{\xi}_i \in \bar{\Xi} \subset \Xi$ with $i = 1, \ldots, N$ in which the model equation (2) is evaluated by a deterministic code determining a set of solution $u_i = u(\boldsymbol{\xi}_i)$. Finally it is necessary to reconstruct the variable $u(\boldsymbol{\xi})$ as a polynomial expansion (see later) in which the coefficients are computed evaluating d-dimensional integrals with an opportune quadrature techniques in which the u_i values are needed.

3.1 Generalized Polynomial Chaos

The original work of Wiener ([12]), in which the homogeneous chaos theory was developed using only Hermite polynomials, has been extended by Askey and Wilson [2]. In this work they obtained the correct set of polynomials which provide an optimal basis for different (continuous) probability distribution types. The optimal basis selection derives from the orthogonality of these polynomials, with respect to weighting functions that correspond to the probability density functions in standard form (the density functions and weighting functions differs for a normalization factor).

In the framework of the Generalized Polynomial Chaos we can employ the orthogonal basis reported in the Askey scheme to approximate the functional form between each random inputs and the stochastic response. The chaos expansion reads

$$\begin{aligned} u(\boldsymbol{\xi}) = a_0 B_0 + \sum_{i_1=1}^{\infty} a_{i_1} B_1(\xi_{i_1}) + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} a_{i_1 i_2} B_2(\xi_{i_1}, \xi_{i_2}) \\ + \sum_{i_1=1}^{\infty} \sum_{i_2=1}^{i_1} \sum_{i_3=1}^{i_2} a_{i_1 i_2 i_3} B_3(\xi_{i_1}, \xi_{i_2}, \xi_{i_3}) + \cdots \end{aligned} \tag{3}$$

where the unbounded expansion is employed and, with every additional summation, an additional polynomial order is introduced. Substituting an order-based index in a term-based ones, the equation (3) can be simplified as

$$u(\boldsymbol{\xi}) = \sum_{k=0}^{\infty} \beta_k \Psi_k(\boldsymbol{\xi}) \tag{4}$$

where we have a one-to-one correspondence between the couples $a_i - \beta_k$ and $B_n - \Psi_k$. Here we assume that Ψ_i form an Hilbert basis of $L_2(\boldsymbol{\xi}, p(\boldsymbol{\xi}))$ the space of the second-order random variable spanned by $\boldsymbol{\xi}$ and $u(\boldsymbol{\xi}) \in L_2(\boldsymbol{\xi}, p(\boldsymbol{\xi}))$, i.e. it is a second order random field

$$||u(\boldsymbol{\xi})||^2 = \int_{\Omega^d} u(\boldsymbol{\xi})^2 p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} < \infty.$$

The knowledge of the β_k allows to fully characterize the output random variable. Each polynomial $\Psi(\boldsymbol{\xi})$ of total degree n_o is a multivariate polynomial

which involve tensorization of one dimensional ones by a multi index m_i:

$$\Psi(\boldsymbol{\xi}) = \prod_{i=1}^{n_o} \psi_{m_i}(\xi_i). \tag{5}$$

An example of multivariate polynomials of degrees up to second, in the bidimensional space, is reported in the Appendix A.

3.2 Expansion truncation

When we deal with a practical problem we have to truncate the infinite expansion (4), so we can write

$$u(\boldsymbol{\xi}) = \tilde{u}(\boldsymbol{\xi}) + \mathcal{O}_T = \sum_{k=0}^{P} \beta_k \Psi_k(\boldsymbol{\xi}) + \mathcal{O}_T, \tag{6}$$

where $\mathcal{O}_T$ is a truncation error. If the polynomial chaos expansion includes a complete basis of polynomials up to the fixed total order n_o and we deal with a d-dimensional space, the number of terms N_{tot} in the expansion (6) is

$$N_{tot} = P + 1 = 1 + \sum_{s=1}^{p} \frac{1}{s!} \prod_{r=0}^{s-1} (d + r) = \frac{(n_o + d)!}{n_o!\, d!}.$$

In a non-intrusive framework is possible to compute the β_k coefficients by two methods: least-square approximation and spectral projection here employed. The main advantage of this method on the other one is that it exploits the orthogonality of the PC basis. Taking the inner product of the output PC expansion (6) with Ψ_k, making use of the orthogonality of the basis, it comes

$$\beta_k = \frac{\langle u(\boldsymbol{\xi}), \Psi_k(\boldsymbol{\xi}) \rangle}{\langle \Psi_k(\boldsymbol{\xi}), \Psi_k(\boldsymbol{\xi}) \rangle}, \quad \forall k. \tag{7}$$

Recalling the definition of the inner product,

$$\langle f(\boldsymbol{\xi}) g(\boldsymbol{\xi}) \rangle = \int_{\Omega^d} f(\boldsymbol{\xi}) g(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi}, \tag{8}$$

the determination of the PC coefficients of the output expansion reduces to the evaluation of N_{tot} d-dimensional integrals

$$I_k = \beta_k \langle \Psi_k \Psi_k \rangle = \int_{\Omega^d} u(\boldsymbol{\xi}) \Psi_k(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi}, \tag{9}$$

that it is all but an easy task to do in an efficient manner, see §3.3 to see how to tackle it. By the way, thanks to the tensorization character of the Ψ_k, the exact evaluation of $\langle \Psi_k, \Psi_k \rangle$ is immediate:

$$\langle \Psi_k, \Psi_k \rangle = \langle \Psi_k^2 \rangle = \prod_{i=1}^{d} \langle \psi_{m_i^k}^2 \rangle.$$

and, at this point, the stochastic solution $u(\boldsymbol{\xi})$ is reconstructed as $\tilde{u}(\boldsymbol{\xi})$. We can now compute the statistical moments of interest: the expected value $\mu(u)$

$$\begin{aligned}\mu(u) \approx \mu(\tilde{u}) &= \int_{\Omega^d} \tilde{u}(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = \int_{\Omega^d} \sum_{k=0}^{P} \beta_k \Psi_k(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} \\ &= \beta_0 \int_{\Omega^d} p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} + \int_{\Omega^d} \sum_{k=1}^{P} \beta_k \Psi_k(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = \beta_0\end{aligned}$$

and the variance $\sigma^2(u)$

$$\begin{aligned}\sigma^2(u) \approx \sigma^2(\tilde{u}) &= \int_{\Omega^d} (\tilde{u}(\boldsymbol{\xi}) - \mu(\tilde{u}))^2 p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = \mu(\tilde{u}^2) - (\mu(\tilde{u}))^2 \\ &= \int_{\Omega^d} \tilde{u}^2 p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} - \beta_0^2 = \int_{\Omega^d} \left(\sum_{k=0}^{P} \beta_k \Psi_k(\boldsymbol{\xi}) \right)^2 p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} - \beta_0^2 \\ &= \int_{\Omega^d} \left(\sum_{k=1}^{P} \beta_k \Psi_k(\boldsymbol{\xi}) \right)^2 p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = \sum_{k=1}^{P} \beta_k^2 \langle \Psi_k^2(\boldsymbol{\xi}) \rangle.\end{aligned}$$

See the Appendix B for some details on the identities employed to write the mean and the variance.

3.3 Numerical multi-dimensional integration

Multi dimensional integration, or quadrature, is a classical numerical problem, that can be solved by means of several methods. Classically we can approximate an integral by the use of summation of a finite number of terms:

$$I_k = \sum_{i=1}^{N} u(\boldsymbol{\xi}_i) \Psi_k(\boldsymbol{\xi}_i) p(\boldsymbol{\xi}_i) \tilde{w}_i + \mathcal{O}_I = \sum_{i=1}^{N} u_i \Psi_k(\boldsymbol{\xi}_i) w_i + \mathcal{O}_I, \tag{10}$$

in which the evaluation of the integrand function in a finite set of points $(\boldsymbol{\xi}_i)$ in the d-dimensional space, multiplied by weights (w_i) should be realized. These values depend on the method chosen for the quadrature $(\tilde{w}_i)$ and, of course, on the joint probability distribution $p(\boldsymbol{\xi})$ associated at the stochastic variables. $\mathcal{O}_I$ represents an integration error.

In this work we are dealing with three different techniques for the numerical quadrature:

- Monte Carlo sampling - it generates a random distribution $\boldsymbol{\xi}_i$ of N points in the d-dimensional space. The weight $\tilde{w}_i$ associated to every point $\boldsymbol{\xi}_i$ is simply $1/N$. This method is robust and converges for every function in L_2 with a convergent rate independent of the dimension d, but the convergence rate $1/\sqrt{N}$ is very slow;

- Full tensorization - starting from a monodimensional quadrature technique in which a sequence of nodes ξ_j^r and weights w_j^r where $r = 1, \ldots, d$, $j = 1, \ldots, N_r$ and N_r is the number of nodes in the rth direction is generated.

These sequences of points are then tensorized to fill the d-dimensional space: $N = \prod_{r=1}^{d} N_r$, $\boldsymbol{\xi}_i = \xi^1_{h_{i1}} \otimes \cdots \otimes \xi^d_{h_{id}}$ and $\tilde{w}_i = w^1_{h_{i1}} \otimes \cdots \otimes w^d_{h_{id}}$ where we employ a matrix* $H = \{h_{ij}\} \in \mathbb{R}^{N \times d}$. This method becomes infeasible for high dimensional problems;

- Smolyak's algorithm [13, 14] has been proposed to reduce the numerical cost of the full tensorization. It is presented in the following section.

3.4 Smolyak's algorithm

Smolyak's cubature technique is based on partial tensorization of one-dimensional quadrature formulas [13]. We consider a sequence of one-dimensional quadrature formulas involving a number of points n_l corresponding to a level l. The number of points n_l grows with l. We define a classical quadrature formula in the one-dimensional case

$$\int_{\Omega} f(\xi)p(\xi)d\xi \approx Q^1_l f \stackrel{\text{def}}{=} \sum_{i=1}^{n_l} w^l_i f(\xi^l_i). \tag{11}$$

Setting $Q^l_0 f = 0$, we define also a difference quadrature formula as

$$\Delta^1_{k\geq 1} f \stackrel{\text{def}}{=} (Q^1_k - Q^1_{k-1})f \quad \text{with} \quad Q^1_0 f = 0. \tag{12}$$

Using a multi-index $\mathbf{k} = (k_1, \ldots, k_d)$, the d-dimensional difference formula is obtained by a tensor product of one-dimensional difference quadrature formulas

$$\Delta_{\mathbf{k}} f \stackrel{\text{def}}{=} (\Delta^1_{k_1} \otimes \cdots \otimes \Delta^1_{k_d})f. \tag{13}$$

Now we have all the components to assemble the d-dimensional Smolyak's quadrature formula of a level l:

$$\int_{\Omega_d} f(\boldsymbol{\xi})p(\boldsymbol{\xi})\mathrm{d}\boldsymbol{\xi} \approx Q^d_l f \stackrel{\text{def}}{=} \sum_{|\mathbf{k}|\leq l+d-1} \Delta_{\mathbf{k}} f, \tag{14}$$

where $l \in \mathbb{N}$, $\mathbf{k} \in \mathbb{N}^d$ and $|\mathbf{k}| \stackrel{\text{def}}{=} \sum_{i=1}^{d} k_i$.

In the Appendix C we reported an example with all the required operations to compute the sparse grid (nodes and weights) in the bidimensional ($d = 2$) case with levels up to second for a Clenshaw-Curtis univariate quadrature formula.

All the quadrature rules used in this work are sequences of roots of polynomials. We used the following kind of polynomials: Legendre and Hermite for the full tensorization, respectively for uniform or Gaussian distributed uncertainties; Clenshaw-Curtis, Fejer and Legendre for the Smolyak algorithm. The Clenshaw-Curtis rule was used also for the simple collocation and for the Petras algorithm. The Petras algorithm is based on the Smolyak ones, but it allows a more efficient use of the computational resources [15]. The Clenshaw-Curtis and Fejer (second kind) are nested very similar rules (they were proposed by Fejer in the 1933 before the work of Clenshaw and Curtis in the 1960) and are based on the extrema points of the first kind polynomials of Chebyshev. The

*We shall say, more properly, it is a tiling of multi indices vectors.

Fejer quadrature rule uses only internal extrema of the polynomials, i.e. the weights of the endpoints are zero. It is also possible to employ a monodimensional trapezoidal rule as starting point to construct a multidimensional grid by the Smolyak algorithm. In this case, at every increase of the level, the monodimensional domain is discretized by a resolution double with respect the previous one. See the Appendix D for the analytical expression of the polynomials used.

