

Improved Non-Intrusive Uncertainty Propagation in Complex Fluid Flow Problems

L.M.M. van den Bos^(*),^{1,2,3}, B. Koren^{1,2}, R.P. Dwight³

¹ Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, the Netherlands

² Centrum Wiskunde & Informatica, P.O. Box 94079, 1090 GB Amsterdam, the Netherlands

³ Delft University of Technology, P.O. Box 5, 2600 AA Delft, the Netherlands

(*) Corresponding author: l.m.m.van.den.bos@cwi.nl

Abstract

The problem of non-intrusive uncertainty quantification is studied, with a focus on two computational fluid dynamics cases. A collocation method using quadrature or cubature rules is applied, where the simulations are selected deterministically. A one-dimensional quadrature rule is proposed which is nested, symmetric, and has positive weights. The rule is based on the removal of nodes from an existing symmetric quadrature rule with positive weights. The set of rules can be used to generate high-dimensional sparse grids using a Smolyak procedure, but such a procedure introduces negative weights. Therefore a new cubature rule is generated, also based on the removal of nodes. Again the rules are symmetric, positive, and nested. In low-dimensional cases, the number of nodes is approximately equal to the number of nodes of a sparse grid. If weight-positivity is dropped, it also has less nodes in high-dimensional cases. Moreover a method is proposed to determine the convergence criterion for each individual node. Because the weights for each node differ, varying the convergence criterion for each node results in less computational time without changing the quadrature or cubature rule. Two CFD cases are studied that show the properties of the proposed methods.

Keywords: Uncertainty Propagation, Robust Simulation, Computational Fluid Dynamics

1 Introduction

The problem of quantifying uncertainty in computational fluid dynamics (CFD) is considered. Let d uncertain parameters be given, including their probability density functions (PDFs). The goal is to obtain accurate statistics on the outputs of interest of the model, without adapting the model itself (i.e. non-intrusively) and using as few evaluations of the model as possible.

The conventional method is Monte Carlo (MC), where the convergence rate of the statistics is of the order $1/\sqrt{N}$ (where N is the number of samples). For sufficiently low dimensions and smooth functions, this can be improved significantly by applying quasi Monte Carlo methods [1], with a convergence rate of $\mathcal{O}((\log N)^d/N)$. For $d \lesssim 10$ this can be further improved by polynomial approximation of the quantity of interest. This approach is used in this paper.

Stochastic Collocation (SC) [3, 11] is a method employing the polynomial approximation, using tensor products or sparse grids in high-dimensional parameter spaces. Piecewise polynomial interpolation on unstructured MC grids has also been studied [21, 22]. If the quantity of interest is sufficiently smooth, in all cases spectral convergence is obtained. Using quadrature rules, statistics can be estimated.

In high-dimensional spaces, the canonical method to construct grids is the tensor product, although this introduces large numbers of nodes. Sparse grids, e.g. the Smolyak Sparse grid [14, 15, 19], use less nodes while retaining polynomial accuracy. The latter technique has been studied thoroughly and extended by various authors [5, 6, 12, 13, 17].

To construct a Smolyak sparse grid, a set of nested one-dimensional quadrature rules is required for the optimal result. No general strategy exists yet to generate nested quadrature rules for general distributions.

In this paper, a method is proposed to generate quadrature rules with existing weights using an existing quadrature rule. If the target distribution is symmetric, this property can be carried over to

the set of quadrature rules. The set of rules is perfectly suited for a Smolyak procedure. If the Smolyak sparse grid still contains too many nodes due to time restrictions, the same principles can be used in high-dimensional spaces to obtain a cubature rule with less nodes than the Smolyak sparse grid.

A cubature rule uses multiple individual evaluations of the model to estimate the statistics. In CFD cases such models are often complex systems of PDEs which are solved numerically. The numerical method estimates the solution with some discretization error. It is customary to use the same convergence criterion for each simulation, but this can be improved significantly by investigating and optimizing the computational cost that is necessary for the evaluation of the cubature rule. A method is proposed to optimize this cost, without any modifications to the cubature rule.

This paper is set up as follows. Firstly, in the next section the problem is described including existing techniques such as the Smolyak procedure and quadrature rules. In Section 3 the procedure to construct nested quadrature rules is introduced, which can also easily be applied to a multi-dimensional setting. A method to optimize convergence criteria for each node is proposed in Section 4. Two numerical examples are discussed in Section 5. Firstly, the rules are applied to the standard lid-driven cavity flow problem computed through a Lattice Boltzmann method. Secondly, the multi-dimensional rule is applied to a three-dimensional airplane aerodynamics problem using the Euler equations of gas dynamics, and considering seven uncertain parameters. In the latter case, each individual model run requires much computational time such that using the smallest number of nodes possible is highly relevant.

2 Uncertainty Quantification - basic principles

2.1 Problem description

Consider the following computational problem in domain $D \subset \mathbb{R}^n$, with $n = 1, 2, 3$, and with solution v :

$$\mathcal{A}(\mathbf{x}, t; v(\boldsymbol{\xi}(\boldsymbol{\omega}))) = \mathcal{S}(\mathbf{x}, t, \boldsymbol{\xi}(\boldsymbol{\omega})),$$

completed with initial and boundary conditions. Here, \mathcal{A} (defined on $D \times \mathbb{R}_+ \times \mathbb{R}$) and \mathcal{S} (defined on $D \times \mathbb{R}_+ \times \Xi$) are the differential and source operator respectively and typically occur in the discretization of a continuous PDE. $\mathbf{x} \in D$ and $t \in \mathbb{R}_+$ are the spatial and temporal parameter respectively. The parameters $\boldsymbol{\xi} : \Omega \mapsto \Xi$ are d square-integrable random variables, which are assumed to be independent, with respect to the probability space (Ω, \mathcal{F}, P) , with $\Xi \subset \mathbb{R}^d$, $\Omega \subset \mathbb{R}^d$, and P the probability measure. $\boldsymbol{\omega}$ is omitted from the notation for sake of clarity. The function $v : \Xi \mapsto D \times \mathbb{R}_+$ describes the solution depending on the parameters $\boldsymbol{\xi}$. Random fields, which are infinite dimensional, can be used after the application of a truncated Karhunen-Loève expansion [7, 24].

The problem is to determine the probability density function and statistical moments of u , with $u(\boldsymbol{\xi}) = s(v(\boldsymbol{\xi}))$. u is a single quantity derived from v through function $s : D \times \mathbb{R}_+ \mapsto \mathbb{R}$. The focus is on determining the statistical moments

$$\mathbb{E}[u^l(\boldsymbol{\xi})] := \int_{\Xi} u^l(\boldsymbol{\xi}) p(\boldsymbol{\xi}) \, d\boldsymbol{\xi}, \text{ for } l = 1, 2, \dots, \quad (2.1)$$

where $p(\boldsymbol{\xi})$ is the probability density function (PDF) of $\boldsymbol{\xi}$.

