Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Unscented Transform and Sparse Grids Applied to Buzzsaw Noise Modeling in Airplane Engines

Gabriel Augusto Correia

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientador
Prof. José Edil Guimarães de Medeiros

Brasília
2021

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Unscented Transform and Sparse Grids Applied to Buzzsaw Noise Modeling in Airplane Engines

Gabriel Augusto Correia

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Prof. José Edil Guimarães de Medeiros (Orientador)
ENE/UnB

Prof. Leonardo Rodrigues Araújo Xavier de Menezes
ENE/UnB

Prof. Bráulio Gutierrez Pimenta
ENM/UnB

Prof. João José Costa Gondim
Coordenador do Curso de Engenharia da Computação

Brasília, 02 de junho de 2021

Agradecimentos

Agradeço aos meus pais, por sempre ressaltarem o valor da educação e por serem exemplos para mim.

Agradeço aos meus avós e tias, pelo apoio e carinho que influenciaram fortemente quem sou.

Agradeço à minha namorada, Lumi, por todo carinho e apoio que me deu nos difíceis anos da graduação e garantir que eu me formasse com algum nível de sanidade.

Agradeço ao meu orientador, Professor José Edil, pelos conselhos, pelas várias horas dedicadas a me orientar e por mostrar na pele o que significa ser um bom professor e orientador.

Agradeço ao meu amigo Alexandre, por ter me ensinado os primeiros passos da programação aos 12 anos.

Finalmente, agradeço à Alexandra Elbakyan e aos demais voluntários que lutam por tornar a ciência propriedade compartilhada da humanidade.

Resumo

Este trabalho apresenta os fundamentos da Transformada da Incerteza (Unscented Transform) e do método computacional das Malhas Esparsas (Sparse Grids) para aplicação em simulações de alta dimensionalidade sujeitas a incertezas. O algoritmo proposto é aplicado na obtenção da assinatura de ruído de uma turbina de avião comercial sujeito a alterações nos ângulos nominais das pás (problema com 16 dimensões) com velocidade supersônica da ponta das pás relativamente ao ar em escoamento, sendo a primeira vez que tal técnica é aplicada em aeroacústica, até onde vai o conhecimento do autor. A simulação obtém com sucesso a assinatura de ruído em tempo inferior à uma hora em máquina GNU/Linux com 8GB de memória e processador de 2.5GHz, mostrando a eficácia da técnica em lidar com problemas de alta dimensionalidade. Perspectivas de aprimoramento da técnica e pesquisas futuras são discutidas ao final.

Palavras-chave: Unscented transform, Sparse grids, Aeroacústica

Abstract

This work presents the foundations of the Unscented Transform and of the computational method of Sparse Grids for applications in high-dimensional simulations subject to uncertainty. The proposed algorithm is applied in obtaining the noise signature of a commercial airplane turbine subject to changes in the blades angles (a 16 dimensional problem) with supersonic speed of the blade tips relative to the air flow, being the first time this technique is applied in this context, as far as the author knows. The simulation successfully obtained the noise signature with time inferior to an hour in a GNU/Linux machine with 8GB of RAM and a processor @ 2.5GHz, showing the technique effectiveness in dealing with high dimensional problems. Perspectives for improvement of the technique and future researches are discussed at the end.

Keywords: Unscented transform, Sparse grids, Aeroacoustics

Contents

List of Figures

Chapter 1

Introduction

In various branches of Engineering, there is the ever present design problem, that is, given project specifications (a problem to solve with constraints), *build* a solution which solves the problem. This contrasts with simulation and testing: the given data is a model or concrete object, which is set under (controlled) conditions and the corresponding response recorded. In this way, they are inverse problems of each other (as in the design problem I know what answer I want, but have to create a way to obtain it).

In this work, we'll be using a Sparse Grid implementation of the Unscented Transform (UT) to solve a particular simulation problem: given a model of an airplane turbine with blade angles as parameter, determine the noise generated by supersonic rotor cascades. Such problem can be solved (maybe in a more computationally efficient way) with methods alternative to ours, for instance those mentioned in [2]. However, there are two main reasons for solving this problem with this particular implementation of the Unscented Transform: 1) to mitigate the curse of dimensionality problem in computations involving high dimensions (16 in our case) while preserving the probabilistic moments of a random continuous distribution, and 2) the prospective of solving the inverse problem: given the requisites, generate the design specifications (see [3] for a successful example relating to Analog-Digital Converters using a different implementation of the UT called *Extended UT*). Of course, the design considering the noise generated by the turbine is important, for example, because in residential areas there is a noise threshold that should not be exceeded and "reducing aircraft noise has been seen as a priority for the civil aircraft industry" [2].

Note that, although we have this particular problem in mind, the solution method is quite general: we take as input a continuous random variable $\mathbf{x}$ and a "black box" model f and use the Unscented Transform to generate a discrete distribution $\mathbf{X}$ with the same moments as $\mathbf{x}$ (up to a certain order). We then generate a range of responses with f according to the uncertainty $\mathbf{X}$. As the Unscented Transform generates a linear system

of equations [3] (in contrast with a nonlinear behavior), it is hoped that it'll be possible to use this to solve the inverse problem (the design problem). This work is expected to provide a first step towards this goal in subsequent works. In our particular application, f is the turbine model and $\mathbf{x}$ describes the blade angles variability (for instance due to manufacturing tolerances or fatigue). This general situation is illustrated in Figure 1.1, while a more detailed depiction will be provided in Chapter 4. The UT takes as input the number of desired sigma-points and the random variable $\mathbf{x}$ (often in the form of a probability distribution function) and output N pairs (s, w) corresponding to the discrete distribution $\mathbf{X}$ (s is a selected point and w the corresponding probability weight). The points s are presented to the model f and the weights w are later used in the analysis step.

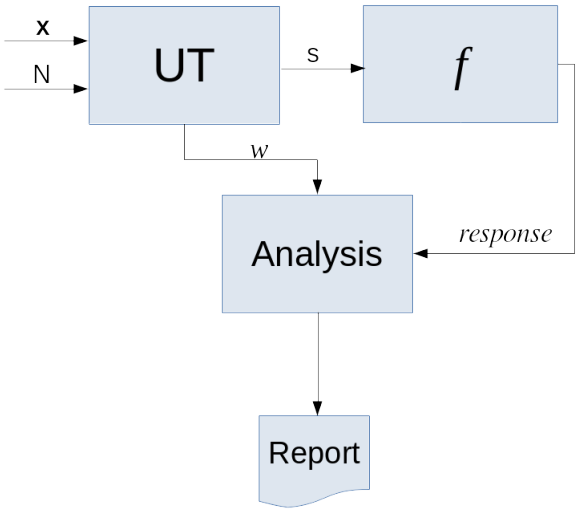

Figure 1.1: Simplified flow showing the use of the Unscented Transform (UT): $\mathbf{x}$ is the random variable, N is the number of desired sigma-points, (s,w) are the sigma-points and f is a model or function.

This work aims in introducing the Sparse Grid method in the context of the Unscented Transform as a way to understand complex systems (in the sense of number of parameters) subject to uncertainty in a computationally feasible way and presenting a case study in the area of aeroacoustics showing the applicability of such technique in a novel way. In a sense, it is a sequel to the work started by de Medeiros in [3], where the unidimensional Unscented Transform was thoroughly studied and applied to the design of a Quantizer in the context of analog-to-digital and digital-to-analog converters, as we extend the realm of applications as well as surpass the dimensional limitation of that work.