3.5 Interaction between truncation and integration error

In the previous sections Polynomial Chaos expansion and the multi dimensional integration in order to compute PC coefficients have been introduced. An accurate estimation of statistical properties is dependent on two sources of error: related to truncation error of the PC $\mathcal{O}_T$ and $\mathcal{O}_I$ related to the integration on every coefficient β_k. The only way to reduce the truncation error is to increase the total degree of the expansion n_o, while increasing the level l of the quadrature in the Smolyak algorithm allows to reduce the integration error.[†] Unfortunately, due to the non-polynomial behavior of the integrand function, it is impossible to quantify each error contribution with an *a priori* estimate. In an ideal situation one should increase the level of the polynomial expansion when the level of the Sparse Grid quadrature is high enough to ensure that the integration error is negligible respect to the truncation ones. A detailed investigation of the interaction between the truncation error and the integration ones, in the context of the Sobol's sensitivities indices, can be found in [11].

In the present work different Sparse Grid techniques with increasing levels are coupled with Polynomial Chaos expansions of increasing degrees. This investigation aimed to identify, if it exists, the good choice between polynomial expansion and quadrature level. Numerical results for the g-function and nozzle flow are reported in the second part of this paper (§II).

4 ANOVA representation and dimensional decomposition

In this section we want to introduce briefly the ANOVA technique (§4.1) and a possible dimensional reduction strategy (§4.2) based on this dimensional decomposition.

[†] This is strictly true if we consider a class of functions wich have bounded mixed partial derivatives of order r, i.e. $f \in H^r([0,1]^d)$ with

$$H^r([0,1]^d) \stackrel{\text{def}}{=} \left\{ f : [0,1]^d \to \mathbb{R} : \max_{|\mathbf{k}|_\infty \leq r} \left\| \frac{\partial^{|\mathbf{k}|_1} f}{\partial^{k_1} \dots \partial^{k_d}} \right\| < \infty \right\}.$$

In this case the error estimation is [16]:

$$\epsilon(N) = \mathcal{O}\left(N^{-r} (\log N)^{(d-1)(r-1)} \right),$$

with N, the total number of quadrature points, which grows with the level of the Sparse Grid tensorization.

4.1 ANOVA dimensional decomposition

The numerical evaluation of a multidimensional function is affected by the *curse of dimensionality* [17]. A way to avoid this difficulty is related to Hilbert's thirteenth problem and the work of Kolmogorov and his student Arnold. At the end of the '50s Arnold refuted Hilbert's conjecture showing that any continuous function in the three-dimensional space $[0,1]^3$ can be represented as a superposition of continuous functions in two variables. Successively Kolmogorov generalized this result showing that it is true for every $f \in \mathcal{C}([0,1]^d)$ and that there is the possibility of superposition of only monodimensional functions. We can formalize this result saying that every function $f(\boldsymbol{\xi}) : [0,1]^d \to \mathbb{R}$ with $f \in \mathcal{C}([0,1]^d)$ has a representation

$$f(\boldsymbol{\xi}) = \sum_{i=1}^{2d+1} f_i(\sum_{j=1}^{d} \phi_{i,j}(\xi_j)), \tag{15}$$

where f_i and $\phi_{i,j}$ are one-dimensional continuous functions defined on $\mathbb{R}$ with $\phi_{i,j}$ independent on f_i. Here we follow the exposition of Griebel in [18], where many refinements of this results can be found. A fundamental question is how to choose the inner and outer functions in (15). An important result is that it is possible to replace the fixed number of variables $2d+1$ in (15) with a variable number m. From this result, with $m = d$ and $\phi_{i,j} = 1$ if $i = j$ and zero otherwise follows a so-called additive model:

$$f(\boldsymbol{\xi}) \approx \sum_{i=1}^{d} f_i \xi_i. \tag{16}$$

We can see the ANOVA decomposition, well known in statistics (see [19]), as a multivariate generalization of (16). The ANOVA decomposition of a d-dimensional function is

$$\begin{aligned} f(\boldsymbol{\xi}) = f_0 + \sum_{j_1}^{d} f_{j_1}(\xi_{j_1}) + \sum_{j_1<j_2}^{d} f_{j_1,j_2}(\xi_{j_1},\xi_{j_2}) \\ + \sum_{j_1<j_2<j_3}^{d} f_{j_1,j_2,j_3}(\xi_{j_1},\xi_{j_2},\xi_{j_3}) + \cdots + f_{j_1,\ldots,j_d}(\xi_{j_1},\ldots,\xi_{j_3}), \end{aligned} \tag{17}$$

or, in a more compact form

$$f(\boldsymbol{\xi}) = f_0 + \sum_{s=1}^{d} \sum_{j_1<\cdots<j_s}^{d} f_{j_1,\ldots,j_s}(\xi_{j_1},\ldots,\xi_{j_s}). \tag{18}$$

This form contains a fixed number of terms (2^d) and it is called ANOVA if

$$\int_0^1 f_{j_1,\ldots,j_s} \mathrm{d}\xi_k = 0 \quad \text{for all} \quad k = j_1,\ldots,j_s. \tag{19}$$

It follows from (16) that the members are orthogonal and can be expressed as integrals of $f(\boldsymbol{\xi})$:

$$\begin{aligned} f_0 &= \int f(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} \\ f_{j_1} &= \int f(\boldsymbol{\xi}) \prod_{i \neq j_1} \mathrm{d}\xi_i - f_0 \\ f_{j_1,j_2} &= \int f(\boldsymbol{\xi}) \prod_{i \neq j_1,j_2} \mathrm{d}\xi_i - f_{j_1} - f_{j_2} - f_0 \\ f_{j_1,j_2,j_3} &= \dots \end{aligned}$$

If the inputs $\boldsymbol{\xi}$ consist of independently uniform distributed random variables the variance $\sigma^2(f)$ can be decomposed as

$$\sigma^2(f) = \sum_{s=1}^{d} \sum_{j_1 < \dots < j_s}^{d} \sigma^2(f_{j_1,\dots,j_s}),$$

where

$$\sigma^2(f_{j_1,\dots,j_s}) = \int f^2_{j_1,\dots,j_s} \mathrm{d}\boldsymbol{\xi}. \tag{20}$$

From this decomposition is possible to define the global sensitivity index

$$S_{j_1,\dots,j_s} = \sigma^2(f_{j_1,\dots,j_s})/\sigma^2(f)$$

and the total sensitivity index (TSI_k) as the summation of all global sensitivity indexes in which the dimension k is present. If only the first order terms are considered the following relations hold

$$\begin{aligned} \sum_{i=1}^{d} S_i &\leq 1 \\ \sum_{i=1}^{d} TSI_i &\geq 1. \end{aligned}$$

4.2 Dimensional reduction strategy based on ANOVA

Based on the ANOVA decomposition of a function it is possible to evaluate the distribution of the overall variance respect to the ANOVA components, and if a threshold is fixed (e.g. 95% of the total variance), a truncation dimension is obtained. The truncation dimension is the effective dimensionality associated to the problem (of course it can depend on the chosen threshold).

The problem loses its original dimensionality (d) and it can be replaced with its approximation of dimension ($d-n$), where n is the number of the not significative dimensions. The dimensional reduction, based on the ANOVA analysis, allows to tackle the high dimensionality in the UQ context. The strategy we used in the following sections can be resumed as follows:

- performing ANOVA analysis on the original system (dimension d);

- fixing a threshold level on the variance, the number of unimportant dimensions (n) is obtained;
- the original model is reduced from dimension d to $d - n$ (for all the dimensions dropped the mean values is employed);
- a standard stochastical approach can be now performed on the reduced model.

Part II
Numerical results

Here we present two test cases that we used to explore the coupling between the Polynomial Chaos expansion and the Sparse Grid. The two test cases are an algebraic function, namely the so-called g-function, and a stochastic compressible flow in a de Laval nozzle where the geometry and the operating conditions are uncertain. This last test case have been chosen because it represents a physical problem with a reduced computational cost. We aim to compute the statistics, mean and variance, for the output of the two cases: the value of the g-function and the position of the shock in the nozzle. We introduced five, ten and fifteen uniform distributed uncertainties for the g-function (§5) and five and ten, uniform and normal distributed, uncertainties for the nozzle flow (§6). Different strategies, i.e. different kind of quadrature rules, are chosen to compute the coefficients (β_k) of the polynomial expansion (6). The effect of the mutual interaction between the sparse grid level and the degree of the polynomial expansion has been also investigated.

5 g-function

The g-function [20] is defined on the unit hypercube $g : [0,1]^d \to \mathbb{R}$ and its expression is:

$$g(\boldsymbol{\xi}) = \prod_{i=1}^{d} \frac{|4\xi_i - 2| + a_i}{1 + a_i} \quad \text{where} \quad a_i = \frac{i-1}{2}. \tag{21}$$

We introduced the randomness by modeling the independent variables (ξ_i) as stochastic variables uniform distributed on the domain of interest:
$\xi_i \sim \mathcal{U}[0,1]$ for $i = 1, \ldots, d$. This function is non-smooth and non-monotonous. A plot of the g-function in two variable is reported in figure 1.

This function is a challenging test both for the PC expansion, due to the presence of the absolute value which prevents the spectral convergence of the expansion, and for the cubature formula, due to the non-smooth behavior. For all the dimensions (d) the mean of the function $\mu(g(\boldsymbol{\xi})) = 1$, while the variance has an analytical description, function of the dimension d chosen:

$$\sigma^2(g(\boldsymbol{\xi})) + 1 = \prod_{i=1}^{d} \left(\frac{1}{3(1+a_i)^2} + 1 \right) \tag{22}$$

The coefficients a_i can be used to specified the role of the corresponding variables, the more coefficient is close to zero (by the way, in our case, the first coefficient is exactly zero), the more correspondent variable is important. Noting that a_i decreasing linearly with i, it is easy to see that growing up with the dimension d only less important variables are added.

We consider five §5.1, ten §5.2 and fifteen §5.3 dimensions problems. In this case statistic computations consist in computing the mean and variance of the function with the quadrature rules presented in §3.4 and identifying the minimum number of simulation that allows to reach a certain order of exactness

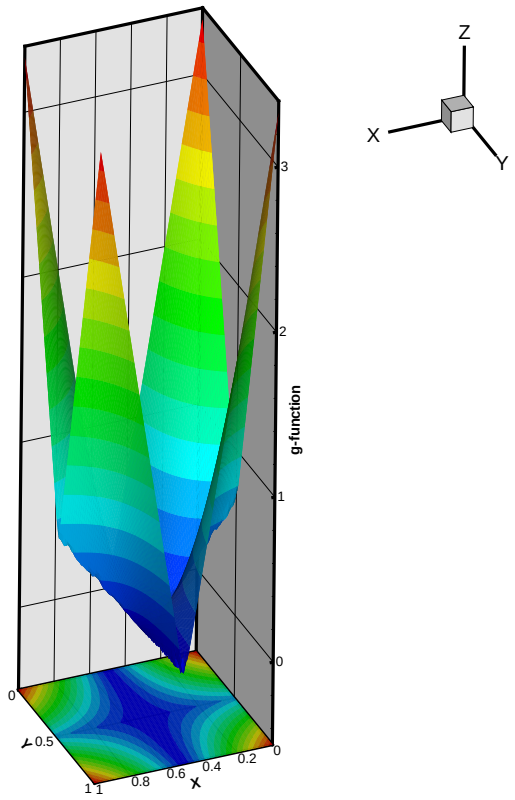

Figure 1: g-function in two variables.

on the statistical moments computed. The error is computed for the mean (err_μ) and variance (err_{σ^2}) as follows

$$\begin{aligned} err_\mu &= \frac{|\mu_{ref} - \mu|}{\mu_{ref}} \\ err_{\sigma^2} &= \frac{|\sigma^2_{ref} - \sigma^2|}{\sigma^2_{ref}}, \end{aligned} \tag{23}$$

where the reference solution is the exact ones ($\mu_{ref} = 1$ and Eq.22).

5.1 g-function with 5 uncertainties

In this section we show the results for the mean and the variance computed for the g-function in five stochastic dimensions. In figure 2, 3 and 4 the mean, the variance and a zoom on the interest domain are represented respectively.

We considered the following strategies: the full tensorization (full) from degree one to eleven; Petras' routine with a level up to fourteen; Fejer's rule with sparse grid level up to nine with a different polynomial structures (degrees from one to four); Clenshaw-Curtis (CC) quadrature rule with level up to eight with polynomial expansion of degrees from one to four.

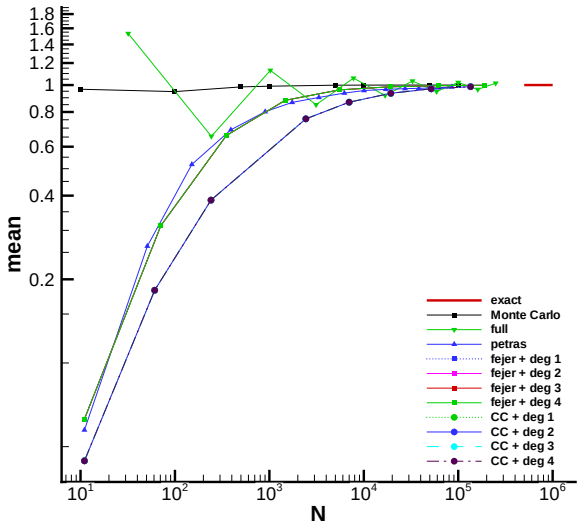

Figure 2: Mean values for the g-function ($d = 5$)

In the following we reported the table 1 with the magnitude of the error (23) for the five dimensional problem. We choose to indicate with the symbol $\square$ when a simulation has not performed, i.e. too high or too low number of points, and with the symbol $\boxtimes$ when an order of magnitude of the error has not observed, but the simulation has been performed, i.e. the method do not reach this accuracy.