2.2 Quadrature and cubature rules

The current approach is to approximate the integral of Equation (2.1) using a weighted cubature rule, i.e. using

$$\mathbb{E}[u^l(\boldsymbol{\xi})] \simeq \sum_{k=1}^N u^l(\boldsymbol{\xi}_k) w_k, \text{ for } l = 1, 2, \dots,$$

where $\{\boldsymbol{\xi}_k\}_{k=1, \dots, N} \subset \Xi$ is a finite number of samples with cubature weights $\{w_k\}_{k=1, \dots, N} \subset \mathbb{R}$. $u^l(\boldsymbol{\xi}_k) = (u(\boldsymbol{\xi}_k))^l$ is determined by solving the discrete problem

$$\mathcal{A}(\mathbf{x}, t; v(\boldsymbol{\xi}_k)) = \mathcal{S}(\mathbf{x}, t, \boldsymbol{\xi}_k)$$

for $v(\boldsymbol{\xi}_k)$, and by evaluating $u(\boldsymbol{\xi}_k)$. We call this a quadrature rule if Ξ is one-dimensional and a cubature rule otherwise. All properties of a cubature rule are also properties of a quadrature rule, but not vice versa.

The degree of a cubature rule is a measure for the accuracy. The degree is defined as the largest number K such that all polynomials of degree less or equal than K are integrated exactly.

Relevant properties for quadrature and cubature rules are nesting, positive weights, and symmetry. An ordered set of cubature rules is nested if the nodes of a smaller cubature rule are also nodes of a larger cubature rule. A cubature rule is called positive if all weights are strictly positive. A cubature rule is symmetric if it has the same symmetry as the PDF under consideration.

The Gauss quadrature rule [10] is symmetric if the PDF is symmetric and positive, irrespective of the PDF. However, sets of Gauss quadrature rules are not nested. The Clenshaw–Curtis quadrature rule [2] is a symmetric and nested quadrature rule, but does not guarantee positive weights if a non-uniform distribution is used.

2.3 Vandermonde-matrix

If N nodes $\{\xi_k\}$ of a quadrature rule are given, the weights can be determined by solving the system:

$$\sum_{k=1}^N \xi_k^j w_k = \int_{\Xi} \xi^j p(\xi) d\xi, \text{ for } j = 0, 1, \dots, N-1. \quad (2.2)$$

This is a linear system; the matrix of the system is a *Vandermonde-matrix* V , defined by $V_{j,k} = \xi_k^j$. Matrix V is non-singular if all nodes are distinct.

If N nodes $\{\xi_k\}$ of a cubature rule are given, a similar system can be constructed:

$$\sum_{k=1}^N m_j(\xi_k) w_k = \int_{\Xi} m_j(\xi) p(\xi) d\xi, \text{ for } j = 1, 2, \dots, N.$$

Here, m_j is the j^{th} monomial under some ordering. We call the matrix of this system the *generalized Vandermonde-matrix* G , defined by $G_{j,k} = m_j(\xi_k)$. In contrast to the one-dimensional case, this matrix is not non-singular if all nodes are distinct. For tensor product rules, the matrix is non-singular as it can be formed by the Kronecker product of the Vandermonde-matrices of the individual quadrature rules.

2.4 Smolyak cubature rule

The Smolyak procedure [14, 15, 19] is a method to construct a “sparse” cubature rule from a family of quadrature rules. It relies on the combination of several tensor product grids, which are generated using the set of quadrature rules. Best results are obtained if the quadrature rules are nested, reducing the total number of nodes.

The Smolyak sparse grid can be generated using the following explicit expression [20]:

$$S_k = \sum_{\substack{K-d+1 \leq \|\alpha\|_1 \leq K \\ \alpha \in \mathbb{N}^d}} (-1)^{K-\|\alpha\|_1} \binom{d-1}{K-\|\alpha\|_1} \bigotimes_{k=1}^d Q_{N_k},$$

where $\{N_k\}_{k=1}^N \subset \mathbb{N}$ is an increasing sequence and Q_{N_k} is an N_k -node quadrature rule. $\{N_k\}$ typically grows exponentially. The Smolyak cubature rule has a high degree then: at least $2(K-d)+1$. We only consider Smolyak cubature rules generated by sequences with exponential growth.

As the Smolyak cubature rule combines smaller tensor cubature rules, results are best if the tensor grids (hence the quadrature rules) are nested. This can be clearly seen in Figure 1. For nested quadrature rules, e.g. the Clenshaw–Curtis quadrature rule, a Smolyak grid of degree 9 consists of 65 nodes (see Figure 1c). However, the Clenshaw–Curtis quadrature rule does not depend on the underlying distribution. The non-nested Gaussian quadrature rule (Figures 1a and 1b) yields much more nodes, but does take the distribution into account.

3 Reduction of quadrature rules

In this section a set of quadrature rules is constructed which is nested, symmetric, and positive. The procedure is based on the removal of nodes from an existing quadrature rule. Symmetry and positivity of the existing quadrature rule are maintained by the procedure. We call the new rules *reduced quadrature rules* and the step to construct these the *reduction step*. These rules are not unique, which can be entailed to use prior information about the integrand.

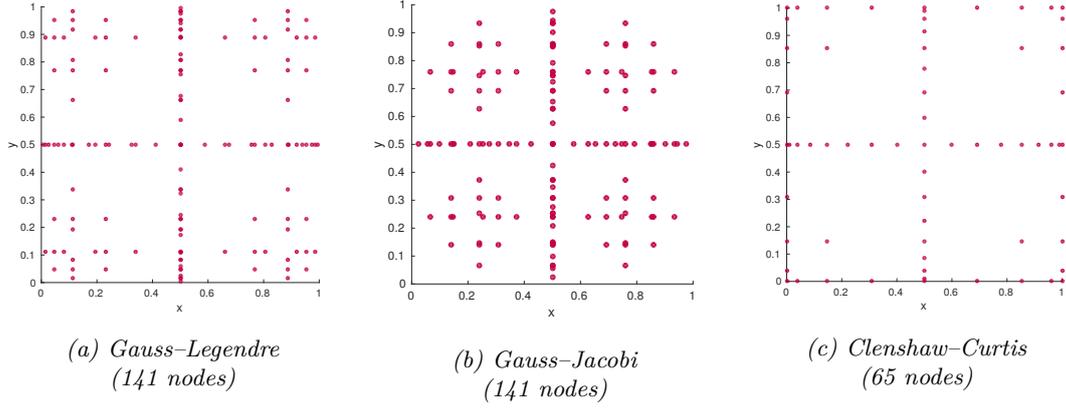


Figure 1: Two-dimensional Smolyak cubature rule nodes of several quadrature rules.