The path of this work is the following: we begin Chapter 2 by talking about the application, then we revise the bibliography related to the Unscented Transform in 1 dimension (definition, equations, UT as quadrature, interpolation, orthogonal polynomials,

etc). Then, in Chapter 3, we tackle the UT in the general n-dimensional setting, with a discussion of Sparse Grids and it's use in the implementation of the multidimensional UT. Finally, we will implement this for the specific problem of determining the noise signature of the turbine according to the blades angle (Chapter 4), giving concluding remarks in Chapter 5.

Chapter 2

Unscented Transform

It is desirable to be able to give a certain probability distribution (e.g. Gaussian distribution) relating to the different possible configurations, say of the turbofan engine, and from this predict the noise signature. In what follows, we will use the simplifying assumption that the only attribute subject to change are the blade angles. Details about the complete model are discussed in [2].

Thus, we want to determine as precisely as possible how much variations in the blade angles affect the noise signature of the engine. Even with this simplifying assumption, simulations are mostly unfeasible due to the high number of dimensions involved: there are 16 blades, so we need 16 probability distributions and then simulate each of the possible combinations of the blade angles, which simple combinatorical estimates show to be completely unfeasible from a computational point of view.

If we could, however, reduce the number of points needed in this computation while preserving the probabilistic attributes of the distributions (i.e. the moments), we would be closer to a feasible simulation proposal. This is where our first main tool appears: the Unscented Transform.

2.1 Unidimensional Unscented Transform

The Unscented Transform is a functional that takes as input a continuous random variable and returns a discrete version preserving a certain number of probabilistic moments. This transformation appeared in the context of Kalman Filtering and nonlinear approximation [4], giving an idea of the applicability of this tool. In fact, "The unscented transformation (UT) was developed to address the deficiencies of linearization by providing a more direct and explicit mechanism for transforming mean and covariance information." [4]

Let's begin by defining the Unscented Transform, according to [3]. In what follows, RV will stand for Random Variable, UT for Unscented Transform, boldface small letters stand for continuous RV and boldface capital letters stand for discrete RV.

Definition 1 [3] The α-order *Unscented Transform* of a continuous RV $\mathbf{x}$ with probability density function $p_x(\mathbf{x})$ is a set of n pairs $\{(s_i, w_i)\}_{i=1}^{n}$ (called sigma-points) characterizing a discrete probability density function $F_x(\mathbf{X})$ such that

$$E\{\mathbf{X}^k\} = \sum_{i=1}^{n} s_i^k w_i = \int_{-\infty}^{+\infty} x^k p_x(\mathbf{x}) dx = E\{\mathbf{x}^k\} \tag{2.1}$$

for all $k = 1, \ldots, \alpha$, that is, both the discrete and the continuous distributions have the same moments up to the α-th order. By abuse of notation we also call each s_i a *sigma-point*, while the w_i are the associated *weights*.

One important aspect of the UT is that "the sigma points are not drawn at random; they are deterministically chosen so that they exhibit certain specific properties (e.g., have a given mean and covariance). As a result, high-order information about the distribution can be captured with a fixed, small number of points." [4] This contrasts with Monte-Carlo methods, which are probabilistic in essence [5].

Definition 2 [3] Given a Riemann-Stieljes integrable function over a compact interval $[a, b]$, let $P = a = t_1 < t_2 < \cdots < t_n = b$ be a partition of $[a, b]$ and $\lambda_1, \ldots, \lambda_n$ real numbers, then

$$Q_n(f) = \sum_{i=1}^{n} \lambda_i f(t_i) \tag{2.2}$$

is called a *mechanical quadrature* of f with *Cotes numbers* $\lambda_1, \ldots, \lambda_n$ [3].

The mechanical quadrature is an arbitrary process that maps sums of weighted samples of a function $f(x)$ into numbers. Of special interest for this work is the special case for the mechanical quadrature in which the Cotes numbers λ_n are defined such that Eq.(2.2) holds if $f(x)$ is an arbitrary polynomial π_{n-1} of order up to $n-1$ and $u(x)$ is a non-decreasing function (giving a measure for the integral). In this case $Q_n(f)$ is called a quadrature of the interpolatory type for reasons we explicit in the following sections. Thus, the UT as per Definition 1 is a special case of this interpolatory quadrature in which we recognize that $f(x) = x^k$ are monomials, the sigma points $s_i = x_i$ are the abscissas of the mechanical quadrature, the weights $w_i = \lambda_i$ are the Cotes numbers and $du(x) = p_x(x)dx$ relates to the probability density function that characterizes the RV of interest [3].

By the strong law of large numbers and the relative frequency interpretation of probabilities, one sees that the expectancy yields an approximation for a given integral (via uniform distribution and the appropriate constant of proportionality) on the real line. On the other hand, the UT gives a mechanical quadrature of the first moment, which also approximates the integral. Thus, we see the UT as an alternative method for the Monte-Carlo method (which numerically samples random numbers in a domain and apply them to the function, computing the average to give the integral approximation) [3].

The problem is that, according to numerical simulations done in [3, p.35], the UT approximates very poorly the probability function, even though it accurately estimates the moments. The Extended Unscented Transform of Menezes [6] improves in this issue. Algorithm 3.3.1 of [3, p.37] gives a direct method of calculation of the ExUT. However, for the purposes of this work we do not need the ExUT. Instead, we calculate the UT sigma-points and weights via an algorithm based on the Gaussian quadrature, as the following subsection shows.

2.1.1 Interpolatory Quadrature and Orthogonal polynomials

The process of constructing a function which takes on given *data values* at given *data points* is called *interpolation* [7].

To better work with this concept, we define the *interpolation problem* as in [3].

Definition 3 [3] The *interpolation problem* consists of determining a set of parameters a_i so that for $n+1$ given real or complex pairs (x_i, y_i), $i = 0, \ldots, n$, with $x_i \neq x_k$ for $i \neq k$,

$$\Phi(x_i; a_0, \ldots, a_n) = y_i \tag{2.3}$$

holds for $i = 0, \ldots, n$ in which $\Phi(x; a_0, ..., a_n)$ is a family of single variable functions completely defined by the parameters a_i. We call the pairs (x_i, y_i) interpolation *support points*, the locations x_i *support abscissas* and the values y_i *support ordinates*.