If we consider both the mean and the variance, Monte Carlo seems to be the most efficient method. All the strategies employed more simulations to reach the same accuracy of the Monte Carlo, excepted the Fejer rule coupled with a polynomial expansion of degree four for the variance.

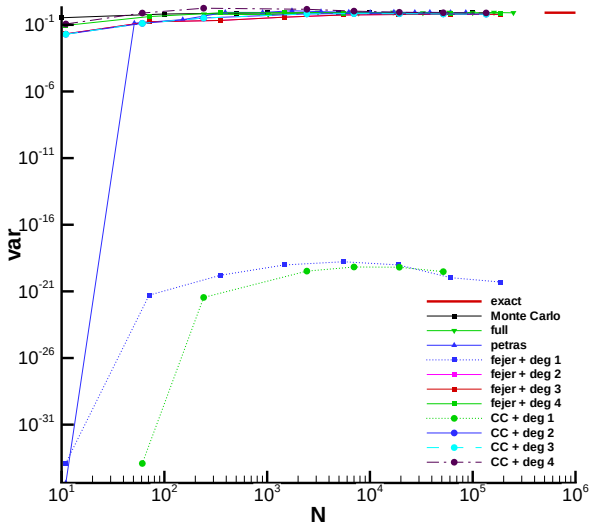

Figure 3: Variance values for the g-function ($d = 5$)

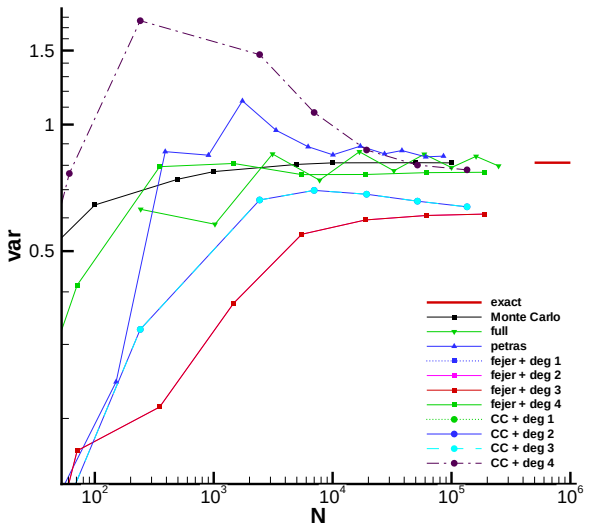

Figure 4: Variance (zoom) values for the g-function ($d = 5$)

Mean					
Method	E-01	E-02	E-03	E-04	E-05
MC	10	80	960	9040	55020
Full	32	7776	⊠	⊠	⊠
Petras	11	3343	⊠	⊠	⊠
Fejer + PC 1	11	5503	⊠	⊠	⊠
Fejer + PC 2	11	5503	⊠	⊠	⊠
Fejer + PC 3	11	5503	⊠	⊠	⊠
Fejer + PC 4	11	5503	⊠	⊠	⊠
CC + PC 1	11	19313	⊠	⊠	⊠
CC + PC 2	11	19313	⊠	⊠	⊠
CC + PC 3	11	19313	⊠	⊠	⊠
CC + PC 4	11	19313	⊠	⊠	⊠
Variance					
Method	E-01	E-02	E-03	E-04	E-05
MC	10	440	7010	⊠	⊠
Full	243	3125	⊠	⊠	⊠
Petras	51	6223	⊠	⊠	⊠
Fejer + PC 1	⊠	⊠	⊠	⊠	⊠
Fejer + PC 2	11	⊠	⊠	⊠	⊠
Fejer + PC 3	11	⊠	⊠	⊠	⊠
Fejer + PC 4	11	351	⊠	⊠	⊠
CC + PC 1	⊠	⊠	⊠	⊠	⊠
CC + PC 2	11	⊠	⊠	⊠	⊠
CC + PC 3	11	⊠	⊠	⊠	⊠
CC + PC 4	11	19313	⊠	⊠	⊠

Table 1: Error magnitude for g-function with five uncertainties. The symbol □ indicates when a simulation has not performed and the symbol ⊠ when an order of magnitude of the error has not observed.

5.2 g-function with 10 uncertainties

We now present the results obtained for the statistical evaluation of the g-function with a stochastical dimension of ten. We report the mean, the variance and a zoom on the interest zone in figure 5, 6 and 7 respectively. We compared the same methods of the previous section, but with a low level (or degree) due to a grow in the numerical cost for the moderate high dimension of the problem. The list of the simulations is here reported: full tensorization up to degree three; Petras routine up to level eight; Fejer rule coupled with polynomial expansion of degrees from one to four with level up to six; Clenshaw-Curtiss rule with level up to seven with polynomial expansions of degrees up to four.

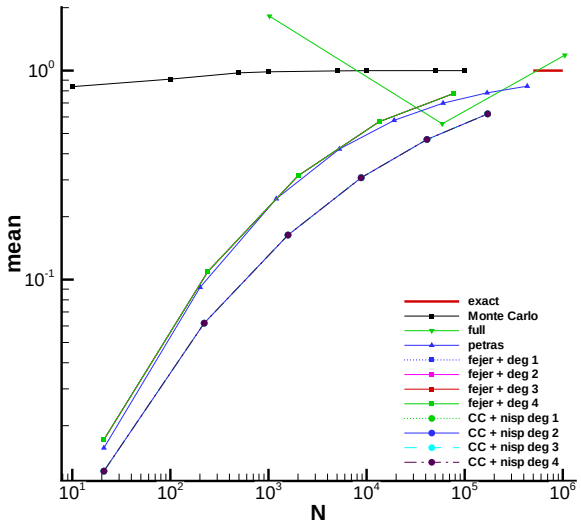

Figure 5: Mean values for the g-function ($d = 10$)

In the case of ten dimensions all the methods tested, included the Monte Carlo ones, are less predictive with respect to the five uncertainty case. All the methods reached a maximum accuracy of 10% with respect to Monte Carlo, except for the Fejer rule coupled with a polynomial expansion of degree four that reached 0.1%. This accuracy is achieved with an increasing, with respect to the Monte Carlo, of a factor nine for the number of simulations.

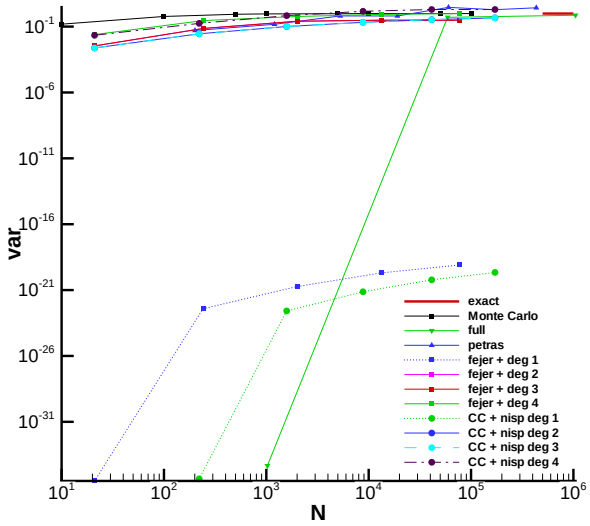

Figure 6: Variance values for the g-function ($d = 10$)

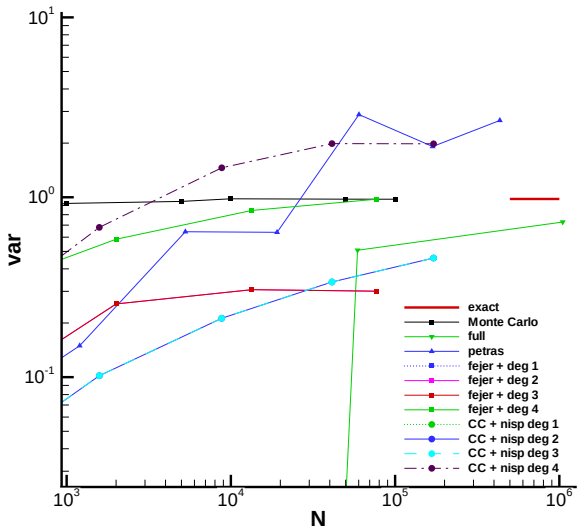

Figure 7: Variance (zoom) values for the g-function ($d = 10$)

	Mean				
Method	E-01	E-02	E-03	E-04	E-05
MC	10	90	1290	16800	⊠
Full	1024	⊠	⊠	⊠	⊠
Petras	21	⊠	⊠	⊠	⊠
Fejer + PC 1	21	⊠	⊠	⊠	⊠
Fejer + PC 2	21	⊠	⊠	⊠	⊠
Fejer + PC 3	21	⊠	⊠	⊠	⊠
Fejer + PC 4	21	⊠	⊠	⊠	⊠
CC + PC 1	21	⊠	⊠	⊠	⊠
CC + PC 2	21	⊠	⊠	⊠	⊠
CC + PC 3	21	⊠	⊠	⊠	⊠
CC + PC 4	21	⊠	⊠	⊠	⊠

	Mean				
Method	E-01	E-02	E-03	E-04	E-05
MC	10	540	8420	⊠	⊠
Full	59049	⊠	⊠	⊠	⊠
Petras	⊠	⊠	⊠	⊠	⊠
Fejer + PC 1	⊠	⊠	⊠	⊠	⊠
Fejer + PC 2	21	⊠	⊠	⊠	⊠
Fejer + PC 3	21	⊠	⊠	⊠	⊠
Fejer + PC 4	21	⊠	77505	⊠	⊠
CC + PC 1	⊠	⊠	⊠	⊠	⊠
CC + PC 2	21	⊠	⊠	⊠	⊠
CC + PC 3	21	⊠	⊠	⊠	⊠
CC + PC 4	⊠	⊠	⊠	⊠	⊠

Table 2: Error magnitude for g-function with ten uncertainties. The symbol □ indicates when a simulation has not performed and the symbol ⊠ when an order of magnitude of the error has not observed.

5.3 g-function with 15 uncertainties

The results for the case of the g-function with fifteen uncertainties is now reported in figure 8, 9 and 10, respectively for the mean, the variance and a zoom on the interest domain. We performed the following list of simulations: full tensorization of degree one[‡]; Petras routine up to level five; Fejer rule up to level four and Clenshaw-Curtis formula up to level five. The last two methods were coupled with polynomial expansions of degrees from one to four.

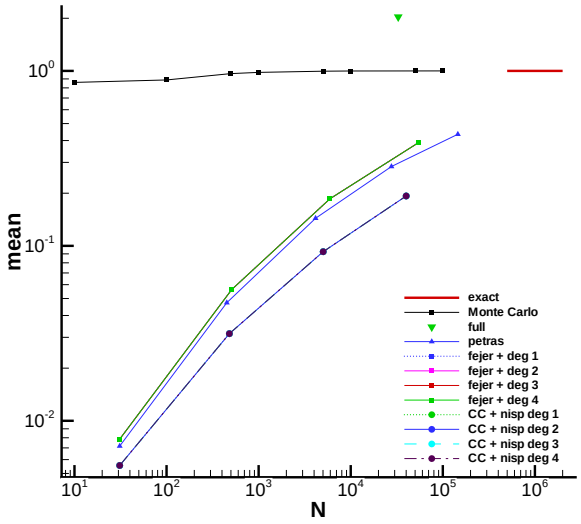

Figure 8: Mean values for the g-function ($d = 15$)

The case of fifteen uncertainties is the more difficult case tested in this work. Qualitatively the performances of all the methods are equal to the previous cases, but for all the methodologies there is a decrease in performances. We need more simulations for the same accuracy of the previous cases. The Monte Carlo is always the best method.

[‡]Note that the level two need $3^{15} = 14\,348\,907$ deterministic runs.