3.1 The reduction step

The principle of the reduction step is to remove nodes from a quadrature rule. Suppose a quadrature rule of N nodes and degree $N - 1$ is given. From Equation (2.2) it is known that

$$\sum_{k=1}^N \xi_k^j w_k = \int_{\Xi} \xi^j p(\xi) d\xi, \text{ for } j = 0, 1, \dots, N - 1.$$

After the removal of a node, we demand that the resulting $N - 1$ nodes form a quadrature rule of degree $N - 2$. So if node ξ_{k_0} is removed, this results in the system

$$\sum_{\substack{k=1, \dots, N \\ k \neq k_0}} \xi_k^j w_k^* = \int_{\Xi} \xi^j p(\xi) d\xi, \text{ for } j = 0, 1, \dots, N - 2.$$

There always exists a k_0 such that the weights of this system (denoted by $\{w^*\}$) are positive. Such k_0 can be constructed as follows.

1. Let the quadrature rule nodes $\{\xi_k\}$ and weights $\{w_k\}$ be given.
2. Consider the matrix W of the system (note the $N - 2$):

$$\sum_{k=1}^N \xi_k^j w_k = \int_{\Xi} \xi^j p(\xi) d\xi, \text{ for } j = 0, 1, \dots, N - 2.$$

3. Matrix W is an $(N - 1) \times N$ -matrix, so it has a non-trivial null vector $\mathbf{c} = (c_1, \dots, c_N)^T$.
4. Because $W\mathbf{c} = \mathbf{0}$, it is true that for all $\alpha \in \mathbb{R}$

$$\sum_{k=1}^N \xi_k^j (w_k - \alpha c_k) = \int_{\Xi} \xi^j p(\xi) d\xi, \text{ for } j = 0, 1, \dots, N - 2.$$

5. Choose α such that $\alpha = \min \left\{ \frac{w_k}{c_k} : c_k > 0 \right\}$. Choose k_0 such that $\frac{w_{k_0}}{c_{k_0}} = \alpha$.
6. Now it follows that $w_{k_0} - \alpha c_{k_0} = 0$ and for all $k = 1, \dots, N$ we have $w_k - \alpha c_k \geq 0$, so picking new weights $w_k^* = w_k - \alpha c_k$ for $k = 1, \dots, N$ allows the removal of node k_0 .

These steps form the reduction step. At least one $c_k > 0$ and at least one $c_k < 0$, because $\sum_{k=1}^N c_k = 0$. Therefore k_0 is not uniquely defined – this will be discussed later.

The resulting set of quadrature rules suit perfectly for a Smolyak procedure, as they are nested and positive. Therefore, independently of the distribution the resulting Smolyak grids consists of 65 nodes (see Figure 2).

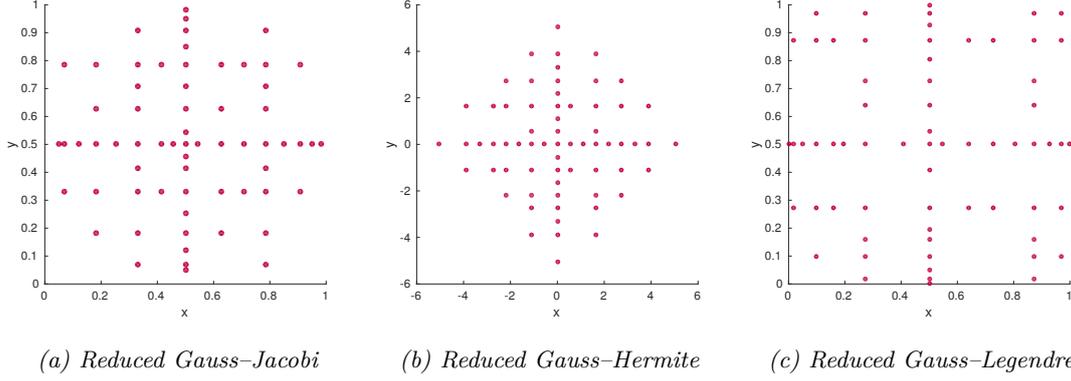


Figure 2: Two-dimensional Smolyak cubature rule nodes of the reduced quadrature rule. All grids consist of 65 nodes.

3.2 Symmetry

The reduction step as stated here does not guarantee symmetry of the nested quadrature rule if the original quadrature rule is nested. It is however easy to adapt it such that it does. Without loss of generality, consider a quadrature rule that is symmetric around 0. Then for N even

$$\sum_{k=1}^N \xi_k^j w_k = \sum_{k=1}^{N/2} \xi_k^j + \sum_{k=N/2+1}^N (-\xi_k)^j,$$

which is 0 if j is odd. If N is odd,

$$\sum_{k=1}^N \xi_k^j w_k = \sum_{k=1}^{\lfloor N/2 \rfloor} \xi_k^j + \sum_{k=\lceil N/2+1 \rceil}^N (-\xi_k)^k + 0^j,$$

which also equals 0 if j is odd. Therefore, it is always true for odd j that

$$\sum_{k=1}^N \xi_k^j w_k = \int_{\Xi} \xi^j p(\xi) d\xi,$$

independently of the degree of the quadrature rule. This can be used to keep the quadrature rule symmetric by constructing k_0 as follows:

1. Let the quadrature rule nodes $\{\xi_k\}$ and weights $\{w_k\}$ be given. Assume the quadrature rule is symmetric around 0.
2. If N is even, consider the matrix W^* of the system:

$$2 \sum_{k=1}^{N/2} \xi_k^j w_k = \int_{\Xi} \xi^j p(\xi) d\xi, \text{ for } j = 0, 2, \dots, N-2.$$

3. If N is odd, consider the matrix W^* of the system:

$$\xi_{\lceil N/2 \rceil}^j w_{\lceil N/2 \rceil} + 2 \sum_{k=1}^{\lfloor N/2 \rfloor} \xi_k^j w_k = \int_{\Xi} \xi^j p(\xi) d\xi, \text{ for } j = 0, 2, \dots, N-3.$$

4. In either case, W^* is non-square and has a non-trivial null vector \mathbf{c}^* .
5. If N is even, let $\mathbf{c} = (c_1^*, c_2^*, \dots, c_{N/2}^*, c_{N/2}^*, \dots, c_1^*)^T$.
6. If N is odd, let $\mathbf{c} = (c_1^*, c_2^*, \dots, c_{\lfloor N/2 \rfloor}^*, c_{\lceil N/2 \rceil}^*, c_{\lceil N/2 \rceil}^*, \dots, c_1^*)^T$.
7. Use \mathbf{c} as null vector in the *original* reduction step. Note that \mathbf{c} is null vector of W .