Of particular interest for this work is the classical *polynomial interpolation problem* [3], in which

$$\Phi(x_i; a_0, \ldots, a_n) \equiv a_0 + a_1 x + \cdots + a_n x^n. \tag{2.4}$$

The classical polynomial interpolation method is stated in the following

Theorem 2.1.1 *Given $n+1$ pairwise distinct data points $x_0, \ldots, x_n$ and associated values $y_0, \ldots, y_n$, there is a unique polynomial of degree at most n which takes on these values.*

This theorem gives the existence of a solution to the polynomial interpolation problem. In fact, it's proof even gives an algorithm for constructing a solution, known as *Lagrange's*

Interpolation formula, given by Eq.(2.5). Via integration, one can establish the weights in (2.1) for a given set of sigma-points by Eq.(2.6). It does not indicate, however, how to choose a suitable set of sigma-points. This is where the Gaussian Quadrature comes in [3].

$$L(x) = \sum_{i=0}^{n} y_i \prod_{\substack{k=0 \\ k\neq i}}^{n} \frac{x - x_k}{x_i - x_k}. \tag{2.5}$$

$$w_i = \lambda_i = \int_a^b \prod_{\substack{k=0 \\ k\neq i}}^{n} \frac{x - x_k}{x_i - x_k} du(x). \tag{2.6}$$

First, let's recall the definition of scalar product on $L^2([a,b])$ with respect to a non-decreasing non-constant weight function $u(x)$ [3]:

$$\langle f, g\rangle = \int_a^b f(x)g(x)du(x). \tag{2.7}$$

If $u(x)$ is absolutely continuous, which is the case treated in the scope of this work, the scalar product reduces to

$$\langle f, g\rangle = \int_a^b f(x)g(x)p_x(x)dx, \tag{2.8}$$

where $p_x(x)$ is the continuous probability density function of interest in the UT computation.

The existence of a sequence of orthogonal polynomials associated with the distribution $p_x(x)$ is established in the following

Theorem 2.1.2 *There exist orthogonal polynomials $\rho_j \in \pi_j$ $j = 0, 1, 2, \ldots$, such that $\langle \rho_i, \rho_k\rangle = 0$, for $i \neq k$. These polynomials are uniquely determined by a three term recurrence relation given by*

$$\rho_0(x) = 1 = \rho_{-1}(x) \tag{2.9}$$

$$\rho_{i+1}(x) = (x - \alpha_{i+1})\rho_i(x) - \beta_{i+1}\rho_{i-1}(x) \tag{2.10}$$

for $i \geq 0$ and

$$\alpha_{i+1} = \frac{\langle x\rho_i(x), \rho_i(x)\rangle}{\|\rho_i(x)\|_{L^2}^2} \text{ for } i \geq 0 \tag{2.11}$$

$$\beta_0 = 1, \beta_{i+1}^2 = \frac{\|\rho_i(x)\|_{L^2}^2}{\|\rho_{i-1}(x)\|_{L^2}^2} \text{ for } i \geq 1 \tag{2.12}$$

where

$$\|f\|_{L^2} := \sqrt{\langle f, f\rangle}. \tag{2.13}$$

The following result is what is sometimes called *Gauss' interpolation*, whose proof gives an explicit way of calculating Cotes' numbers, while Theorem 2.1.4 shows optimality in the sense of the polynomial degree.

Theorem 2.1.3 *If $x_1 < x_2 < \cdots < x_n$ denote the zeros of $\rho_n(x)$, there exist real numbers $\lambda_1, \ldots, \lambda_n$ such that*

$$\int_a^b p(x)du(x) = \sum_{i=1}^{n} \lambda_i p(x_i) \tag{2.14}$$

whenever $p(x)$ is an arbitrary polynomial of degree at most $2n-1$. The distribution $du(x)$ and the integer n uniquely determine these numbers λ_n.

Theorem 2.1.4 *It is not possible to find numbers $x_i, i = 1, \ldots, n$, such that Eq. (2.14) holds for all polynomials $p(x) \in \pi_{2n}$.*

Finally, we state two theorems which provide a practical way for computing the sigma-points and weight pairs $\{(s_i, w_i)\}$ based on the theory of the Gaussian quadrature.

Theorem 2.1.5 *The roots x_i, $i = 1, \ldots, n$, of the orthogonal polynomial $\rho_n(x)$ are the eigenvalues of the tridiagonal matrix*

$$J_n = \begin{bmatrix} \alpha_1 & \sqrt{\beta_2} & 0 & 0 & 0 & \ldots & 0 \\ \sqrt{\beta_2} & \alpha_2 & \sqrt{\beta_3} & 0 & 0 & \ldots & 0 \\ 0 & \sqrt{\beta_3} & \alpha_3 & \sqrt{\beta_4} & 0 & \ldots & 0 \\ & & \ddots & \ddots & \ddots & & \\ 0 & 0 & \ldots & \sqrt{\beta_{n-2}} & \alpha_{n-2} & \sqrt{\beta_{n-1}} & 0 \\ 0 & 0 & \ldots & 0 & \sqrt{\beta_{n-1}} & \alpha_{n-1} & \sqrt{\beta_n} \\ 0 & 0 & \ldots & 0 & 0 & \sqrt{\beta_n} & \alpha_n \end{bmatrix} \tag{2.15}$$

where α_i and β_i are the coefficients of the three term recurrence relation defined in Eq.(2.11-2.12).

Theorem 2.1.6 *Let $v^{(i)} = (v_1^{(i)}, \ldots, v_n^{(i)})^T$ be an eigenvector of J_n defined in Eq.(2.15) for the eigenvalue X_i, i.e., $J_n v^{(i)} = x_i v^{(i)}$. Suppose $v^{(i)}$ is scaled in such a way that*

$$v^{(i)T} v^{(i)} = \langle \rho_0, \rho_0 \rangle = \int_a^b du(x).$$

Then the weights w_i are given by

$$w_i = (v_1^{(i)})^2, i = 1, \ldots, n. \tag{2.16}$$

We are now in position to give an Algorithm that calculates the Unscented Transform. It should be noted that, by Definition 1 of the UT, different outputs $\mathbf{X}$ are possible for a given input $\mathbf{x}$ (as is seen by explicit examples in [3]), so the following outputs only a particular solution.

Algorithm Gaussian quadrature method for the Unscented Transform [3].

1. Start with the desired continuous probability distribution $p_x(x)$ and the desired number of sigma-points n. Compute the α_n and β_n , $n = 1, ..., n$, terms in the three term recurrence relation using equations (2.10)-(2.12).
2. Build the Jacobi J_n matrix in equation (2.15)
3. Use any known method to compute the eigenvalues of J_n and make them the quadrature abscissas x_i.
4. Use any known method to compute the eigenvectors of J_n and scaled them as in Theorem 2.1.6. Use the first eigenvector components as the λ_i quadrature weights.
5. Scale the quadrature abscissas x_i and weights λ_i according to the desired probability distribution function and use them as the sigma-points s_i and weights w_i of the UT.

For an elucidating example demonstrating the calculations of the above algorithm, see the second chapter of [3].