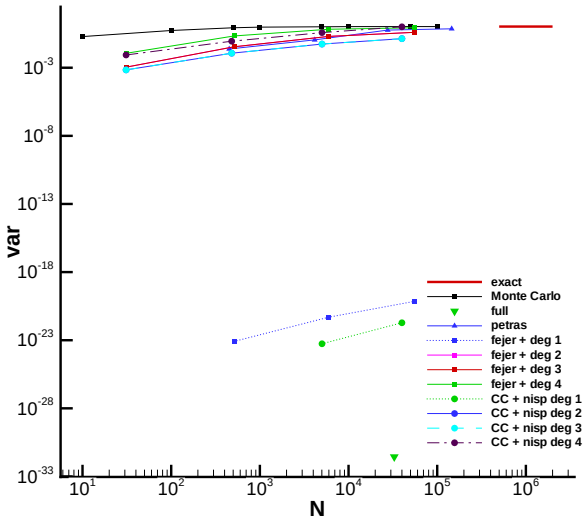

Figure 9: Variance values for the g-function ($d = 15$)

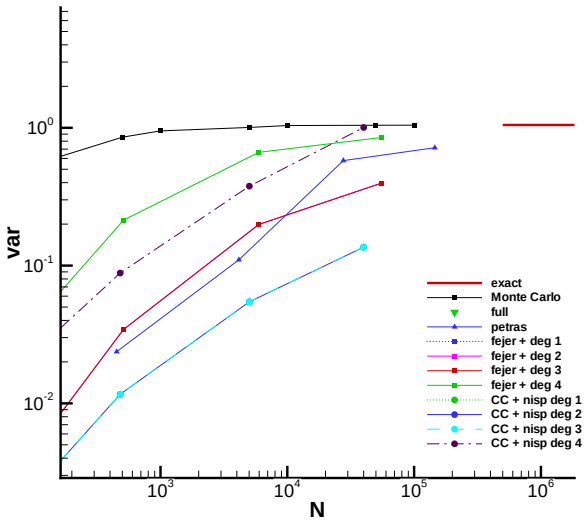

Figure 10: Variance (zoom) for the g-function ($d = 15$)

Mean					
Method	E-01	E-02	E-03	E-04	E-05
MC	10	150	2000	28990	⊠
Full	⊠	⊠	⊠	⊠	⊠
Petras	31	⊠	⊠	⊠	⊠
Fejer + PC 1	31	⊠	⊠	⊠	⊠
Fejer + PC 2	31	⊠	⊠	⊠	⊠
Fejer + PC 3	31	⊠	⊠	⊠	⊠
Fejer + PC 4	31	⊠	⊠	⊠	⊠
CC + PC 1	31	⊠	⊠	⊠	⊠
CC + PC 2	31	⊠	⊠	⊠	⊠
CC + PC 3	31	⊠	⊠	⊠	⊠
CC + PC 4	31	⊠	⊠	⊠	⊠
Variance					
Method	E-01	E-02	E-03	E-04	E-05
MC	10	600	⊠	⊠	⊠
Full	⊠	⊠	⊠	⊠	⊠
Petras	451	⊠	⊠	⊠	⊠
Fejer + PC 1	⊠	⊠	⊠	⊠	⊠
Fejer + PC 2	31	⊠	⊠	⊠	⊠
Fejer + PC 3	31	⊠	⊠	⊠	⊠
Fejer + PC 4	31	⊠	⊠	⊠	⊠
CC + PC 1	⊠	⊠	⊠	⊠	⊠
CC + PC 2	31	⊠	⊠	⊠	⊠
CC + PC 3	31	⊠	⊠	⊠	⊠
CC + PC 4	31	40001	⊠	⊠	⊠

Table 3: Error magnitude for g-function with fifteen uncertainties. The symbol □ indicates when a simulation has not performed and the symbol ⊠ when an order of magnitude of the error has not observed.

6 Nozzle flow

In this section stochastic compressible nozzle flow problem is considered. Algebraic equations for the nozzle flow have been solved. A reference condition is chosen in order to have a shock in the divergent part (see figure 11). Several uncertain parameters are considered: the polytropic coefficient γ; the ratio between outlet pressure and total pressure of the reservoir p_{es}/p_0; the ratio between the exit area and throat area A_e/A_t and some parameters in the equation for the geometrical description of the nozzle. The aim of the present research is to take into account uncertainty on these parameters in order to compute statistics of the output solution, i.e. position of the shock. We analyzed this flow with the presence of five and ten uncertainties. In the first case we deal with two geometrical parameters (α, β), while in the second case we considered seven geometrical parameters (α, β and α_i with $i = 1, \ldots, 5$). See the Appendix E for the geometrical description of the nozzle. For both the cases either uniform and Gaussian distributions are considered. We report two tables (4 and 5) with the values employed for the simulation. In the uniform case the minimum and maximum values are reported, while in the Gaussian case we report the mean and the standard deviation value.

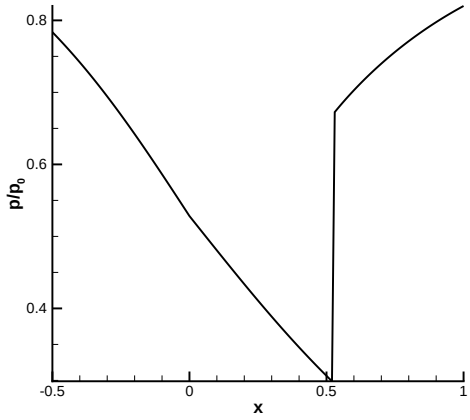

Figure 11: p/p_0 along the nozzle in a flow condition.

	Uniform		Gaussian	
Variable	Min	Max	Mean	Devstd
γ	1.39	1.41	1.4	7.0E-3
p_{es}/p_0	0.80145	0.85145	0.82645	4.1323E-3
A_e/A_t	1.485	1.515	1.5	7.5E-3
α	0	0.01	0.005	2.5E-5
β	0.475	0.525	0.5	2.5E-3

Table 4: Values for the nozzle flow in the five uncertainties case

	Uniform		Gaussian	
Variable	Min	Max	Mean	Devstd
γ	1.39	1.41	1.4	7.0E-3
p_{es}/p_0	0.80145	0.85145	0.82645	4.1323E-3
A_e/A_t	1.485	1.515	1.5	7.5E-3
α	0.00475	0.00525	0.005	2.5E-5
β	0.475	0.525	0.5	2.5E-3
α_1	-0.00525	-0.00475	-0.005	2.5E-5
α_2	-0.05	0.05	0	0.005
α_3	-0.05	0.05	0	0.005
α_4	-0.05	0.05	0	0.005
α_5	-0.05	0.05	0	0.005

Table 5: Values for the nozzle flow in the ten uncertainties case

Such an example in figure 12 we report the function represented by the shock position (x) as a function of two parameters, respectively the polytropic coefficient γ and the pressure ratio p_{es}/p_0 according to the previous table.

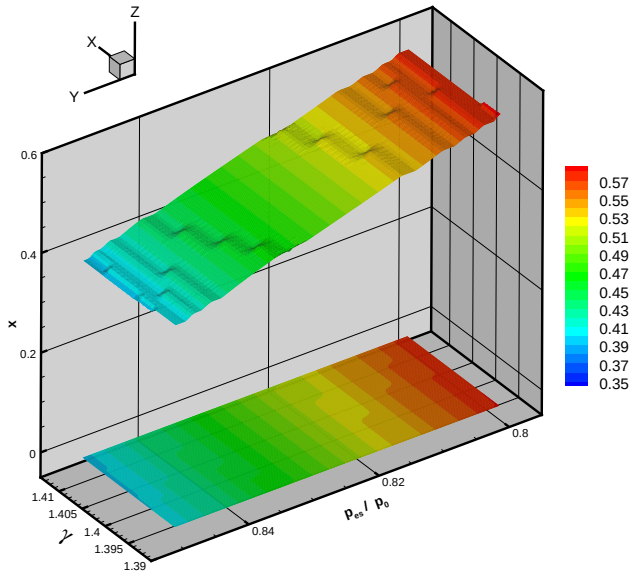

Figure 12: Bidimensional function for the shock position in the nozzle problem

6.1 Nozzle flow with five uncertainties

6.1.1 Uncertainties with uniform distribution

We report here the results of the nozzle problem with five uncertainties with a uniform probability density function. The mean, a zoom on the interest domain, the variance and a zoom on it are reported in figure 13, 14, 15, and 16 respectively. The reference solution is the Monte Carlo ones (200000 runs)

for all the computations. We tested the full tensorization from degree one to ten; the Petras algorithm with level up to eighteen; Clenshaw-Curtis quadrature with level up to nine coupled with polynomial expansions of degrees from one to four. We tried also to couple a partial tensorization, obtained starting from the Legendre quadrature rule by the Smolyak algorithm, with a polynomial expansion of maximum polynomial degree ($n_o = l - 1$ where l is the level of the sparse grid).

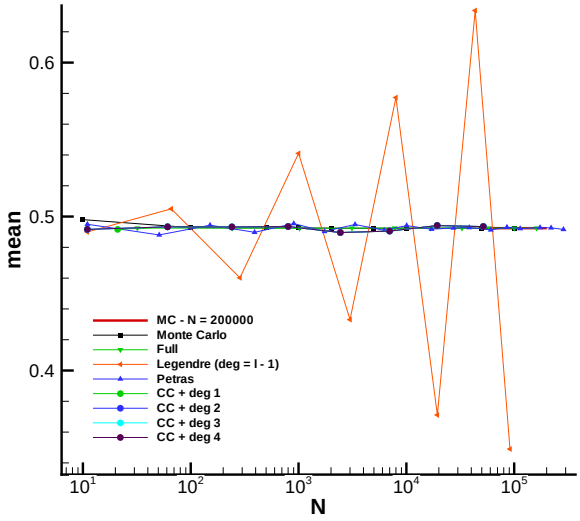

Figure 13: Mean for the nozzle problem with five uniform uncertainties

In the following the error on the mean and on the variance is shown in the case of five uniform distributed (table 6).

All the methods perform better than their counterpart used on the g-function, but, in analogy with the g-function case, no combination between Sparse grid and Polynomial Chaos seem to be efficient and accurate as the Monte Carlo method. However the combination of the Clenshaw-Curtis quadrature and a low degree expansion for the Polynomial Chaos shows to be more efficient than MC to reach an accuracy up to 0.1% for the mean and 1% for the variance.

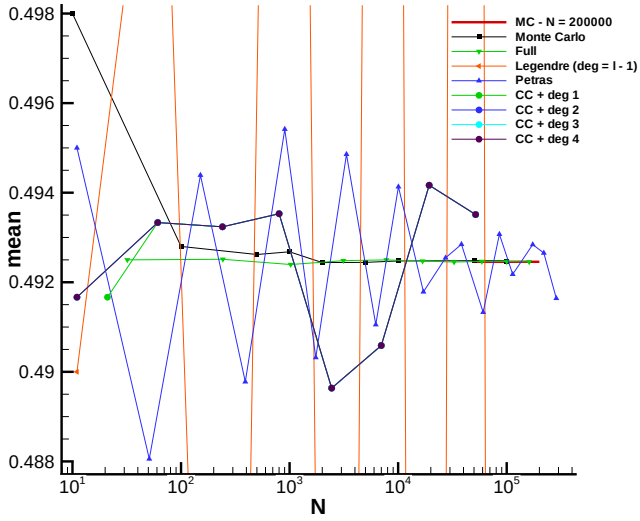

Figure 14: Mean (zoom) for the nozzle problem with five uniform uncertainties

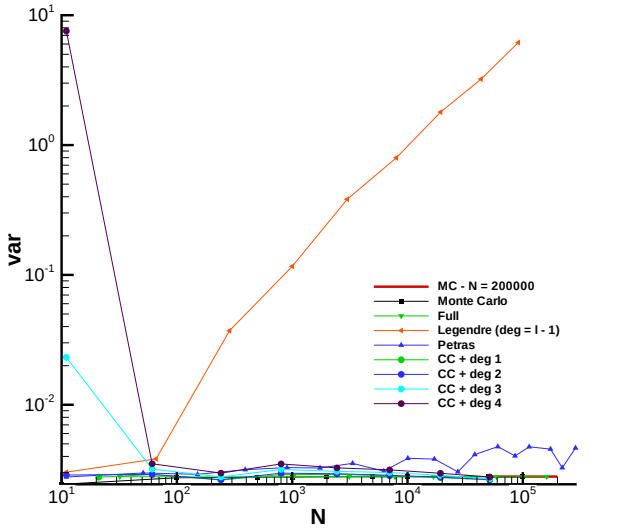

Figure 15: Variance for the nozzle problem with five uniform uncertainties

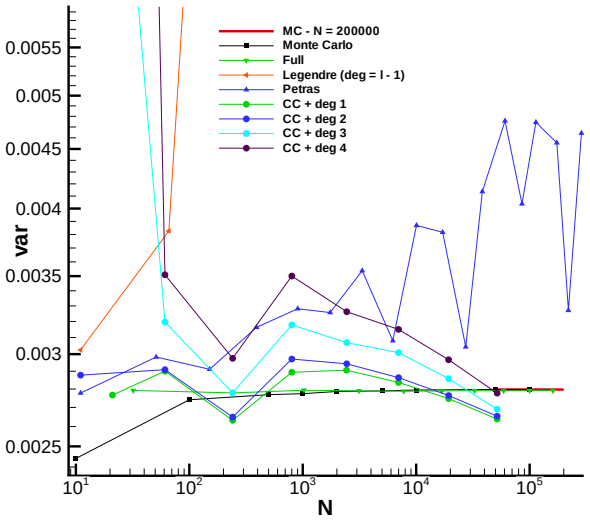

Figure 16: Variance (zoom) for the nozzle problem with five uniform uncertainties

Mean

Method	E-01	E-02	E-03	E-04	E-05
MC	□	10	20	260	1420
Full	□	□	□	32	3125
Legendre	11	⊠	⊠	⊠	⊠
Petras	□	□	11	⊠	⊠
CC + PC 1	□	□	21	⊠	⊠
CC + PC 2	□	□	21	⊠	⊠
CC + PC 3	□	□	21	⊠	⊠
CC + PC 4	□	□	21	⊠	⊠

Variance

Method	E-01	E-02	E-03	E-04	E-05
MC	10	100	860	9010	71940
Full	□	□	32	⊠	⊠
Legendre	⊠	⊠	⊠	⊠	⊠
Petras	11	⊠	⊠	⊠	⊠
CC + PC 1	□	21	⊠	⊠	⊠
CC + PC 2	□	21	⊠	⊠	⊠
CC + PC 3	61	2443	⊠	⊠	⊠
CC + PC 4	61	19313	51713	□	□

Table 6: Error magnitude for the nozzle flow with five uniform uncertainties. The symbol □ indicates when a simulation has not performed and the symbol ⊠ when an order of magnitude of the error has not observed.