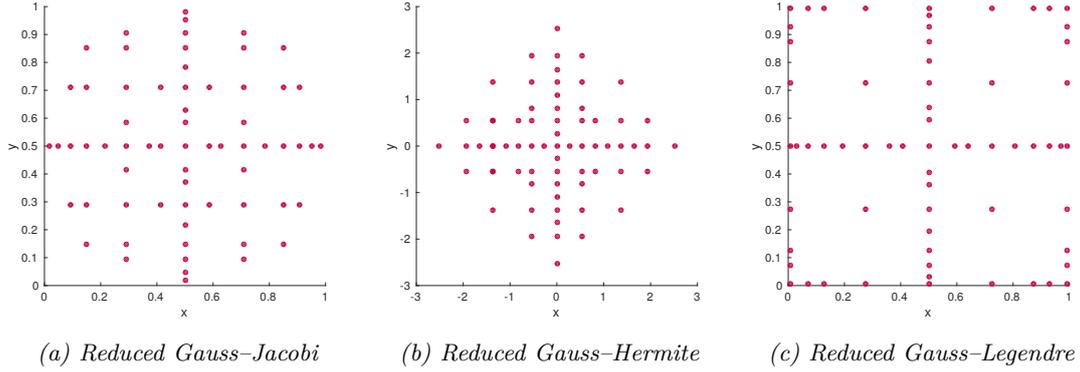


Figure 3: Two-dimensional Smolyak cubature rule nodes of the symmetric reduced quadrature rule. All sparse grids consist of 65 nodes.

If N is odd, it is possible that the middle node is removed by \mathbf{c} . This can be overcome by multiplying the vector with -1 . In all other cases this construction is not unique. Multiplying the null vector with -1 then yields a valid null vector which can be used to remove different nodes.

The set of quadrature rules generated with this procedure is symmetric, nested, and positive. The resulting Smolyak grid is therefore also symmetric (see Figure 3).

3.3 Generalization to cubature rules

The reduction step can be easily generalized to a multi-dimensional setting. The steps from Section 3.1 can be applied straightforwardly.

Incorporating symmetries is more cumbersome, as in a multi-dimensional setting different types of symmetry exist. We consider two types of symmetry: *rotational* and *reflectional* symmetry. Without loss of generality, we assume that the cubature rule is two-dimensional and let $\boldsymbol{\xi} = (x, y)$.

If the cubature rule is symmetric under rotations of 45° along an axis, it is rotationally symmetric. The plane of symmetry is the line $y = x$. In this case, we have that

$$\forall j, l : \sum_{k=1}^N x_k^j y_k^l w_k = \int_{\Xi} x^j y^l p(\boldsymbol{\xi}) d\boldsymbol{\xi} \implies \sum_{k=1}^N x_k^l y_k^j w_k = \int_{\Xi} x^l y^j p(\boldsymbol{\xi}) d\boldsymbol{\xi}.$$

Using this, a null vector can be generated that retains the symmetry after the removal of a node in exactly the same way as in the one-dimensional case.

The second symmetry (*reflectional* symmetry) means that the cubature rule is symmetric along one dimension, e.g. the cubature is symmetric with plane of symmetry equal to $x = 0$. We have a similar situation as the symmetry discussed in one-dimension, i.e., we have for j odd

$$\forall l : \sum_{k=1}^N x_k^j y_k^l w_k = 0 = \int_{\Xi} x^j y^l p(\boldsymbol{\xi}) d\boldsymbol{\xi}.$$

The symmetry can be incorporated in a similar way as in the one-dimensional case.

In the multi-dimensional case, there are two possible reductions of a cubature rule. Firstly, it is possible to keep the weights positive, which is the situation studied in the one-dimensional case. Secondly, if we allow the weights to become negative, it is possible to remove up to 2^d nodes in one reduction step because of the multi-dimensional symmetries. If multiple types of symmetries are considered, this number can become larger (see Table 1 for an overview).

3.4 Selection criteria

The reduction step is not unique, which has been discussed previously. Using knowledge about the integrand it is possible to tailor the quadrature rule to the specific application, e.g. by keeping those nodes where the integrand is highly non-linear. If no information about the integrand is available on beforehand, one of the following node selection criteria can be used:

d	K	N_{Smolyak}	N_{Positive}	N_{Negative}
5	5	61	113	43
5	7	241	544	384
5	9	805	1313	325
5	11	2473	4096	2016
5	13	7245	6005	1607
7	5	113	689	99
7	7	589	1797	325
7	9	2471	19717	901
7	11	9101	28479	2863
7	13	30907	158709	28479
10	5	221	13461	201
10	7	1581	20533	1361
10	9	8810	1368449	3705
10	11	41445	8284617	12489

Table 1: Number of nodes of several cubature rules for several dimensions (d) and several degrees (K). N_{Positive} denotes the number of nodes of the symmetric reduced cubature rule with positive weights, N_{Negative} denotes the number of nodes of the negative symmetric reduced cubature rule, and N_{Smolyak} denotes the number of nodes of a Smolyak sparse grid.

1. Quadrature nodes can be used to interpolate the unknown function. The error of this interpolation procedure can be estimated easily [9]. A criterion selecting the subset of nodes such that this error is as small as possible defines the reduction step uniquely.
2. Using the underlying PDF of the quadrature rule, it is possible to determine the likelihood of the nodes of a quadrature rule by evaluating the PDF at each node and taking the product of the result. A higher likelihood is desirable, because the nodes then form a good sample from the distribution. A criterion selecting the subset of nodes such that the likelihood is optimal defines the reduction step uniquely if all nodes have a different value of the PDF.
3. The reduction step creates a nested cubature rule with positive weights, but does not keep the order of the weights equal, i.e., it is possible that one weight becomes several orders of magnitude smaller than another weight. A criterion selecting the subset of nodes such that the maximum weight of the resulting cubature rule is as small as possible defines the reduction step uniquely.

4 Distribution of error among nodes

In CFD, determining the output of a model given initial and boundary conditions often requires complex numerical methods and often is computationally expensive. As before, let $u(\boldsymbol{\xi})$ be the function of interest. Given a cubature node $\boldsymbol{\xi}_k$, we assume a numerical code can determine $\hat{u}(\boldsymbol{\xi}_k) \approx u(\boldsymbol{\xi}_k)$ such that

$$\|u(\boldsymbol{\xi}_k) - \hat{u}(\boldsymbol{\xi}_k)\| \leq \varepsilon,$$

for any $\varepsilon > 0$ (typically ε is the discretization error). In this section, the optimal strategy for choosing ε for each node is discussed by considering this as an optimization problem.