2.2 Multidimensional UT

The multidimensional UT is formally identical to the one-dimensional case, changing the Random Variable to a *Random Vector* X subject to a measurable function f and the single integral to the appropriate multidimensional integral, for instance the expectancy is [8, 9]:

$$E\{f(X)\} = \int_{\mathbb{R}^n} f(\xi) p_X(\xi) d\xi = \sum w_{i_1} \cdots w_{i_n} f(x_{i_1}, \ldots, x_{i_n}) = E\{f(x)\} \tag{2.17}$$

As for the computation, the multivariate case can be obtained by first using a stochastic decoupling technique $X' = (\sqrt{P_{XX}})^{-1}(X - \tilde{X})$, where X' is a multivariate standard Gaussian RV and P_{XX} represents the covariance with respect to X [10]. Then, for $\tilde{f}(X) := f(\sqrt{P_{XX}}^T X + \tilde{X})$, the Gauss-Hermite-Quadrature is applied on the form

$$E_{X'}\{\tilde{f}(X')\} = \int_{\mathbb{R}^n} \tilde{f}(\xi) p_{X'}(\xi) d\xi \approx \sum w_{i_1} \cdots w_{i_n} \tilde{f}(x_{i_1}, \ldots, x_{i_n}) \tag{2.18}$$

and $E_X\{f(X)\}$ is obtained from $E_{X'}\{\tilde{f}(X')\}$. An alternative to solving the multivariate Gaussian case is to use the spherical cubature rule along with the Gaussian Quadrature

after performing a Cartesian-to-spherical coordinate transformation. In fact, consider the Gaussian case $p_\xi(\xi) = \exp(-\xi\xi^T)$ and let $\xi = \rho y$, with $y^T y = 1, \rho \in [0, \infty)$. In this case, (2.17) becomes $E\{f(X)\} = \int_0^\infty S(\rho)\rho^{n-1}\exp(-r^2)d\rho$, $S(\rho) := \int_{S^{n-1}} f(\rho y)d\phi(y)$ and $\phi(\cdot)$ is the spherical surface measure of S^{n-1}. The spherical integral $S(\rho)$ is solved by the spherical cubature rule, while the expectation by a Gaussian Quadrature rule [8].

For our particular application, we consider the random variables to be independent of each other and the function f to be a multivariable polynomial (as in [11]), considerably simplifying the calculation of the moments. In fact, we have the following general probability result [9]:

Theorem 2.2.1 *If $X_1, \ldots, X_n$ are independent integrable random variables, then $\prod_{i=1}^n X_i$ is integrable and*

$$E(X_1 X_2 \cdots X_n) = \prod_{i=1}^{n} E(X_i). \tag{2.19}$$

Another alternative for obtaining (2.17) is by approximating $p_X(x)$. We can classify this type of suboptimal approximation into two categories, namely Monte Carlo methods, and sigma-point methods. Monte Carlo (MC) methods consist of randomly taking a very large quantity of samples x_i of $\mathbf{X}$ (the method gets more accurate as the number of samples $N \to +\infty$). Sigma point methods, on the other hand, consist on analytically choosing finite N samples s_i and weights w_i. These approaches can be viewed as generalized (negative weights are admitted) discrete approximations of $p_X(x)$ [8].

A different classification is given by [11], where numerical methods for moment integrals are classified as *product* and *non-product rules*. Sparse grids, Monte-Carlo and Quasi-Monte-Carlo are all non-product rules, thus alternative to each other. Again, a key difference is the probabilistic nature of the latter methods, while the former is deterministic. The next chapter details the Sparse Grid construction and applies it to our UT problem, underlying it's importance in damping the high dimensionality cost of computation.

Chapter 3

Sparse Grids

The general approximation problem can be easily described for normed vector spaces [7, p.136]: let $(V, \|\cdot\|)$ be a normed vector space and $T \subset V$. Given $v \in V$ and $\varepsilon > 0$, an ε-approximation for v is an element $u \in T$ such that $\|u - v\| < \varepsilon$. The smaller the ε, the better the approximation (in the sense of the norm). In this work, we'll be mainly interested in function spaces (such as L^p and Sobolev Space's), which are infinite dimensional vector spaces and the natural language for studying those spaces comes from Functional Analysis (see for instance [12]).

Our problem reduces to finding an element of a suitable class which is close enough to the precise element v. To achieve this, we must find a way of constructing such approximation. For instance, if v is the integral of a function, we could approximate it by a partial Riemann sum. In this case, the computational cost is related to the number of summation terms, the choosing of the points in the interval's partition and the calculation of the function in those points.

In the Sparse Grid method, our approximation space will be a subset of a Sobolev Space, with norm related to the function's (higher order) derivatives. The motivation for the Sparse Grid method is breaking the *curse of dimensionality* of classical approximation schemes, that is

$$\|f - f_n\| = O(n^{-r/d}) \tag{3.1}$$

where r denotes the isotropic smoothness of the function f and d the problem's dimension. At first glance, it seems that an obvious solution would suffice: just increase the regularity of the function to $r = O(d)$ and obtain directly $\|f - f_n\| = O(n^{-c})$ for a positive constant c. However, such assumptions are unrealistic and a different approach is necessary [1].

In this following four sections, a formal justification of how Sparse Grids arise naturally as solution to an optimization problem (related to the number of grid points, of course) and it's essential properties are given. We'll be closely following the remarkable paper [1] of Bungartz-Griebel, but ignoring the energy-norm, as it won't be used for our purposes.

3.1 Piecewise linear interpolation on sparse grids

As a first approach to sparse grids and their underlying hierarchical multilevel setting, we discuss the problem of interpolating smooth functions with the help of piecewise d-linear hierarchical bases. For that, we introduce a tensor product-based subspace splitting and study the resulting subspaces. Starting from their properties, sparse grids are defined via an optimization process in a cost–benefit spirit closely related to the notion of n-term approximation. Out of the variety of norms with respect to which such an optimized discretization scheme can be derived, we restrict ourselves to the L^2- and the L^∞-norm, and thus to the respective types of sparse grids. After presenting the most important approximation properties of the latter, a short digression into recurrences and complexity will demonstrate their asymptotic characteristics and, consequently, their potential for problems of high dimensionality.

Let us start with some notation and with the preliminaries that are necessary for a detailed discussion of sparse grids for purposes of interpolation or approximation, respectively. On the d-dimensional unit interval $\bar{\Omega} := [0,1]^d$, we consider multivariate functions $u : \bar{\Omega} \to \mathbb{R}$, $u(x) \in \mathbb{R}, x := (x_1, \ldots, x_d) \in \bar{\Omega}$, with (in some sense) bounded weak mixed derivatives

$$D^\alpha u = \frac{\partial^{|\alpha|_1} u}{\partial x_1^{\alpha_1} \cdots x_d^{\alpha_d}} \tag{3.2}$$

up to some given order $r \in \mathbb{N}_0$. Here, $\alpha \in \mathbb{N}_0^d$ denotes a d-dimensional multi-index with the two norms

$$|\alpha|_1 := \sum_{i=1}^{d} \alpha_j \text{ and } |\alpha|_\infty := \max_{1 \le j \le d} \alpha_j.$$

In the context of multi-indices, we use component-wise arithmetic operations, see [1] for details. We will use the following notation for the special multi-indices

$$\mathbf{0} = (0, \ldots, 0),\ \mathbf{1} = (1, \ldots, 1) \text{ and } \mathbf{2} = (2, \ldots, 2)$$