6.1.2 Uncertainties with Gaussian distribution

Here the results for the compressible nozzle problem with the five uncertainties with Gaussian probability density function are reported. The mean, the variance and a zoom on the interest domain are reported in figure 17, 18, and 19 respectively. The reference solution is Monte Carlo. The quadrature techniques coupled with polynomial chaos are: full tensorization up to nine degree; Petras routine up to six level; Clenshaw-Curtis up to level nine and Fejer rule up to level eight. Violating what is strictly prescribed by the Askey scheme, we used for Clenshaw-Curtis and Fejer the sparse grid to compute the coefficient of a polynomial expansion relating to uniform distributed variables, i.e. Legendre basis polynomial instead of Hermite ones. The same was done also for the Petras algorithm. In this case we grow up until the ten level for Clenshaw-Curtis and level eight for the Petras algorithm. In this case we tried also a simple collocation: we computed directly the mean and variance from the sparse grid weights without a polynomial basis framework.

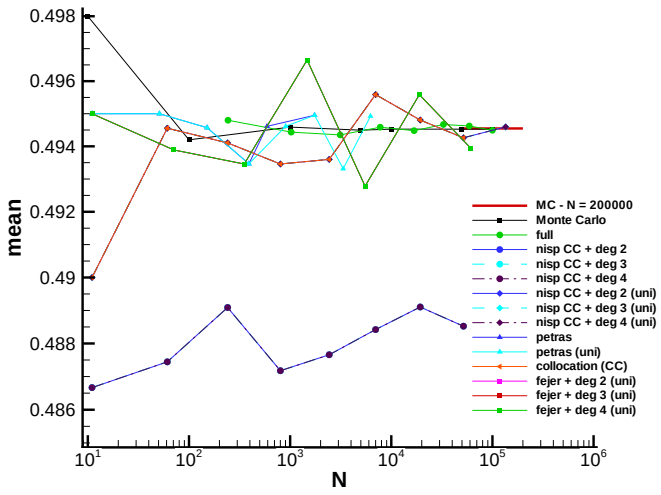

Figure 17: Mean for the nozzle problem with five Gaussian uncertainties

All the methods fail to be competitive with Monte Carlo in accuracy and efficiency and they performed worst than the same dimensional uniform case.

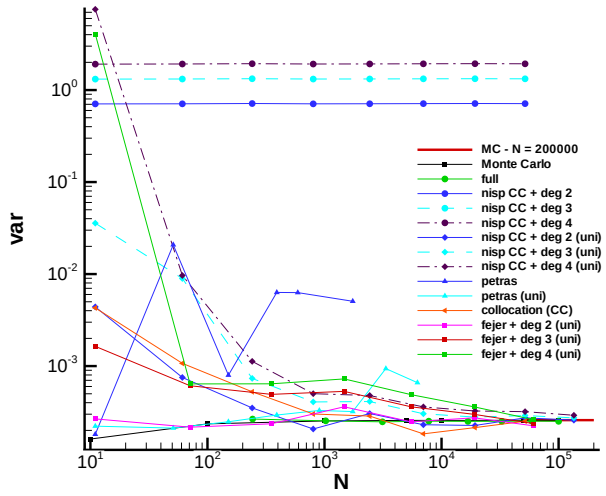

Figure 18: Variance for the nozzle problem with five Gaussian uncertainties

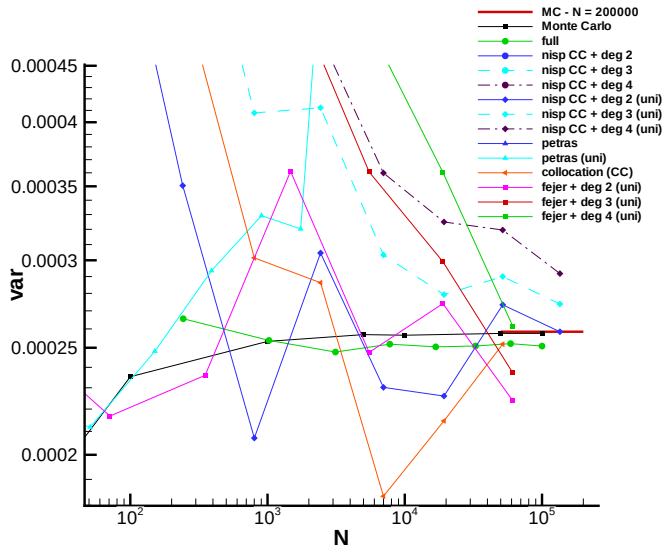

Figure 19: Variance (zoom) for the nozzle problem with five Gaussian uncertainties

Mean					
Method	E-01	E-02	E-03	E-04	E-05
MC	□	□	10	50	5110
Full	□	□	□	243	⊠
CC + PC 2	□	11	⊠	⊠	⊠
CC + PC 3	□	11	⊠	⊠	⊠
CC + PC 4	□	11	⊠	⊠	⊠
CC + PC 2 (uni)	□	□	11	19313	⊠
CC + PC 3 (uni)	□	□	11	19313	⊠
CC + PC 4 (uni)	□	□	11	19313	⊠
Petras	□	□	□	11	⊠
Petras (uni)	□	□	11	⊠	⊠
Collocation (CC)	□	□	11	⊠	⊠
Fejer + PC 2 (uni)	□	□	11	⊠	⊠
Fejer + PC 3 (uni)	□	□	11	⊠	⊠
Fejer + PC 4 (uni)	□	□	11	⊠	⊠

Variance					
Method	E-01	E-02	E-03	E-04	E-05
MC	10	80	3350	⊠	⊠
Full	□	243	⊠	⊠	⊠
CC + PC 2	⊠	⊠	⊠	⊠	⊠
CC + PC 3	⊠	⊠	⊠	⊠	⊠
CC + PC 4	⊠	⊠	⊠	⊠	⊠
CC + PC 2 (uni)	241	51713	□	□	□
CC + PC 3 (uni)	801	⊠	⊠	⊠	⊠
CC + PC 4 (uni)	801	⊠	⊠	⊠	⊠
Petras	⊠	⊠	⊠	⊠	⊠
Petras (uni)	⊠	⊠	⊠	⊠	⊠
Collocation (CC)	801	51713	⊠	⊠	⊠
Fejer + PC 2 (uni)	11	⊠	⊠	⊠	⊠
Fejer + PC 3 (uni)	5503	61183	⊠	⊠	⊠
Fejer + PC 4 (uni)	5503	61183	⊠	⊠	⊠

Table 7: Error magnitude for the nozzle flow with five Gaussian uncertainties. The symbol □ indicates when a simulation has not performed and the symbol ⊠ when an order of magnitude of the error has not observed.

6.2 Nozzle flow with ten uncertainties

In this section we extended the number of the uncertainties from five to ten for the same nozzle problem (see Appendix E for more details).

6.2.1 Uncertainties with uniform distribution

The results for the nozzle problem with ten uniformed distributed uncertainties are here reported: mean (figure 20), the zoom on the interest domain (figure 21) and variance (figure 22). The reference values are the Monte Carlo ones. We considered several strategies: full tensorization of degree two; Petras algorithm up to level seven; Clenshaw-Curtis rule up to level seven coupled with polynomial expansion of degrees from one to four; Kronrod-Patterson quadrature rule up to level five coupled with polynomial expansion of degrees two and three; Fejer rule up to level six with a polynomial expansion of two; the trapezoidal rule up to level six with polynomial expansions of degrees two, three and four and Legendre quadrature rule up to level seven with polynomial expansion of degrees two and three.

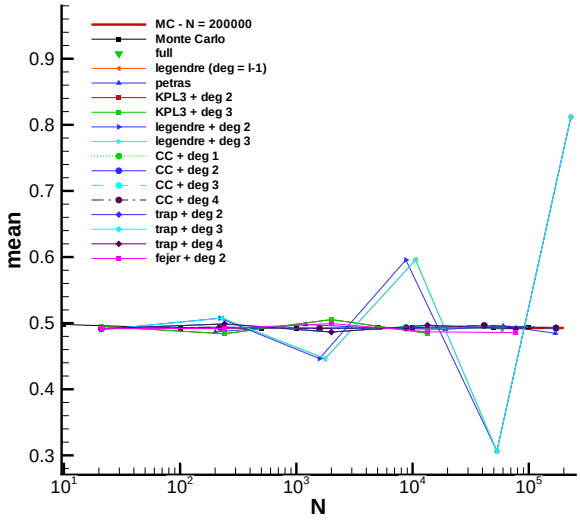

Figure 20: Mean for the nozzle problem with ten uniform uncertainties

This case appears more difficult than the previous one due to the higher dimension. The Monte Carlo performs always better than the other methods and the Petras, Legendre and trapezoidal rule failed to reach an accuracy of 10%, with respect to Monte Carlo, in all the range of simulations performed.

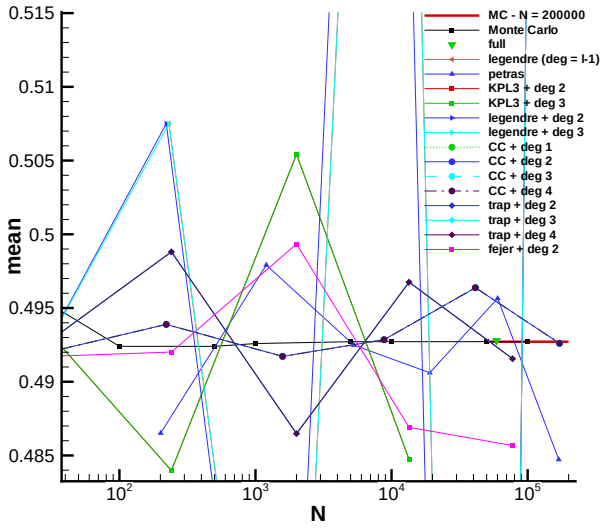

Figure 21: Mean (zoom) for the nozzle problem with ten uniform uncertainties

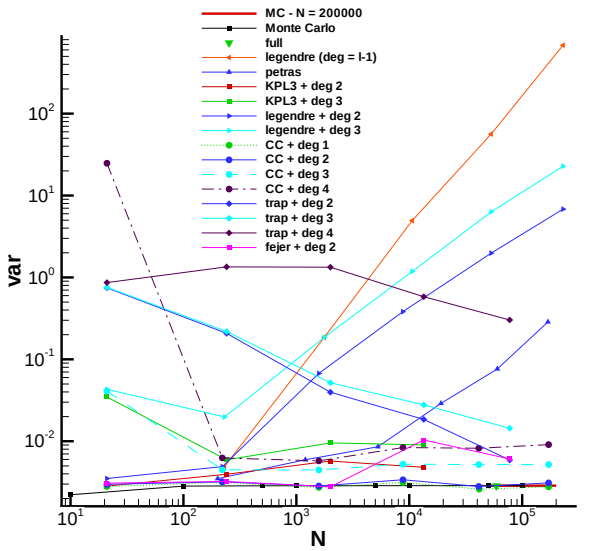

Figure 22: Variance for the nozzle problem with ten uniform uncertainties

Mean

Method	E-01	E-02	E-03	E-04	E-05	E-06
MC	□	10	20	100	2110	35960
Full	□	□	□	59049	□	□
Leg ($l = n_o + 1$)	231	⊠	⊠	⊠	⊠	⊠
Petras	201	⊠	⊠	⊠	⊠	⊠
KPL3 + PC 2	□	21	⊠	⊠	⊠	⊠
KPL3 + PC 3	□	21	⊠	⊠	⊠	⊠
Leg + PC 2	21	⊠	⊠	⊠	⊠	⊠
Leg + PC 3	21	⊠	⊠	⊠	⊠	⊠
CC + PC 1	□	□	21	171425	□	□
CC + PC 2	□	□	21	171425	□	□
CC + PC 3	□	□	21	171425	□	□
CC + PC 4	□	□	21	171425	□	□
Trap + PC 2	□	21	13441	⊠	⊠	⊠
Trap+ PC 3	□	21	13441	⊠	⊠	⊠
Trap+ PC 4	□	21	13441	⊠	⊠	⊠
Fejer + PC 2	□	21	⊠	⊠	⊠	⊠