Let $\{\boldsymbol{\xi}_k\}$ and $\{w_k\}$ be cubature rule nodes and weights respectively. Let \mathcal{Q} be the cubature rule operator. We want to determine $\{\varepsilon_k\}$ such that if $\|u(\boldsymbol{\xi}_k) - \hat{u}(\boldsymbol{\xi}_k)\| < \varepsilon_k$, then $\|\mathcal{Q}(\hat{u}) - \mathcal{Q}(u)\| < \mathcal{E}$, where \mathcal{E} is given beforehand.

4.1 Cost minimization

Let $C(\varepsilon; \boldsymbol{\xi})$ be a function proportional to the computational cost necessary to determine $\hat{u}(\boldsymbol{\xi})$ with error ε . We assume C is smooth, positive, and decreasing, i.e., determining a more accurate solution requires more computational cost. Then the cost of determining $\mathcal{Q}(u)$ equals $\sum_{k=1}^N C(\varepsilon_k; \boldsymbol{\xi}_k)$. We want to minimize this cost under the condition $\|\mathcal{Q}(\hat{u}) - \mathcal{Q}(u)\| \leq \mathcal{E}$, i.e. we need to solve

$$\arg \min_{\varepsilon_1, \dots, \varepsilon_N} \left\{ \sum_{k=1}^N C(\varepsilon_k; \boldsymbol{\xi}_k) \mid \|\mathcal{Q}(\hat{u}) - \mathcal{Q}(u)\| \leq \mathcal{E} \right\}.$$

It is cumbersome to determine $\|\mathcal{Q}(\hat{u}) - \mathcal{Q}(u)\|$ accurately, so we use the following estimate:

$$\begin{aligned}\|\mathcal{Q}(\hat{u}) - \mathcal{Q}(u)\| &= \|\mathcal{Q}(\hat{u} - u)\| \\ &= \left\| \sum_{k=1}^N w_k (\hat{u}(\boldsymbol{\xi}_k) - u(\boldsymbol{\xi}_k)) \right\| \\ &\leq \sum_{k=1}^N |w_k| \|\hat{u}(\boldsymbol{\xi}_k) - u(\boldsymbol{\xi}_k)\| \\ &= \sum_{k=1}^N |w_k| \varepsilon_k.\end{aligned}$$

The goal is therefore to solve the following minimization problem:

$$\arg \min_{\varepsilon_1, \dots, \varepsilon_N} \left\{ \sum_{k=1}^N C(\varepsilon_k; \boldsymbol{\xi}_k) \mid \sum_{k=1}^N |w_k| \varepsilon_k \leq \mathcal{E} \right\}. \quad (4.1)$$

A feasible solution of this problem is choosing $\varepsilon_k = \mathcal{E} / (\sum_{k=1}^N |w_k|)$ for all k . Solving the minimization problem consists of two steps: (i) determining the optimal solution under condition $\sum_{k=1}^N |w_k| \varepsilon_k = \mathcal{E}$ and (ii) proving that such a solution is an optimal solution to the minimization problem of Equation (4.1).

The latter is easy to see. Let $\{\varepsilon_k\}$ be a solution to the minimization problem:

$$\arg \min_{\varepsilon_1, \dots, \varepsilon_N} \left\{ \sum_{k=1}^N C(\varepsilon_k; \boldsymbol{\xi}_k) \mid \sum_{k=1}^N |w_k| \varepsilon_k < \mathcal{E} \right\}.$$

Then increase ε_1 such that $\sum_{k=1}^N |w_k| \varepsilon_k = \mathcal{E}$ and a better solution is obtained (because the cost decreases), i.e., an optimal solution of Equation (4.1) always has $\sum_{k=1}^N |w_k| \varepsilon_k = \mathcal{E}$.

The minimization problem

$$\arg \min_{\varepsilon_1, \dots, \varepsilon_N} \left\{ \sum_{k=1}^N C(\varepsilon_k; \boldsymbol{\xi}_k) \mid \sum_{k=1}^N |w_k| \varepsilon_k = \mathcal{E} \right\}$$

can easily be solved using the Lagrange multiplier method. Let $f(\varepsilon_1, \dots, \varepsilon_N) = -\sum_{k=1}^N C(\varepsilon_k; \boldsymbol{\xi}_k)$ and $g(\varepsilon_1, \dots, \varepsilon_N) = \sum_{k=1}^N |w_k| \varepsilon_k - \mathcal{E}$. Then an optimal solution to the minimization problem above reads:

$$\begin{cases} g(\varepsilon_1, \dots, \varepsilon_N) = 0, \\ \nabla f(\varepsilon_1, \dots, \varepsilon_N) - \lambda \nabla g(\varepsilon_1, \dots, \varepsilon_N) = \mathbf{0}. \end{cases}$$

The solution of this system reads

$$\varepsilon_k = (\mathrm{d}_\varepsilon C)^{-1}(\boldsymbol{\xi}_k, -\lambda |w_k|), \text{ for } k = 1, \dots, N.$$

4.2 Karush-Kuhn-Tucker conditions

In CFD, one typically obtains

$$C(\varepsilon; \boldsymbol{\xi}) = \varepsilon^{-\alpha},$$

where α is related to the rate of convergence of the discretization. Hence $\mathrm{d}_\varepsilon C = -\alpha \varepsilon^{-\alpha-1}$ and therefore

$$\varepsilon_k = K \left(\frac{|w_k|}{\alpha} \right)^{\frac{1}{1-\alpha}}.$$

K is a scaling parameter to ensure $\sum_{k=1}^N |w_k| \varepsilon_k = \mathcal{E}$.

ε_k depends strongly on w_k and is not bounded from below. If a cubature rule with varying weights is used, any ε_k can become very small. Solutions for smaller ε_k typically require much more computer

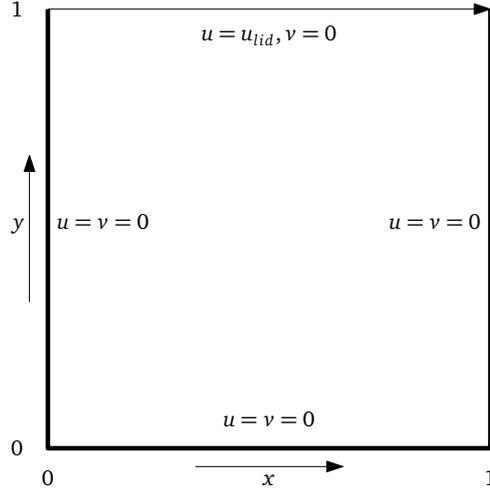


Figure 4: The geometry and boundary conditions of the lid-driven cavity flow problem.

memory to determine, such that in practice ε_k is also bounded from below. It is possible to account for this by adapting the minimization problem of Equation (4.1) to:

$$\arg \min_{\varepsilon_1, \dots, \varepsilon_N} \left\{ \sum_{k=1}^N C(\varepsilon_k; \xi_k) \mid \sum_{k=1}^N |w_k| \varepsilon_k \leq \mathcal{E} \text{ and } \forall k = 1, \dots, N : \varepsilon_k \geq \mathfrak{C} \right\}.$$