In the following, for $q \in \{2, \infty\}$ and $r \in \mathbb{N}_0$, we study the spaces $X^{q,r}(\bar{\Omega})$ of all functions of bounded (with respect to the L_q-norm) mixed derivatives up to the order r, and $X_0^{q,r}(\bar{\Omega})$ will be the subspace of $X^{q,r}(\bar{\Omega})$ consisting of those $u \in X^r(\bar{\Omega})$ vanishing on the boundary $\partial\Omega$. Note that, for the theoretical considerations, we shall restrict ourselves to the case of homogeneous boundary conditions. Furthermore, note that we omit the ambient dimension d when clear from the context. Concerning the smoothness parameter $r \in \mathbb{N}_0$, we need $r = 2$ for the case of piecewise linear approximations which, for the moment, will be in the centre of interest. Finally, for functions $u \in X_0^{q,r}(\bar{\Omega})$ and multi-

indices α with $|\alpha|_\infty \leq r$, we introduce the seminorms

$$\begin{aligned} u_{\alpha,\infty} &:= \|D^\alpha u\|_\infty, \\ u_{\alpha,2} &:= \|D^\alpha u\|_2 = \left(\int_{\bar{\Omega}} |D^\alpha u|^2 dx \right)^{1/2}. \end{aligned} \tag{3.3}$$

Now, with the multi-index $l = (l_1, \ldots, l_d) \in \mathbb{N}^d$, which indicates the level in a multivariate sense, we consider the family of d-dimensional standard rectangular grids Ω_l on $\bar{\Omega}$ with mesh size $h_l = (h_{l_1}, \ldots, h_{l_d}) := 2^{-l} = (2^{-l_1}, \ldots, 2^{-l_d})$. That is, the grid Ω_l is equidistant with respect to each individual coordinate direction, but, in general, may have different mesh sizes in the different coordinate directions. The grid points $x_{l,i}$ of grid Ω_l are just the points $x_{l,i} := (x_{l_1,i_1}, \ldots, x_{l_n,i_n}) := i \cdot h_l, \mathbf{0} \leq i \leq 2^l$. Figure 3.1 shows a clear image of the grid and this terminology for $l = (2, 1)$.

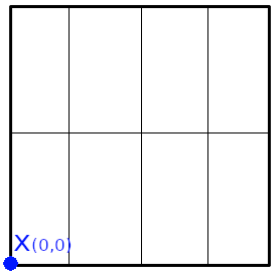

Figure 3.1: Grid $\Omega_{(2,1)}$ with mesh size $h_l = (1/4, 1/2)$. The grid points $x_{l,i}$ are simply the intersection of the straight lines parallel to the square sides.

Thus, here and in the following, the multi-index l indicates the level (of a grid, a point, or, later on, a basis function, respectively), whereas the multi-index i denotes the location of a given grid point $x_{l,i}$ in the respective grid Ω_l.

Next, we have to define discrete approximation spaces and sets of basis functions that span those discrete spaces. In a piecewise linear setting, the simplest choice of a 1D basis function is the standard hat function $\phi(x)$ (see top graph of Figure 3.3),

$$\phi(x) = \begin{cases} 1 - |x|, & \text{if } x \in [-1, 1] \\ 0, & \text{otherwise.} \end{cases} \tag{3.4}$$

This mother of all piecewise linear basis functions can be used to generate an arbitrary $\phi_{l_j,i_j}(x_j)$ with support $[x_{l_j,i_j} - h_{l_j}, x_{l_j,i_j} + h_{l_j}] = [(i_j - 1)h_{l_j}, (i_j + 1)h_{l_j}]$ by dilation and translation, that is,

$$\phi_{l_j,i_j}(x_j) := \phi\left(\frac{x_j - i_j \cdot h_{l_j}}{h_{l_j}} \right). \tag{3.5}$$

The resulting 1D basis functions are the input of the tensor product construction which provides a suitable piecewise d-linear basis function in each grid point $x_{l,i}$, as in Figure 3.2:

$$\phi_{l,i}(x) = \prod_{j=1}^{d} \phi_{l_j,i_j}(x_j). \tag{3.6}$$

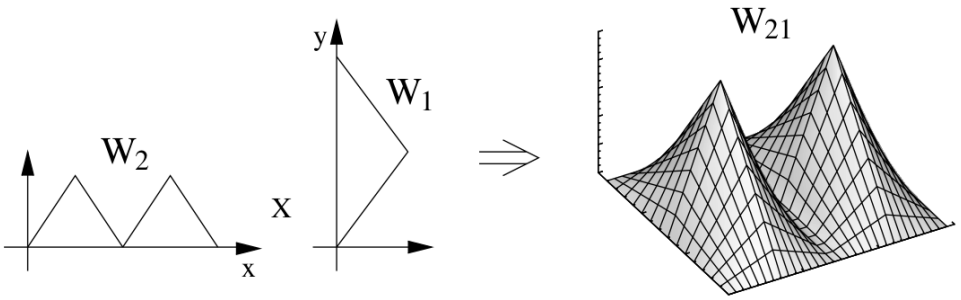

Figure 3.2: Tensor product approach for piecewise bilinear basis functions. Source: Bungartz-Griebel [1]

.

Since we deal with homogeneous boundary conditions (i.e., with $X_0^{q,2}(\bar{\Omega})$), only those $\phi_{l,i}(x)$ that correspond to inner grid points are taken into account for the definition of

$$V_l := \text{span}\{\phi_{l,i} : 1 \leq i \leq 2^l - 1\} \tag{3.7}$$

the space of piecewise d-linear functions with respect to the interior of Ω_l. By definition of span of a set of vectors, the $\phi_{l,i}$ form a basis of V_l, with one basis function $\phi_{l,i}$ of a support of the fixed size $2 \cdot h_l$ for each inner grid point $x_{l,i}$ of Ω_l , and this basis $\{\phi_{l,i}\}$ is just the standard *nodal point basis* of the finite-dimensional space V_l.

Additionally, we introduce the hierarchical increments W_l,

$$W_l := \text{span}\{\phi_{l,i} : 1 \leq i \leq 2^l - 1, \, i_j \text{ odd for all } 1 \leq j \leq d\} \tag{3.8}$$

for which the relation

$$V_l = \bigoplus_{k \leq l} W_k \tag{3.9}$$

can be easily seen. The V_l can be seen as the union of the solid and dashed line functions in Figure 3.3, while W_l represents only the solid lines. Note that the supports of all basis functions $\phi_{l,i}$ spanning W_l are mutually disjoint. Thus, with the index set

$$I_l := \{i \in \mathbb{N}^d : 1 \leq i \leq 2^l - 1, \, i_j \text{ odd for all } 1 \leq j \leq d\} \tag{3.10}$$

we get another basis of V_l , the *hierarchical basis*,

$$\{\phi_{k,i} : i \in I_k, k \leq l\} \tag{3.11}$$

which generalizes the well-known 1D basis shown in Figure 3.3 to the d-dimensional case by means of a tensor product approach. With these hierarchical difference spaces W_l, we can define

$$V := \sum_{l_1=1}^{\infty} \cdots \sum_{l_d=1}^{\infty} W_{(l_1,\ldots,l_d)} = \bigoplus_{l \in \mathbb{N}^d} W_l \tag{3.12}$$

with it's natural hierarchical basis

$$\{\phi_{l,i} : i \in I_l, l \in \mathbb{N}^d\} \tag{3.13}$$

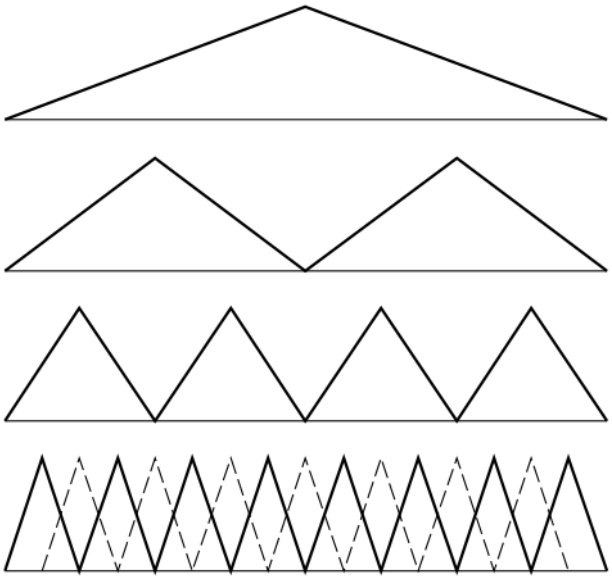

Figure 3.3: Piecewise linear hierarchical basis (solid) vs. nodal point basis (dashed). Source: Bungartz-Griebel [1]

.