Variance

Method	E-01	E-02	E-03	E-04	E-05	E-06
MC	10	60	230	7510	⊠	⊠
Full	□	□	59049	□	□	⊠
Leg ($l = n_o + 1$)	⊠	⊠	⊠	⊠	⊠	⊠
Petras	⊠	⊠	⊠	⊠	⊠	⊠
KPL3 + PC 2	21	⊠	⊠	⊠	⊠	⊠
KPL3 + PC 3	⊠	⊠	⊠	⊠	⊠	⊠
Leg + PC 2	⊠	⊠	⊠	⊠	⊠	⊠
Leg + PC 3	⊠	⊠	⊠	⊠	⊠	⊠
CC + PC 1	□	21	⊠	⊠	⊠	⊠
CC + PC 2	21	41625	⊠	⊠	⊠	⊠
CC + PC 3	221	⊠	⊠	⊠	⊠	⊠
CC + PC 4	⊠	⊠	⊠	⊠	⊠	⊠
Trap + PC 2	⊠	⊠	⊠	⊠	⊠	⊠
Trap + PC 3	⊠	⊠	⊠	⊠	⊠	⊠
Trap + PC 4	⊠	⊠	⊠	⊠	⊠	⊠
Fejer + PC 2	⊠	⊠	⊠	⊠	⊠	⊠

Table 8: Error magnitude for the nozzle flow with ten uniform uncertainties. The symbol □ indicates when a simulation has not performed and the symbol ⊠ when an order of magnitude of the error has not observed.

6.2.2 Uncertainties with Gaussian distribution

The last cases analyzed was the nozzle problem with ten Gaussian distributed uncertainties. The mean and the variance are reported in figure 23 and 24 respectively. The reference values are always the Monte Carlo ones. We compared the full tensorization of degree two; the Petras routine up to level seven; Clenshaw-Curtis quadrature formula with level up to six coupled with polynomial expansion of degree from one to four and the trapezoidal rule, up to level six coupled with polynomial structures of degrees from two to four.

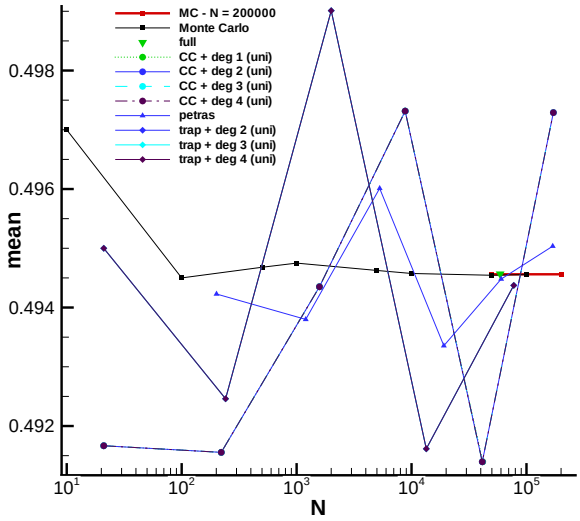

Figure 23: Mean for the nozzle problem with ten Gaussian uncertainties

In this case the general trend is the same of the uniform ones, but it shows to be more difficult. Petras and trapezoidal rule failed to achieve an accuracy of 10% and the same occurred also for the Clenshaw-Curtis coupled with an high polynomial degree (three or four).

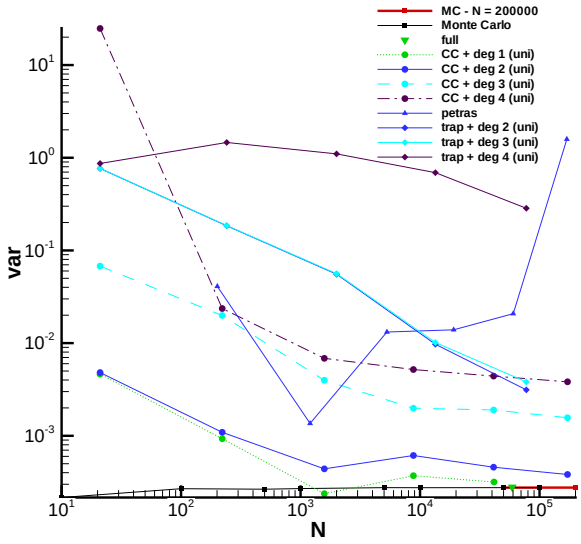

Figure 24: Variance for the nozzle problem with ten Gaussian uncertainties

Mean

Method	E-01	E-02	E-03	E-04	E-05	E-06
MC	□	□	10	50	5870	⊠
Full	□	□	□	□	□	59049
CC + PC 1 (uni)	□	□	21	⊠	⊠	⊠
CC + PC 2 (uni)	□	□	21	⊠	⊠	⊠
CC + PC 3 (uni)	□	□	21	⊠	⊠	⊠
CC + PC 4 (uni)	□	□	21	⊠	⊠	⊠
Petras	□	□	201	60225	⊠	⊠
Trap + PC 2 (uni)	□	□	21	77505	□	□
Trap + PC 3 (uni)	□	□	21	77505	□	□
Trap + PC 4 (uni)	□	□	21	77505	□	□

Variance

Method	E-01	E-02	E-03	E-04	E-05	E-06
MC	10	60	3350	70640	⊠	⊠
Full	□	59049	□	□	□	⊠
CC + PC 1 (uni)	1581	171425	□	□	□	⊠
CC + PC 2 (uni)	1581	⊠	⊠	⊠	⊠	⊠
CC + PC 3 (uni)	⊠	⊠	⊠	⊠	⊠	⊠
CC + PC 4 (uni)	⊠	⊠	⊠	⊠	⊠	⊠
Petras	⊠	⊠	⊠	⊠	⊠	⊠
Trap + PC 2 (uni)	⊠	⊠	⊠	⊠	⊠	⊠
Trap + PC 3 (uni)	⊠	⊠	⊠	⊠	⊠	⊠
Trap + PC 4 (uni)	⊠	⊠	⊠	⊠	⊠	⊠

Table 9: Error magnitude for the nozzle flow with ten gaussian uncertainties. The symbol □ indicates when a simulation has not performed and the symbol ⊠ when an order of magnitude of the error has not observed.

7 ANOVA analysis and dimension reduction

In this section we want to explore the strategy described in §4.2.

We aim to compute the error between the d-dimensional function (f) and its $(d-n)$-dimensional surrogate ($\tilde{f}$) both for the mean and for the variance as follows

$$err_{\mu} = \frac{\mu(f) - \mu(\tilde{f})}{\mu(f)} \tag{24}$$

$$err_{\sigma^2} = \frac{\sigma^2(f) - \sigma^2(\tilde{f})}{\sigma^2(f)}. \tag{25}$$

$$\tag{26}$$

We analyzed the case of the g-function in (§7.1) and the nozzle flow (§7.2) in the case of uniform distributed variables.

7.1 g-function

Here we exploit the analytical behavior of the g-function to compute statistics from the dimensional reduced model without perform computations on it. The g-function was built with terms losing their importance when the dimension increase, so a dimensional reduction can be obtained drop the last terms. We consider a d-dimensional g-function and we neglect the last n terms, i.e. the stochastic dimension is reduced and the terms dropped are substituted with their mean value (always 0.5 in this case). We make a substitution of the function $g = g(\boldsymbol{\xi}) : \mathbb{R}^d \to \mathbb{R}$ with his surrogate $\tilde{g} = g(\tilde{\boldsymbol{\xi}}) : \mathbb{R}^{d-n} \to \mathbb{R}$ where the surrogate vector $\tilde{\boldsymbol{\xi}} \in \mathbb{R}^{d-n}$ is defined as follows $\tilde{\boldsymbol{\xi}} \stackrel{\text{def}}{=} (\xi_1, \ldots, \xi_{d-n})$. In this case is possible to evaluate analytically the exact value for the mean and variance:

$$\mu(\tilde{g}) = \prod_{i=d-n+1}^{d} \frac{a_i}{i + a_1} \tag{27}$$

$$\sigma^2(\tilde{g}) = \prod_{i=d-n+1}^{d} \left(\frac{a_i}{i + a_1}\right)^2 \left(\prod_{i=1}^{d-n} \frac{3a_i^2 + 6a_i + 4}{3(1 + a_i)^2} - 1\right).$$

These relations, with (22), allow to compute trivially also the error made evaluating the mean and the variance on $\tilde{g}$ instead of g. In table 10 we report the error on the mean and variance corresponding to a threshold on the total variance ($\%\sigma^2$) neglecting the last n uncertainties of the original d-dimensional problem. The table 10 consider a possible dimension reduction of the three cases analyzed in (§5): five, ten and fifteen uniform distributed uncertainties.

In conclusion the dimension reduction, based on the ANOVA analysis, seems to be a non accurate technique for the g-function.

7.2 Nozzle flow

In this section we want to present some results on the dimensional reduction of the model based on the ANOVA analysis, for the nozzle flow problem.

d	n	$\%\sigma^2$	err_μ	err_{σ^2}
5	2	80	0.60	0.87
10	6	82	0.82	0.97
15	10	73	0.87	0.99

Table 10: Error on the mean and variance after ANOVA based reduction for the g-function.

As in the case of the g-function, presented in the previous section, we performed an ANOVA analysis on the complete problem to quantify the contribution of each uncertainties to the total variance. This preliminary analysis allows to evaluate the possibility of a dimensional reduction on the stochastic space. We analyzed the two cases of five and ten uncertainties with uniform distribution.

The ANOVA analysis shows that a large amount of the total variance is linked to the uncertainty on the pressure ratio (p_{es}/p_0). For the five uncertainties problem the total contribution is more than 98% considering only the uncertainties on the pressure ratio, while for the ten uncertainties case it is 96%. This analysis seems to justify a model reduction of four and nine uncertainties respectively for the five and ten uncertainties case.

For the nozzle problem flow the analytical values for the mean and variance are unknown, however we can perform a Monte Carlo simulation on the reduced case to estimate an 'exact' value for the reduced problem. In this manner we are able to quantify the error associated to the prescribed variance contribution linked to the selection of only one uncertainty (on the pressure ratio p_{es}/p_0).

The table 11 resume all the results obtained. The error for the mean (err_μ) and variance (err_{σ^2}) is computed as reported in 24.

d	n	$\%\sigma^2$	err_μ	err_{σ^2}
5	4	98	-1.84E-04	0.014
10	9	96	3.40E-04	0.039

Table 11: Error on the mean and variance after ANOVA based reduction for the nozzle problem

Differently from the g-function case, analyzing the nozzle flow the stochastic reduction seems to be a promising strategy. We report the results (table 12) obtained performing a Polynomial Chaos analysis, with a full tensorization, on the flow problem with only one uncertainty. These results are been compared with their counterparts reported in the tables 6 and 8.

In this case the stochastic dimension reduction based on a preliminary ANOVA analysis shows comfortable results, in fact if one wants to evaluate the variance with an accuracy of the 1%, in the worst case he can obtain a reduction of the number of simulation of a factor 15. For an error of magnitude 0.1%, a factor of 10 in the number of numerical simulations, is achieved for the mean.

Five uncertainties

Mean						
Method	E-01	E-02	E-03	E-04	E-05	E-06
MC	□	10	20	260	1420	⊠
Full (reduced)	□	□	2	62	⊠	⊠
Variance						
Method	E-01	E-02	E-03	E-04	E-05	E-06
MC	10	100	860	9010	71940	⊠
Full (reduced)	□	2	⊠	⊠	⊠	⊠

Ten uncertainties

Mean						
Method	E-01	E-02	E-03	E-04	E-05	E-06
MC	□	10	20	100	2110	35960
Full (reduced)	□	□	2	84	⊠	⊠
Variance						
Method	E-01	E-02	E-03	E-04	E-05	E-06
MC	10	60	230	7510	⊠	⊠
Full (reduced)	⊠	4	⊠	⊠	⊠	⊠

Table 12: Error magnitude for the nozzle flow after ANOVA based reduction. The symbol □ indicates when a simulation has not performed and the symbol ⊠ when an order of magnitude of the error has not observed.

8 Conclusions

In this work we explored potentialities and deficiencies of the coupling between Polynomial Chaos and Sparse Grid. We analyzed two different problem: the g-function, and the classical stochastic compressible flow in a de-Laval nozzle.