This problem can be solved using the Karush-Kuhn-Tucker (KKT) conditions. Let f and g as above and let $h(\varepsilon) = \mathfrak{C} - \varepsilon$, then the solution of the adapted system reads:

$$\left\{ \begin{array}{l} g(\varepsilon_1, \dots, \varepsilon_N) = 0, \\ \nabla f(\varepsilon_1, \dots, \varepsilon_N) - \lambda \nabla g(\varepsilon_1, \dots, \varepsilon_N) - \sum_{k=1}^N \mu_k h'(\varepsilon_k) = \mathbf{0}, \\ h(\varepsilon_k) \leq 0 \quad \forall k = 1, \dots, N, \\ \mu_k \geq 0 \quad \forall k = 1, \dots, N, \\ \mu_k \cdot h(\varepsilon_k) = 0 \quad \forall k = 1, \dots, N. \end{array} \right.$$

If $\varepsilon_k = \mathfrak{C}$ is not a feasible solution, no solution exists at all. It is not trivial to give a closed form expression for the optimal ε_k in this case.

The optimal solution can be determined iteratively. First, determine ε_k using the Lagrange multiplier method discussed above. Then apply the following iterative procedure: determine those ε_k with $h(\varepsilon_k) \leq 0$, say $\{\varepsilon_{i_1}, \varepsilon_{i_2}, \dots, \varepsilon_{i_I}\}$. Update for $j = 1, \dots, I$ $\varepsilon_{i_j} \leftarrow \mathfrak{C}$ and rescale all ε_k with $\varepsilon_k > \mathfrak{C}$. Continue until all $h(\varepsilon_k) > 0$ or $\varepsilon_k = \mathfrak{C}$. In the latter case, the problem can be infeasible, which can be verified by filling in the obtained solution in the equations above.

5 Numerical results

In this section the proposed cubature rule and the distribution of error are applied to two different numerical test cases.

5.1 Lid-driven cavity flow

5.1.1 Problem description

The standard lid-driven cavity flow problem is studied (see Figure 4). The geometry of this two-dimensional flow problem is a unit square box, with four Dirichlet boundary conditions. Only one side has a non-zero boundary condition. The boundary condition at the singular corners is $\mathbf{u} = \mathbf{0}$, where \mathbf{u} is the flow vector.

Two uncertain parameters are imposed: the Reynolds number and the viscosity. Both parameters are assumed to be Beta(a, b)-distributed, with $a = b = 4$. The PDF of the Beta-distribution with parameters

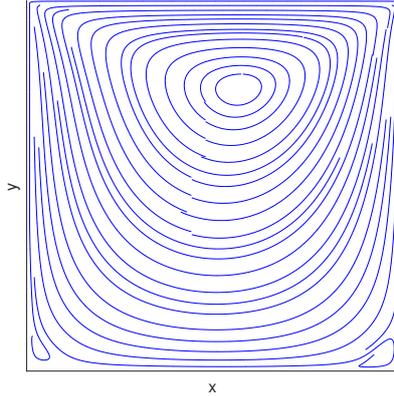


Figure 5: The mean streamline distribution of the lid-driven cavity flow UQ-problem.

a and b equals:

$$p(x; a, b) \propto x^{a-1}(1-x)^{b-1} \text{ for } 0 \leq x \leq 1.$$

The distributions are scaled such that $u_{lid} \in (0.5, 1.5)$ and $\nu \in (0.0038, 0.05)$. Then the Reynolds number is between 10 and 400.

As the applied methods are non-intrusive, a solver for the deterministic problem is necessary. For the current study, a Lattice-Boltzmann method has been implemented. The straightforward D2Q9 BGK-model using Zou–He boundary conditions is used [25], which is a second-order accurate method. The four extreme cases of the uncertain parameters are verified using results from Ghia et al. [8].

5.1.2 Results

Four methods are applied to this problem:

1. Monte Carlo;
2. Stochastic collocation using a Smolyak sparse grid of reduced Gauss quadrature rules;
3. Stochastic collocation using reduced tensor cubature rules, with positive weights;
4. Stochastic collocation using reduced tensor cubature rules, with negative weights.

Stochastic collocation methods using cubature rules (which are applied here) can be accelerated by distributing the error over the nodes. The three methods above are applied twice: once with this distribution and once without. The mean solution of the UQ-problem (which is visually the same for all applied methods) is depicted in Figure 5.

Using a high-fidelity tensor grid (constructed with two 65-node Gaussian rules), a reference solution is obtained. The solutions of the methods above can be compared with this solution to study the convergence. The accuracy is measured with the L^2 -norm of the difference between the mean solution $\bar{u}(N)$, calculated using N nodes, and the reference solution \bar{u}^* , i.e.

$$\epsilon_N := \|\bar{u}(N) - \bar{u}^*\|_2.$$

The convergence for all methods is depicted in Figure 6, where the left figure shows the convergence for the methods applied *without* the distribution of error and the right figure for the methods applied *with* distribution. Assigning to each node a different convergence criterion using the minimization procedure approximately halved the computational cost in each case.

The $\mathcal{O}(1/\sqrt{N})$ converge of Monte Carlo is clearly visible, although only 100 simulations have been performed. The collocation methods show spectral convergence, which was to be expected because the fluid velocity is smooth with respect to the uncertain parameters. If no distribution of error is applied, the Smolyak grid and the reduced cubature rule with positive weights perform equally. The rule with negative weights shows slightly less good convergence. This was expected, as the dimension of the problem is small. If distribution of error is applied, the convergence of the three methods is approximately equal, as the advantage of having positive weights disappears. Especially the convergence of the reduced rules is

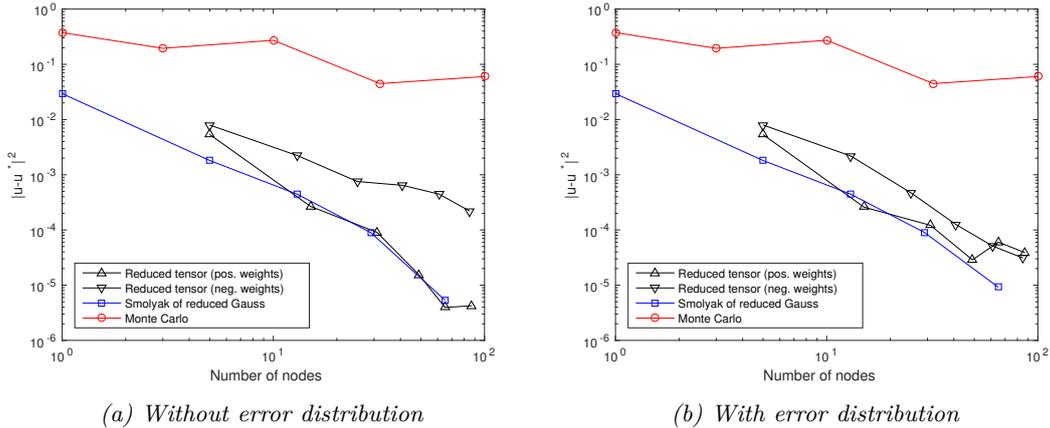


Figure 6: The convergence of the lid-driven cavity flow problem, using a second-order accurate Lattice-Boltzmann code.

not monotonic. We attribute this to the fact that the error is distributed based on the weights of the largest cubature rule, which differ from the weights of the nested rules.