Except for completion with respect to the H^1-norm, V is simply the underlying Sobolev space $H_0^1(\bar{\Omega})$, i.e., $\bar{V} = H_0^1(\bar{\Omega})$. Analytically, this means that any function in our space can be approximated by an element of V with arbitrary precision. For the Electrical Engineer and those familiar with Fourier Series, this is exactly like a Fourier representation of a function in L^2, in the sense that the trigonometric series are dense in L^2.

Later we shall deal with finite-dimensional subspaces of V. Note that, for instance, with the discrete spaces

$$V_n^{(\infty)} := \bigoplus_{|l|_\infty \leq n} W_l \tag{3.14}$$

the limit

$$\lim_{n\to\infty} V_n^{\infty} = \lim_{n\to\infty} \bigoplus_{|l|_\infty \leq n} W_l := \bigcup_{n=1}^{\infty} V_n^{(\infty)} = V \tag{3.15}$$

exists due to $V_n^{(\infty)} \subset V_{n+1}^{(\infty)}$. Hence, any function $u \in H_0^1(\bar{\Omega})$ and, consequently, any $u \in X_0^{q,2}(\bar{\Omega})$ can be uniquely split by

$$u(x) = \sum_l u_l(x),\ u_l(x) = \sum_{i \in I_l} v_{l,i} \cdot \phi_{l,i}(x) \in W_l, \tag{3.16}$$

where the $v_{l,i} \in \mathbb{R}$ are the coefficient values of the hierarchical product basis representation of u also called *hierarchical surplus*.

Before we turn to finite-dimensional approximation spaces for $X_0^{q,2}(\bar{\Omega})$, we summarize the most important properties of the hierarchical subspaces W_l, still following closely [1].

3.2 Basic properties of hierarchical subspaces

Concerning the subspaces W_l, the crucial questions are how important W_l is for the interpolation of some given $u \in X_0^{q,2}(\bar{\Omega})$ and what computational and storage cost come along with it. From Eqs.(3.8) and (3.10), we immediately learn the dimension of W_l, i.e., the number of degrees of freedom (grid points or basis functions, respectively) associated with W_l:

$$|W_l| = |I_l| = 2^{|l-\mathbf{1}|_1}. \tag{3.17}$$

Equation (3.17) already answers the second question.

The following discussion of a subspace's contribution to the overall interpolant according to Eq.(3.16) will be based upon two norms: the maximum norm $\|\cdot\|_\infty$ and the L^p-norm $\|\cdot\|_p$ (p = 2 in general). First we look at the different hierarchical basis functions $\phi_{l,i}(x)$. The following Lemma is simply a straightforward calculation based on the definitions given:

Lemma 3.2.1 *For any piecewise d-linear basis function $\phi_{l,i}(x)$, the following equations hold:*

$$\begin{aligned} \|\phi_{l,i}\|_\infty &= 1, \\ \|\phi_{l,i}\|_p &= \left(\frac{2}{p+1}\right)^{d/p} \cdot 2^{-|l|_1/p},\ p \geq 1. \end{aligned} \tag{3.18}$$

Next, we consider the hierarchical coefficient values $v_{l,i}$ in more detail. They can be computed from the function values $u(x_{l,i})$ in the following way:

$$u_{l,i} = \left(\prod_{j=1}^{n} \left[-\frac{1}{2} \; 1 \; \frac{1}{2} \right]_{x_{l_j,i_j},l_j} \right) =: \left(\prod_{i=1}^{d} I_{x_{l_j,i_j},l_j} \right) u =: I_{x_{l,i},l} u. \tag{3.19}$$

This is due to the definition of the spaces W_l and their basis functions (3.8), whose supports are mutually disjoint and do not contain coarse grid points $x_{k,j}$, $k < l$, in their interior. Definition (3.19) illustrates why $v_{l,i}$ is also called hierarchical surplus. In (3.19), as usual in multigrid terminology, $I_{x_{l,i},l}$ denotes a d-dimensional stencil which gives the coefficients for a linear combination of nodal values of its argument u. This operator-based representation of the hierarchical coefficients $v_{l,i}$ leads to an integral representation of $v_{l,i}$, as follows. The proof of the following three lemmas will also be omitted due to their simplicity and purely computational nature. As always, details can be found in [1].

Lemma 3.2.2 *Let $\psi_{l_j,i_j}(x_j) := -2^{(l_j+1)} \cdot \phi_{l_j,i_j}(x_j)$. Further, let $\prod_{j=1}^{d} \psi_{l,i}(x) := \psi_{l_j,i_j}(x_j)$. For any coefficient value $v_{l,i}$ of the hierarchical representation (3.16) of $u \in X_0^{q,2}(\bar{\Omega})$, the following relation holds:*

$$v_{l,i} = \int_\Omega \psi_{l,i}(x) D^{\mathbf{2}} u(x) dx. \tag{3.20}$$

The above lemma (and its proof) show the close relations of our hierarchical basis approach to integral transforms like wavelet transforms. Applying successive partial integration to (3.20), twice for $d = 1$ and $2d$ times for general dimensionality, we get,

$$v_{l,i} = \int_\Omega \psi_{l,i}(x) D^{\mathbf{2}} u(x) dx \int_\Omega \hat{\psi}_{l,i}(x) u(x) dx, \tag{3.21}$$

where $\hat{\psi}_{l,i}$ equals $D^{\mathbf{2}} \psi_{l,i}(x)$ in a weak sense (i.e., in the sense of distributions) and is a linear combination of 3^d Dirac pulses of alternating sign. Thus, the hierarchical surplus $v_{l,i}$ can be interpreted as the coefficient resulting from an integral transform with respect to a function $\hat{\psi}_{l,i}(x)$ of an oscillating structure.

We are now able to give bounds for the hierarchical coefficients with respect to the different seminorms introduced in Eq.(3.3).

Lemma 3.2.3 *Let $u \in X_0^{q,2}(\bar{\Omega})$ be given in its hierarchical representation. Then, the following estimates for the hierarchical coefficients $v_{l,i}$ hold:*

$$\begin{aligned} |v_{l,i}| &\leq 2^{-d} \cdot 2^{-2|l|_1} \cdot |u|_{\mathbf{2},\infty}, \\ |v_{l,i}| &\leq 2^{-d} \cdot \left(\frac{2}{3} \right)^{d/2} \cdot 2^{-(3/2)|l|_1} \cdot |u|_{supp(\phi_{l,i})|\mathbf{2},2}, \end{aligned} \tag{3.22}$$

where $supp(f)$ denotes the support of the function f.