We explored several kind on Sparse Grid quadrature and we compared all the results with the Monte Carlo method. No combination of Sparse Grid and Polynomial Chaos shows comparable results, with Monte Carlo ones, in term of accuracy and efficiency, i.e. an higher number of simulations is always required to reach a fixed magnitude of error. The computation of the variance has been more difficult than mean to compute, the g-function resulted to be more difficult to analyze respect to the nozzle flow problem.

The possibility of a reduction of the stochastical dimension of the problem was also explored. A preliminary analytical analysis on the g-function showed that the reduction for this kind of function is useless. For the nozzle flow problem an estimate of the error, made using the dimensional reduced model, was performed using the Monte Carlo values of the complete and reduced problem. This results showed that for this kind of problem this strategy is efficient and allows to a good estimate for the statistical moments of the complete model. A full tensorization with a Polynomial Chaos expansion has been used on the stochastical dimension reduced model and it has been showed that, in the worst case (nozzle flow), a reduction of a factor 10 is achieved on the number of simulations required.

9 Acknowledgements

Remi Abgrall has been partially supported by the ERC Advanced Grant ADDECCO N. 226316. Gianluca Geraci has been fully supported by the ERC Advanced Grant ADDECCO N. 226316. The authors are greatly indebted with Jean-Marc Martinez for his insightful comments and support.

References

[1] Patrick Knupp and Kambiz Salari. *Verification of Computer Codes in Computational Science and Engineering*. Chapman & Hall, 2003.

[2] Richard Askey and James Wilson. Some basic hypergeometric orthogonal polynomials that generalize Jacobi polynomials. *Memoirs of the American Mathematical Society*, 54(319), 1985.

[3] Ivo Babuška, Fabio Nobile, and Rauúl Tempone. A Stochastic Collocation Method for Elliptic Partial Differential Equations with Random Input Data. *SIAM Review*, 52(2):317, 2010.

[4] Lionel Mathelin and Yousuff M. Hussaini. A Stochastic Collocation Algorithm for Uncertainty Analysis, 2003.

[5] G.J.Alex Loeven, J. A. S. Witteveen, and H Bijl. Probabilistic collocation: an efficient non-intrusive approach for arbitrarily distributed parametric uncertainties. In *AIAA, Aerospace sciences meeting and exhibit*, number January, pages 2007–317, 2007.

[6] Géraud Blatman and Bruno Sudret. Adaptive sparse polynomial chaos expansion based on least angle regression. *Journal of Computational Physics*, 230(6):2345–2367, March 2011.

[7] J.A.S. Witteveen and Gianluca Iaccarino. Simplex Elements Stochastic Collocation in Higher-Dimensional Probability Spaces. In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 12 - 15 April 2010, Orlando, Florida*, number April, 2010.

[8] Jasmine Foo, Xiaoliang Wan, and George Em Karniadakis. The multi-element probabilistic collocation method (ME-PCM): Error analysis and applications. *Journal of Computational Physics*, 227(22):9572–9595, November 2008.

[9] Xiang Ma and Nicholas Zabaras. An adaptive high-dimensional stochastic model representation technique for the solution of stochastic partial differential equations. *Journal of Computational Physics*, 229(10):3884–3915, May 2010.

[10] Nitin Agarwal and N.R. Aluru. A domain adaptive stochastic collocation approach for analysis of MEMS under uncertainties. *Journal of Computational Physics*, 228(20):7662–7688, November 2009.

[11] Thierry Crestaux, Olivier Le Maître, and Jean-Marc Martinez. Polynomial chaos expansion for sensitivity analysis. *Reliability Engineering & System Safety*, 94(7):1161–1172, July 2009.

[12] Norbert Wiener. The Homogeneous Chaos. *American Journal of Mathematics*, 60(4):897–936, 1938.

[13] S. A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Math. Dokl.*, (4):240–243, 1963.

[14] Thomas Gerstner and Michael Griebel. Numerical integration using sparse grids. *Numerical Algorithms*, 18:209–232, 1998.

[15] Knut Petras. Smolyak cubature of given polynomial degree with few nodes for increasing dimension. *Numerische Mathematik*, 93:729–753, 2003.

[16] M. Holtz. *Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance*. Dissertation, Institut für Numerische Simulation, Universität Bonn, 2008.

[17] Richard Ernest Bellman. *Adaptive control processes: a guided tour*. Princeton University Press, 1961.

[18] M. Griebel. Sparse grids and related approximation schemes for higher dimensional problems. In L. Pardo, A. Pinkus, E. Suli, and M.J. Todd, editors, *Foundations of Computational Mathematics (FoCM05), Santander*, pages 106–161. Cambridge University Press, 2006.

[19] R. Fisher. *Statistical Methods for Research Workers*. Olivier & Boyd, 1925.

[20] Andrea Saltelli and Sobol Ilya M. About the use of rank transformation in sensitivity analysis of model output. *Reliability Engineering & System Safety*, 50(3):225–239, 1995.

Part III
Appendix

A Polynomial basis and truncated expansion

In this Appendix we want to show how to calculate the polynomial truncated expansion (Eq. 6), for a two-dimensional problem ($d = 2$), for a fixed total polynomial expansion order. For the example Hermite basis (see table 13 and figure (25)) is chosen[§].

n	$\psi_n(x)$
0	1
1	x
2	$x^2 - 1$
3	$x^3 - 3x$
4	$x^4 - 6x^2 + 3$
5	$x^5 - 10x^3 + 15x$

Table 13: First five Hermite polynomials.

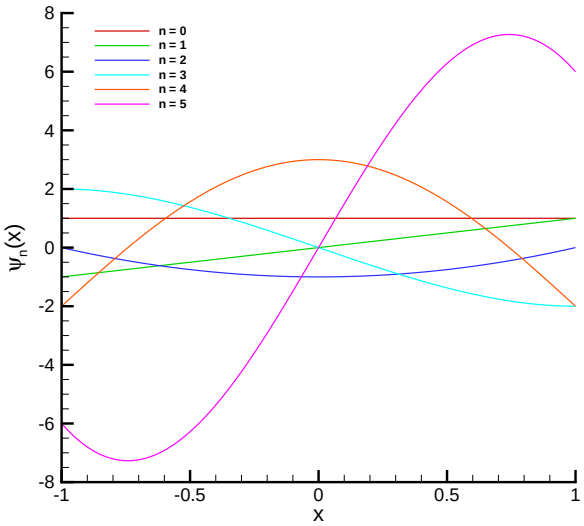

Figure 25: First five Hermite polynomials.

[§] We choose the so called "probabilists' Hermite polynomials basis".

Total order $n_o = 0$

First we compute the number of terms required:

$$N_{tot} = P + 1 = \frac{(n_o + d)!}{n_o!\, d!} = \frac{(0+2)!}{0!\, 2!} = 1.$$

The solution will be expanded as

$$u(\boldsymbol{\xi}) = \sum_{k=0}^{P} \beta_k \Psi_k(\boldsymbol{\xi}) = \beta_0 \Psi_0(\boldsymbol{\xi}).$$

Now we need to compute only $\Psi_0(\boldsymbol{\xi})$ and remember the equation (5) that reads $\Psi_k(\boldsymbol{\xi}) = \prod_{i=1}^{n_o} \psi_{m_i^k}(\xi_i)$ we only need to identify the admissible set of multi indexes $\mathbf{m} \in \mathbb{R}^d$: $|\mathbf{m}| = \sum_{i=1}^{d} m_i \leq n_o$. In this case the only admissible set is $\mathbf{m} = (0,0)$:

$$\Psi_0(\boldsymbol{\xi}) = \psi_0(\xi_1)\psi_0(\xi_2) = 1.$$

The total order expansion of degree zero is: $u(\boldsymbol{\xi}) = \beta_0$.

Total order $n_o = 1$

In this case we need $N_{tot} = 3$. The total order expansion will be

$$u(\boldsymbol{\xi}) = \beta_0 \Psi_0(\boldsymbol{\xi}) + \beta_1 \Psi_1(\boldsymbol{\xi}) + \beta_2 \Psi_2(\boldsymbol{\xi}).$$

We must compute only Ψ_1 and Ψ_2. The admissible set of multi indexes $\mathbf{m}$ is $\{(0,0)(1,0)(0,1)\}$. So we can calculate:

$$\begin{aligned} \Psi_1(\boldsymbol{\xi}) &= \psi_1(\xi_1)\psi_0(\xi_2) = \xi_1 \\ \Psi_2(\boldsymbol{\xi}) &= \psi_0(\xi_1)\psi_1(\xi_2) = \xi_2 \end{aligned}$$

The polynomial truncated expansion is: $u(\boldsymbol{\xi}) = \beta_0 + \beta_1\xi_1 + \beta_2\xi_2$.

Total order $n_o = 2$

The total number of terms is $N_{tot} = 6$. The admissible set of multi indexes $\mathbf{m}$ is $\{(0,0)(1,0)(0,1)(1,1)(2,0)(0,2)\}$. The first three polynomials are already computed, we have to calculate only Ψ_k with $k = 3, \ldots, 5$. So we can write:

$$\begin{aligned} \Psi_3(\boldsymbol{\xi}) &= \psi_1(\xi_1)\psi_1(\xi_2) = \xi_1\xi_2 \\ \Psi_4(\boldsymbol{\xi}) &= \psi_2(\xi_1)\psi_0(\xi_2) = \xi_1^2 - 1 \\ \Psi_5(\boldsymbol{\xi}) &= \psi_0(\xi_1)\psi_2(\xi_2) = \xi_2^2 - 1 \end{aligned}$$

The expansion truncated at the degree $n_o = 2$ is:

$$u(\boldsymbol{\xi}) = \beta_0 + \beta_1\xi_1 + \beta_2\xi_2 + \beta_3(\xi_1\xi_2) + \beta_4(\xi_1^2 - 1) + \beta_5(\xi_2^2 - 1).$$

B Some identities to compute the mean and the variance

In this section we want to show some identities employed to compute the mean (3.2) and the variance (3.2) in the Polynomial Chaos framework.

$$1) \qquad \int_{\Omega_d} \sum_{k=1}^{P} \beta_k \Psi_k(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = 0 \rightarrow \sum_{k=1}^{P} \beta_k \int_{\Omega_d} \Psi_k(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = 0$$

Proof. Remember that $\int_{\Omega^i} \psi_{m_i^k}(\xi_i) p_i(\xi_i) \mathrm{d}\xi_i = 0$ where $\Psi_k(\boldsymbol{\xi}) = \prod_{i=1}^{d} \psi_{m_i^k}(\xi_i)$, $p(\boldsymbol{\xi}) = \prod_{i=1}^{d} p_i(\xi_i)$ and $\Omega_d = \Omega^1 \times \cdots \times \Omega^d$ it follows clearly the previous identity. □

$$2) \qquad \sigma^2(f) = \mu(f^2) - (\mu(f))^2$$

Proof.

$$\begin{aligned}
\sigma^2(f) &= \int (f(x) - \mu(f))^2 p(x) \mathrm{d}x = \int \left(f^2(x) - 2f(x)\mu(f) + \mu^2(f)\right) p(x) \mathrm{d}x \\
&= \int f^2(x) p(x) \mathrm{d}x - 2 \int f(x) \mu(f) p(x) \mathrm{d}x + \int \mu^2(f) p(x) \mathrm{d}x \\
&= \mu(f^2) - 2\mu^2(f) + \mu^2(f) = \mu(f^2) - (\mu(f))^2 .
\end{aligned}$$

□

$$3) \qquad \int_{\Omega_d} \left(\sum_{k=1}^{P} \beta_k \Psi_k(\boldsymbol{\xi}) \right)^2 p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = \sum_{k=1}^{P} \beta_k^2 \langle \Psi_k^2(\boldsymbol{\xi}) \rangle$$

Proof.

$$\begin{aligned}
\int \left(\sum_{k=1}^{P} \beta_k \Psi_k(\boldsymbol{\xi}) \right)^2 p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} &= \sum_{k=1}^{P} \beta_k^2 \int \Psi_k^2(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} + 2 \sum_{k=1}^{P-1} \sum_{j=k+1}^{P} \beta_k \int \Psi_k(\boldsymbol{\xi}) \Psi_j(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} \\
&= \sum_{k=1}^{P} \beta_k^2 \int \Psi_k^2(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \mathrm{d}\boldsymbol{\xi} = \sum_{k=1}^{P} \beta_k^2 \langle \Psi_k^2(\boldsymbol{\xi}) \rangle
\end{aligned}$$

where the orthogonality property between polynomial has been employed. □

C Smolyak's algorithm exemple

In this section we present an application, with all the computations needed, of the Smolyak algorithm for a bidimensional space. We present the calculation of the sparse grid (nodes and weigth) up to second level for a univariate Clenshaw-Curtis quadrature formula.