5.2 Three-dimensional aircraft

5.2.1 Problem description

In this section, a three-dimensional flow problem is considered. We consider an aircraft aerodynamics case with seven uncertain parameters. Only the reduced cubature rule with negative weights with distribution of the error is used.

The geometry is based on the sample geometry “twin-engine utility aircraft” of the program `sumo` [4]. As flow model the Euler equations are considered, solved by the second-order accurate finite-volume code `SU2` [16]. The volume mesh is generated using `TetGen` [18].

The seven uncertain parameters can be split into two categories: geometrical and operational parameters.

Three geometrical uncertain parameters are considered, each with a normal distribution and standard deviation of 5% of the mean. The three parameters are the *leading edge radius*, *maximum camber as percentage of the chord*, and *distance of maximum camber from leading edge*. The mean of these parameters is characterized by the NACA2412 airfoil.

Four operational uncertain parameters are considered, each with a Beta(4, 4)-distribution (see the previous section for the PDF). These parameters are the *angle of attack* (with range $2.31^\circ \pm 5\%$), *side-slip angle* (with range $0^\circ \pm 0.5^\circ$), *Mach number* (with range $0.72 \pm 5\%$), and the *free-stream pressure* (with range $101\,325\text{ N/m}^2 \pm 5\%$).

5.2.2 Results

#	c_l	\hat{c}_l	c_d	\hat{c}_d	c_{sf}	\hat{c}_{sf}
0	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
1	0.3640	0.3645	0.0226	0.0228	-0.0015	-0.0002
2	0.1326	0.1330	0.0005	0.0005		
3	0.0483	0.0485				
4	0.0176	0.0177				

Table 2: The first four non-central moments determined either using the cubature rule directly on the results (without hat) or using a high-degree cubature rule on the least-squares estimation (with hat). Empty places are values smaller than 10^{-5} .

The reduced cubature rule with negative weights used in this problem consists of 1,293 nodes, which is small compared to the 2,465 nodes of a Smolyak grid or the 8,713 nodes of the reduced cubature rule with positive weights.

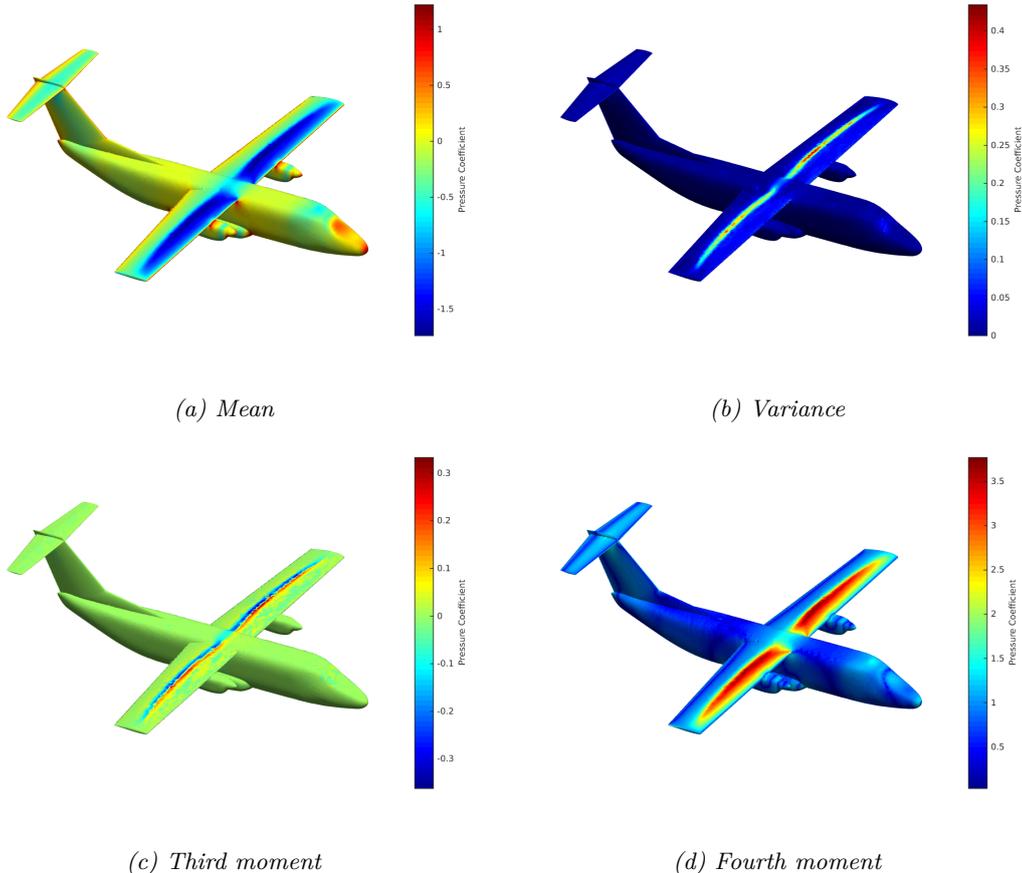


Figure 7: The first four central moments of the pressure coefficient at the wetted surface of the aircraft. Of the k^{th} moment the k^{th} root is taken such that the units are equal.

The lift, drag and side-force coefficients are determined through pressure integration over the surface (see Table 2). The least-squares estimation is obtained by fitting a polynomial of degree 5 on the nodes and determining the respective integrals. The degree 5 is chosen such that the polynomial is of maximal degree with the system (consisting of 1,293 equations) still well-posed.

Although the cubature rule has negative weights, the variance is non-negative, which is not evident. The estimations of the cubature rule and the least-squares estimation are of the same order of magnitude. As there is no reference data available, the values cannot be compared to a true value. The low-order moments are physically of the correct order of magnitude.

The moments of the pressure coefficients can be calculated in a similar manner, but are location dependent. We calculate these ignoring the geometrical uncertainties (i.e., reducing the problem to a 4-dimensional problem) – see Figure 7. The variance highlights the location of the shocks, which shows the location of the highest uncertainty. This result is consistent with UQ analyses of airfoils with shocks (see e.g. [23]).