Finally, the results from the previous three lemmata lead to bounds for the contribution $u_l \in W_l$ of a subspace W_l to the hierarchical representation (3.16) of a given $u \in X_0^{q,2}(\bar{\Omega})$.

Lemma 3.2.4 *Let $u \in X_0^{q,2}(\bar{\Omega})$ be given in its hierarchical representation (3.16). Then, the following estimates for its components $u_l \in W_l$ hold:*

$$\begin{aligned} \|u_l\|_\infty &\leq 2^{-d} \cdot 2^{-2|l|_1} \cdot |u|_{\mathbf{2},\infty}, \\ \|u_l\|_2 &\leq 3^{-d} \cdot 2^{-2|l|_1} \cdot |u|_{\mathbf{2},\infty}. \end{aligned} \tag{3.23}$$

In the next section, the information gathered above will be used to construct finite-dimensional approximation spaces U for V or $X_0^{q,2}(\Omega)$, respectively. Such a U shall be based on a subspace selection $I \subset \mathbb{N}^d$,

$$U := \bigoplus_{l \in I} W_l \tag{3.24}$$

with corresponding interpolants or approximants

$$u_U := \sum_{l \in I} u_l,\ u_l \in W_l. \tag{3.25}$$

The estimate

$$\|u - u_U\| = \|\sum_{l} u_l - \sum_{l \in I} u_l\| \leq \sum_{l \notin I} u_l\| \leq \sum_{l \notin I} b(l) \cdot |u| \tag{3.26}$$

will allow the evaluation of the approximation space U with respect to a norm $\|\cdot\|$ and a corresponding seminorm $|\cdot|$ on the basis of the bounds from above indicating the *benefit* $b(l)$ of W_l.

Summarizing, what this technical section lays is the basis for constructing and deducing formal properties of Sparse Grids, in particular what concerns the approximation accuracy and computational cost.

3.3 Formal derivation and properties of sparse grids

The hierarchical multilevel splitting introduced in the previous section brings along a whole family of hierarchical subspaces W_l of V. However, for discretization purposes, we are more interested in decompositions of finite-dimensional subspaces of V or $X_0^{q,2}(\bar{\Omega})$ than in the splitting of V itself. Therefore, we now turn to finite sets I of active levels l in the summation (3.8). For some $n \in \mathbb{B}$, for instance, one possibility $V_n^{(\infty)}$ has already been mentioned in Eq.(3.14). The finite-dimensional V_n is just the usual space of piecewise d-linear functions on the rectangular grid $\Omega_{(n,\ldots,n)}$ with equidistant mesh size $h_n = 2^{-n}$ in

each coordinate direction, with $|V_n^{(\infty)}| = (2^n - 1)^d = O(2^{d \cdot n}) = O(h_n^{-d})$ inner grid points in the underlying grid.

For the error $u - u_n$ of the interpolant $u_n^{(\infty)} \in V_n^{(\infty)}$ of a given function $u \in X_0^{q,2}(\bar{\Omega})$ with respect to the different norms we are interested in, the following lemma states the respective results.

Lemma 3.3.1 *For $u \in X_0^{q,2}(\bar{\Omega})$, the following estimates hold:*

$$\begin{aligned} \|u - u_n^{(\infty)}\|_\infty &\leq \frac{d}{6^d} \cdot 2^{-2n} \cdot |u|_{\boldsymbol{2},\infty} = O(h_n^2), \\ \|u - u_n^{(\infty)}\|_2 &\leq \frac{d}{9^d} \cdot 2^{-2n} \cdot |u|_{\boldsymbol{2},2} = O(h_n^2). \end{aligned} \tag{3.27}$$

Proof For the L^2-norm, Eq.(3.23) provides

$$\|u - u_n^{(\infty)}\|_2 \leq \sum_{|l|_\infty > n} \|u_l\|_2 \leq 3^{-d} \cdot |u|_{\boldsymbol{2},2} \cdot \sum_{|l|_\infty > n} 2^{-2|l|_1}$$

from which we get

$$\begin{aligned} \|u - u_n^{(\infty)}\|_2 &\leq 3^{-d} \cdot |u|_{\boldsymbol{2},2} \cdot \left(\sum_l 4^{-|l|_1} - \sum_{|l|_\infty \leq n} 4^{-|l|_1} \right) \\ &= 3^{-d} \cdot |u|_{\boldsymbol{2},2} \cdot \left[\left(\frac{1}{3}\right)^d - \left(\sum_{i=1}^{n} 4^{-i} \right)^d \right] \\ &= 3^{-d} \cdot |u|_{\boldsymbol{2},2} \cdot 3^{-d} \cdot (1 - (1 - 4^{-n})^d) \\ &\leq \frac{d}{9^d} \cdot |u|_{\boldsymbol{2},2} \cdot 4^{-n}. \end{aligned}$$

The respective proof for the L^∞-norm is analogous and can be found in [1]. □

It is important to note that we get the same order of accuracy as in standard approximation theory, although our regularity assumptions differ from those normally used there.

This last lemma and the cardinality of $V_n^{(\infty)}$ clearly reveal the crucial drawback of such full grid, the curse of dimensionality discussed in the beggining of this chapter. With d increasing, the number of degrees of freedom that are necessary to achieve an accuracy of $O(h)$ or $O(h^2)$, respectively, grows exponentially. Therefore, we ask how to construct discrete approximation spaces that are better than $V_n^{(\infty)}$ in the sense that the same number of invested grid points leads to a higher order of accuracy. Hence, in the following, we look for an optimum $V^{(opt)}$ by solving a restricted optimization problem of the type

$$\max_{u \in X_0^{q,2}:|u|=1} \|u - u_{V^{(opt)}}\| = \min_{U \subset V:|U|=w} \max_{u \in X_0^{q,2}:|u|=1} \|u - u_U\| \tag{3.28}$$

for some prescribed cost or work count w. The aim is to profit from a given work count as much as possible. A way of interpreting Eq.(3.28) is that maximizing the accuracy of our approximation is the same as minimizing the maximum error (upper bound) admitted for each admissible subspace. Note that an optimization the other way round could be done as well. Prescribe some desired accuracy ε and look for the discrete approximation scheme that achieves this with the smallest work count possible. This is in fact the point of view of computational complexity. Of course, any potential solution $V^{(opt)}$ of (3.28) has to be expected to depend on the norm $\|\cdot\|$ as well as on the seminorm $|\cdot|$ used to measure the error of u's interpolant $u_U \in U$ or the smoothness of u, respectively. According to our hierarchical setting, we will allow discrete spaces of the type $U := \bigoplus_{l \in I} W_l$ for an arbitrary finite index set $I \subset \mathbb{N}^d$ as candidates for the optimization process only. Therefore, spaces U, the respective grids, and the underlying index sets $I \subset \mathbb{N}^d$ have to be identified. There are two obvious ways to tackle such problems: a continuous one based on an analytical approach where the multi-index l is generalized to a nonnegative real one, and a discrete one which uses techniques known from combinatorial optimization. Here, we'll only show the continuous method, while the reader interested in the combinatorial one can find the details in [1].