The univariate Clenshaw-Curtis formula is reported in the table (14).

l	n_l	ξ_i^l	w_i^l
1	1	1/2	1
2	3	0	1/6
		1/2	2/3
		1	1/6

Table 14: Univariate Clenshaw-Curtis formula

$l = 1$

Using the integral in the form (14) in this case we have

$$Q_1^2 f = \sum_{|\mathbf{k}|\leq 2} \Delta_{\mathbf{k}} f, \quad \text{where} \quad \Delta_{\mathbf{k}} f = \left(\Delta_{k_1}^1 \otimes \Delta_{k_2}^1\right) f \tag{28}$$

and $\mathbf{k} \in \mathbb{N}^2$. Now we have to select the admissible set of $\mathbf{k}$ remember that $|\mathbf{k}| = \sum_{i=1}^2 k_i$. We have the admissible set:

$$\mathbf{k} = \{(0,0)(0,1)(1,0)(0,2)(2,0)(1,1)\},$$

but remember that $\Delta_0^1 f = 0$, we can eliminate all the set in which at least one of the terms is zero. In this way the admissible set reduces to

$$\mathbf{k} = \{(1,1)\}.$$

The equation (28) becomes

$$Q_1^2 f = \Delta_{(1,1)} f = (\Delta_1^1 \otimes \Delta_1^1) f. \tag{29}$$

Remember the equations (12 and 11) it is possible to write

$$\Delta_1^1 f = Q_1^1 f - Q_0^1 f = Q_1^1 f = w_1^1 f(\xi_1). \tag{30}$$

We can now rewrite the equation (29) and obtain the final formula

$$Q_1^2 f = w_1^1 f(\xi_1^1) \otimes w_1^1 f(\xi_1^1) = w_1^1 w_1^1 f(\xi_1^1, \xi_1^1) = 1 \cdot f\left(\frac{1}{2}, \frac{1}{2}\right)$$

and we can derive the sparse grid for the level one based on a Clenshaw-Curtis quadrature formula, in the probabilistic reference space, of coordinates $(1/2, 1/2)$ with unitary weight. In the figure (26) this grid is reported.

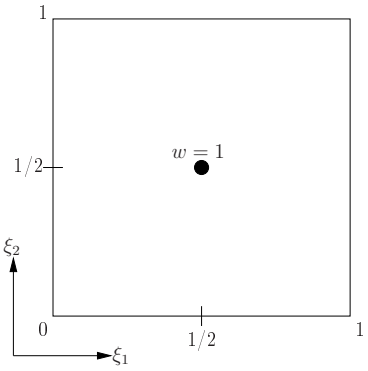

Figure 26: Sparse grid level 1.

$l = 2$

In the case of level 2 the equation (14) becomes

$$Q_2^2 f = \sum_{|\mathbf{k}| \leq 3} \Delta_{\mathbf{k}} f. \tag{31}$$

We have to add at the previous set for the level one this set:

$$\mathbf{k} = \{(1,2)(2,1)(3,0)(0,3)\},$$

and if we exclude the element in which at least one component is zero we obtain:

$$\mathbf{k} = \{(1,2)(2,1)\}. \tag{32}$$

Of course this set need to be added to the previous one (of the level one).

The approximation for the integral that we have to compute can be write as

$$Q_2^2 f = \Delta_{(1,1)} f + \Delta_{(1,2)} f + \Delta_{(2,1)} f,$$

in which we have already computed the term $\Delta_{(1,1)}$. We have to calculate the other two terms of the previous equation.

Remember the equation (13) we can write

$$\begin{aligned} \Delta_{(1,2)} f &= (\Delta_1^1 \otimes \Delta_2^1) f \\ \Delta_{(2,1)} f &= (\Delta_2^1 \otimes \Delta_1^1) f \end{aligned} \tag{33}$$

so we have to compute only the term Δ_2^1 (the term Δ_1^1 is already computed for the level one case). Using the equation (12)

$$\Delta_2^1 f = (Q_2^1 - Q_1^1) f,$$

in which we can write the term $Q_2^1 f$ by the equation (11)

$$Q_2^1 f = \sum_{i=1}^{3} w_i^2 f(\xi_i^2) = w_1^2 f(\xi_1^2) + w_2^2 f(\xi_2^2) + w_3^2 f(\xi_3^2).$$

We can now write 33 as

$$\begin{aligned}\Delta_{(1,2)}f &= w_1^1 f(\xi_1^1) \otimes \left(w_1^2 f(\xi_1^2) + w_2^2 f(\xi_2^2) + w_3^2 f(\xi_3^2) - w_1^1 f(\xi_1^1)\right)\\ \Delta_{(2,1)}f &= \left(w_1^2 f(\xi_1^2) + w_2^2 f(\xi_2^2) + w_3^2 f(\xi_3^2) - w_1^1 f(\xi_1^1)\right) \otimes w_1^1 f(\xi_1^1).\end{aligned}$$

The final approximation for the integral can be write as

$$\begin{aligned}Q_2^2 f = {} & w_1^1 w_1^1 f(\xi_1^1, \xi_1^1) + w_1^1 w_1^2 f(\xi_1^1, \xi_1^2) + w_1^1 w_2^2 f(\xi_1^1, \xi_2^2) + w_1^1 w_3^2 f(\xi_1^1, \xi_3^2)\\ & - w_1^1 w_1^1 f(\xi_1^1, \xi_1^1) + w_1^2 w_1^1 f(\xi_1^2, \xi_1^1) + w_2^2 w_1^1 f(\xi_2^2, \xi_1^1) + w_3^2 w_1^1 f(\xi_3^2, \xi_1^1) - w_1^1 w_1^1 f(\xi_1^1, \xi_1^1).\end{aligned}$$

If the univariate quadrature formula is the Clenshaw-Curtiss ones which nodes and weights are reported in the table (14) the corresponding sparse grid can be obtained in the bidimensional space

$$\begin{aligned}Q_2^2 f = {} & \frac{1}{6} f\left(\frac{1}{2}, 0\right) + \frac{2}{3} f\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{1}{6} f\left(\frac{1}{2}, 1\right)\\ & + \frac{1}{6} f\left(0, \frac{1}{2}\right) + \frac{2}{3} f\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{1}{6} f\left(1, \frac{1}{2}\right) - 1 f\left(\frac{1}{2}, \frac{1}{2}\right)\\ = {} & \frac{1}{6} f\left(\frac{1}{2}, 0\right) + \frac{1}{3} f\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{1}{6} f\left(\frac{1}{2}, 1\right) + \frac{1}{6} f\left(0, \frac{1}{2}\right) + \frac{1}{6} f\left(1, \frac{1}{2}\right).\end{aligned}$$

We resumed the nodes and the weights of the sparse grid for the second level in the bidimensional case in the table (15) and in the figure (27) the grid is shown.

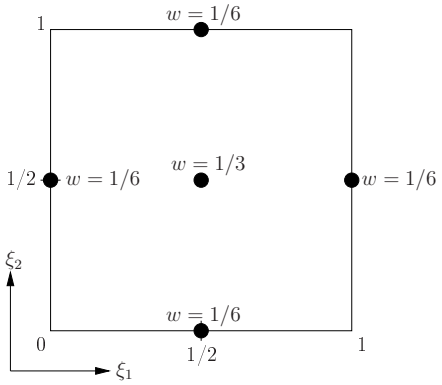

Figure 27: Sparse grid level 2.

ξ_1	ξ_2	w_i^2
1/2	0	1/6
1/2	1/2	1/3
1/2	1	1/6
0	1/2	1/6
1	1/2	1/6

Table 15: Second level bidimensional sparse grid (Cleanshaw-Curtiss)

As example we report in the figure 28 the evolution of the number of the simulations N with respect to the stochastic dimension d for a full tensorization of total degree $n_o = 3$, and for the Sparse Grid employing a Clenshaw-Curtis and a trapezoidal rule with a level $l = n_o+1$. The number of simulation required with the Petras routine is also shown.

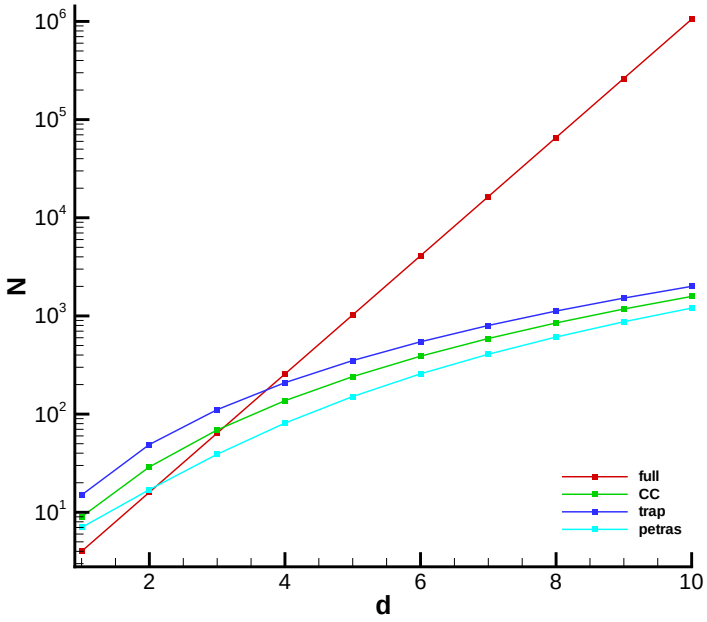

Figure 28: Total number of simulation with respect to the stochastic dimension

D Analitical expression of polynomials

Here the complete list of polynomials employed in this work is reported:

Legendre

$$P_n(x) = \frac{1}{2^n n!} \frac{\mathrm{d}^n}{\mathrm{d}x^n} \left[(x^2 - 1)^n \right]$$

Hermite

$$H_n(x) = (-1)^n e^{x^2/2} \frac{\mathrm{d}^n}{\mathrm{d}x^n} \left(e^{-x^2/2} \right)$$

Chebyshev (I kind)

They are defined by succession

$$\begin{cases} T_0(x) & = 1 \\ T_1(x) & = x \\ T_{n+1} & = 2xT_n(x) - T_{n-1}(x) \end{cases}$$

E Geometry description of the nozzle

We report here the equations used for the geometrical description of the nozzle. The nozzle employed is a de Laval ones. In both case, five or ten uncertainties, the description of the convergent part is the same; the divergent part is different according to the indications given in the following. In the figure (29) the mean profile of the nozzle is reported. Note that the reference geometry is the same for the two case.

Five uncertainties

$$\begin{aligned} y_c &= 1 + 0.75x^2 \\ y_d &= 1 + \alpha x + \beta x^2 + \gamma_d x^3 \quad \text{with} \quad \gamma_d = A_e/A_t - 1 - \alpha - \beta \end{aligned}$$

Ten uncertainties

$$\begin{aligned} y_c &= 1 + 0.75x^2 \\ y_d &= 1 + \alpha x + \beta x^2 + \alpha_1 x^3 + \alpha_2 x^4 + \alpha_3 x^5 + \alpha_4 x^6 + \alpha_5 x^7 + \gamma_d x^8 \quad \text{with} \\ \gamma_d &= A_e/A_t - 1 - \alpha - \beta - \alpha_1 - \alpha_2 - \alpha_3 - \alpha_4 - \alpha_5 \end{aligned}$$

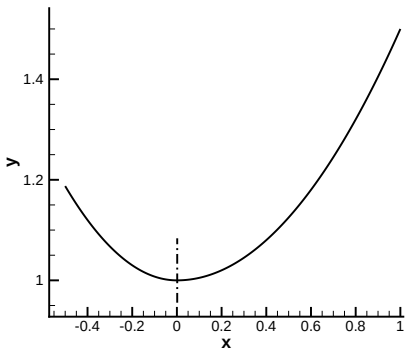

Figure 29: Mean geometry of the nozzle

Centre de recherche INRIA Bordeaux – Sud Ouest
Domaine Universitaire - 351, cours de la Libération - 33405 Talence Cedex (France)

Centre de recherche INRIA Grenoble – Rhône-Alpes : 655, avenue de l'Europe - 38334 Montbonnot Saint-Ismier
Centre de recherche INRIA Lille – Nord Europe : Parc Scientifique de la Haute Borne - 40, avenue Halley - 59650 Villeneuve d'Ascq
Centre de recherche INRIA Nancy – Grand Est : LORIA, Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - BP 101 - 54602 Villers-lès-Nancy Cedex
Centre de recherche INRIA Paris – Rocquencourt : Domaine de Voluceau - Rocquencourt - BP 105 - 78153 Le Chesnay Cedex
Centre de recherche INRIA Rennes – Bretagne Atlantique : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex
Centre de recherche INRIA Saclay – Île-de-France : Parc Orsay Université - ZAC des Vignes : 4, rue Jacques Monod - 91893 Orsay Cedex
Centre de recherche INRIA Sophia Antipolis – Méditerranée : 2004, route des Lucioles - BP 93 - 06902 Sophia Antipolis Cedex

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, BP 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399