6 Conclusion

Two computational fluid dynamics cases with uncertain inputs have been studied using a new cubature rule and an optimization procedure. The new cubature rule can be created such that it is symmetric or positive or both. The set of cubature rules is always nested. Conventional quadrature rules such as the Clenshaw–Curtis or Gaussian quadrature rules are either not positive or not nested. The Smolyak cubature rule is nested, but does not have positive weights.

The reduced quadrature rule is perfectly suited for input of the Smolyak procedure. Existing theory about the degree of a Smolyak grid holds and the set of grids shows convergence, which was shown in the lid-driven cavity flow problem.

The reduced cubature rule with positive weights needs approximately an equal amount of nodes as

the Smolyak grid in small dimensions, but has positive weights. The convergence is approximately equal to that of the Smolyak grid.

If complex CFD cases are studied with many uncertain parameters, reducing the number of nodes is essential. We showed that the reduced cubature rule with negative weights can be used in this case, using a small number of nodes. In both CFD cases studied here, the variance was positive, which is not evident as the cubature rule contains negative weights. This can become problematic in other CFD cases.

The optimization procedure discussed improves the computational time of a method without reducing the error. The convergence remains approximately the same. However, if nodes are removed from a cubature rule, all weights are recomputed, such that the set of convergence criteria needs to be re-determined. This is an option for further research.

The initial cubature rule influences the result in two ways: firstly the set of nodes is used for removal and secondly the convergence criteria are deduced using these nodes. If this cubature rule does not contain the number of nodes necessary for convergence, more nodes can be necessary, but no general strategy exists to add nodes to a cubature rule. This is also an open research question.

7 Acknowledgments

This research is supported by the Dutch Technology Foundation STW, which is part of the Netherlands Organisation for Scientific Research (NWO), and which is partly funded by the Ministry of Economic Affairs (project P14-03 EUROS).

References

- [1] R.E. Caflisch. Monte Carlo and quasi-Monte Carlo methods. *Acta Numerica*, 7:1–49, 1998.
- [2] C.W. Clenshaw and A.R. Curtis. A method for numerical integration on an automatic computer. *Numerische Mathematik*, 2:197–205, 1960. doi:10.1007/BF01386223.
- [3] M.S. Eldred and J. Burkardt. Comparison of non-intrusive polynomial chaos and stochastic collocation methods for uncertainty quantification. In *47th AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, number AIAA 2009–976. American Institute of Aeronautics and Astronautics, 2009. doi:10.2307/2670057.
- [4] D. Eller. Larosterna.com: sumo - aircraft geometry and surface modeling tool. <http://www.larosterna.com/sumo.html>, 2009. Accessed 2016-05-09.
- [5] J. Garcke. *Sparse Grids and Applications*, chapter Sparse Grids in a Nutshell, pages 57 – 80. Springer, 2013. doi:10.1007/978-3-642-31703-3.
- [6] T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numerical Algorithms*, 18:209 – 232, 1998. doi:10.1023/A:1019129717644.
- [7] R.G. Ghanem and P.D. Spanos. *Stochastic Finite Elements: A Spectral Approach*. Springer, 1991. doi:10.1007/978-1-4612-3094-6.
- [8] U. Ghia, K.N. Ghia, and C.T. Shin. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics*, 48:387–411, 1982. doi:10.1016/0021-9991(82)90058-4.
- [9] A. Gil, J. Segura, and N.M. Temme. Chebyshev expansion. In *Numerical Methods for Special Functions*, chapter 3, pages 51–86. Society for Industrial and Applied Mathematics, 2007.
- [10] G.H. Golub and J.H. Welsch. Calculation of Gauss quadrature rules. *Mathematics of Computation*, 23:221–230, 1969. doi:10.2307/2004418.
- [11] H.N. Najm. Uncertainty quantification and polynomial chaos techniques in computational fluid dynamics. *Annual Review of Fluid Mechanics*, 41:35–52, 2009. doi:10.1146/annurev.fluid.010908.165248.
- [12] F. Nobile, R. Tempone, and C.G. Webster. A sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46:2309 – 2345, 2008. doi:10.1137/060663660.

- [13] F. Nobile, R. Tempone, and C.G. Webster. An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM Journal on Numerical Analysis*, 46: 2411–2442, 2008. doi:10.1137/070680540.
- [14] E. Novak and K. Ritter. High dimensional integration of smooth functions over cubes. *Numerische Mathematik*, 75:79–97, 1996. doi:10.1007/s002110050231.
- [15] E. Novak and K. Ritter. Simple cubature formulas with high polynomial exactness. *Constructive Approximation*, 15:499–522, 1999. doi:10.1007/s003659900119.
- [16] F. Palacios, T.D. Economou, A.C. Aranake, S.R. Copeland, A.K. Lonkar, T.W. Lukaczyk, D.E. Manosalvas, K.R. Naik, A.S. Padrón, B. Tracey, A. Variyar, and J.J. Alonso. Stanford university unstructured (SU²): Open-source analysis and design technology for turbulent flows. In *52nd AIAA Aerospace Sciences Meeting including The New Horizons Forum and Aerospace Exposition*, number AIAA 2014–0243. American Institute of Aeronautics and Astronautics, 2014. doi:10.2514/6.2014-0243.
- [17] D. Pflüger. *Spatially Adaptive Sparse Grids for High-Dimensional Problems*. PhD thesis, Technische Universität München, 2010.
- [18] H. Si. TetGen, a Delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software*, 41:11, 2015. doi:10.1145/2629697.
- [19] S.A. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Mathematics, Doklady*, 4:240–243, 1963.
- [20] G.W. Wasilkowski and H. Wozniakowski. Explicit cost bounds of algorithms for multivariate tensor product problems. *Journal of Complexity*, 11:1–56, 1995. doi:10.1006/jcom.1995.1001.
- [21] J.A.S. Witteveen and G. Iaccarino. Simplex elements stochastic collocation in higher-dimensional probability spaces. In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, number AIAA 2010–2924. American Institute of Aeronautics and Astronautics, 2010. doi:10.2514/6.2010-2924.
- [22] J.A.S. Witteveen and G. Iaccarino. Simplex stochastic collocation with ENO-type stencil selection for robust uncertainty quantification. *Journal of Computational Physics*, 239:1–21, 2013. doi:10.1016/j.jcp.2012.12.030.
- [23] J.A.S. Witteveen, R. Pečnik, and G. Iaccarino. Uncertainty quantification of the transonic flow around the RAE 2822 airfoil. *Stanford University Center for Turbulence Research: Annual Research Briefs 2009*, pages 93–103, 2009.
- [24] D. Xiu. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press, 2010.
- [25] Q. Zou and X. He. On pressure and velocity boundary conditions for the Lattice Boltzmann BGK model. *Physics of Fluids*, 9:1591–1598, 1997.