For the continuous optimization, grid and its representation I (formerly a finite set of multi-indices) is nothing but a bounded subset of $\mathbb{R}_+^d$, and a hierarchical subspace W_l just corresponds to a point $l \in \mathbb{R}_+^d$.

First we have to formulate the optimization problem (3.28). To this end, and inspired by (3.17), the local cost function $c(l)$ is defined as a straightforward generalization of the number of degrees of freedom involved:

$$c(l) := 2^{|l|_1 - d} = 2^{l_1 + \cdots + l_d - d} \tag{3.29}$$

For the local benefit function $b(l)$, we use the squared upper bounds for $\|u_l\|$ according to Eq.(3.23). At the moment, we do not fix the norm to be used here. Obviously, the search for an optimal $I \subset \mathbb{R}_+^d$ can be restricted to $I \subset I^{(max)} := [0, N]^d$ for a sufficiently large N without loss of generality. Based on the two local quantities $c(l)$ and $b(l)$, the *global cost* $C(I)$ and the *global benefit* $B(I)$ of a grid I are defined by

$$C(I) := \int_I c(l)dl, B(I) = \int_I b(l)dl. \tag{3.30}$$

This leads to the desired restricted optimization problem according to (3.28):

$$\max_{C(I)=w} B(I). \tag{3.31}$$

All this means is that we would like to minimize the number of grid points (codified by the cost function) while increasing the accuracy of our approximation (as per the benefit function).

For the solution of (3.31), we start from an arbitrary $I \subset I^{(max)}$ that has a sufficiently smooth boundary ∂I. With a sufficiently smooth mapping τ,

$$\tau : \mathbb{R}^d_+ \to \mathbb{R}^d_+, \tau(l) = 0 \text{ for } l \in \partial\mathbb{R}^d_+ \tag{3.32}$$

we define a small disturbance $\phi_{\epsilon,\tau}$ of the grid I:

$$\phi_{\epsilon,\tau} : I \to I_{\epsilon,\tau} \subset I^{(max)}, \phi_{\epsilon,\tau}(l) := l + \epsilon \cdot \tau(l), \epsilon \in \mathbb{R}. \tag{3.33}$$

For the global cost of the disturbed grid $I_{\epsilon,\tau}$, we get

$$C(I_{\epsilon,\tau}) = \int_{I_{\epsilon,\tau}} c(k)dk = \int_I c(l + \epsilon \cdot \tau(l)) \cdot |\det D\phi_{\epsilon,\tau}|dl. \tag{3.34}$$

Taylor expansion of $c(l + \epsilon \cdot \tau(l))$ in $\epsilon = 0$ provides

$$c(l + \epsilon \cdot \tau(l)) = c(l) + \epsilon \cdot \nabla c(l) \cdot \tau(l) + O(\epsilon^2), \tag{3.35}$$

where $\nabla c(l) \cdot \tau(l)$ denotes the scalar product. Furthermore, a straightforward calculation shows

$$|\det D\phi_{\epsilon,\tau}| = 1 + \epsilon \cdot \text{div}\tau + O(\epsilon^2). \tag{3.36}$$

Thus, since $I \subset I^{(max)}$ with $I^{(max)}$ bounded, Gauss's theorem leads to

$$C(I_{\epsilon,\tau}) = C(I) + \epsilon \cdot \int_{\partial I} c(l) \cdot \tau(l)d\vec{S} + O(\epsilon^2). \tag{3.37}$$

Consequently, for the derivative with respect to ϵ, we get

$$\left.\frac{\partial C(I_{\epsilon,\tau})}{\partial \epsilon}\right|_{\epsilon=0} = \lim_{\epsilon\to 0} \frac{C(I_{\epsilon,\tau}) - C(I)}{\epsilon} = \int_{\partial I} c(l) \cdot \tau(l)d\vec{S}. \tag{3.38}$$

Similar arguments hold for the global benefit $B(I)$ and result in

$$\left.\frac{\partial B(I_{\epsilon,\tau})}{\partial \epsilon}\right|_{\epsilon=0} = \lim_{\epsilon\to 0} \frac{B(I_{\epsilon,\tau}) - B(I)}{\epsilon} = \int_{\partial I} b(l) \cdot \tau(l)d\vec{S}. \tag{3.39}$$

Now, starting from the *optimal* grid $I^{(opt)}$, Lagrange's principle for the optimization under a constraint can be applied, and we get

$$\lambda \cdot \int_{\partial I^{(opt)}} c(l) \cdot \tau(l) d\vec{S} = \int_{\partial I^{(opt)}} b(l) \cdot \tau(l) d\vec{S}. \tag{3.40}$$

Since τ vanishes on the boundary of $\mathbb{R}^d_+$, i.e., $\tau(l) = 0$ when any component of l vanishes, (3.40) is equivalent to

$$\lambda \cdot \int_{\partial I^{(opt)} \setminus \partial \mathbb{R}^d_+} c(l) \cdot \tau(l) d\vec{S} = \int_{\partial I^{(opt)} \setminus \partial \mathbb{R}^d_+} b(l) \cdot \tau(l) d\vec{S}. \tag{3.41}$$

Finally, since (3.41) is valid for all appropriate smooth disturbances τ,

$$\lambda \cdot c(l) = b(l) \tag{3.42}$$

holds for all $l \in \partial I^{(opt)} \setminus \partial \mathbb{R}^d_+$.

This is a quite interesting result, because (3.42) says that the ratio of the local benefit $b(l)$ to the local cost $c(l)$ is constant on the boundary $\partial I^{(opt)} \setminus \partial \mathbb{R}^d_+$ of any grid $I^{(opt)}$ that is optimal in our sense. This means that the *global* optimization process (3.28) or (3.31), respectively, in which we look for an optimal grid can be reduced to studying the *local* cost–benefit ratios $b(l)/c(l)$ of the subspaces associated with l. Therefore, if we come back to real hierarchical subspaces W_l and to indices $l \in \mathbb{N}^d$, all one has to do is to identify sets of subspaces W_l with constant cost–benefit ratio in the subspace scheme. The grid $I^{(opt)}$, then, contains the region where the cost–benefit ratio is bigger than or equal to the constant value on the boundary $\partial I^{(opt)} \setminus \partial \mathbb{R}^d_+$."

3.4 L^2 based sparse grids

Owing to Lemma 3.3.1, the L^2- and L^∞-norm of W_l's contribution u_l to the hierarchical representation (3.8) of $u \in X_0^{q,2}(\Omega)$ are of the same order of magnitude. Therefore there are no differences in the character of the cost–benefit ratio, and the same optimal grids $I^{(opt)}$ will result from the optimization process described above. According to the cost and benefit functions above, we define

$$\mathrm{cbr}_\infty(l) := \frac{b_\infty(l)}{c(l)} := \frac{2^{-4|l|_1} \cdot |u|^2_{2,\infty}}{4^d \cdot 2^{|l-1|_1}} = \frac{1}{2^d} \cdot 2^{-5|l|_1} \cdot |u|^2_{2,\infty} \tag{3.43}$$

$$\mathrm{cbr}_2(l) := \frac{b_2(l)}{c(l)} := \frac{2^{-4|l|_1} \cdot |u|^2_{2,2}}{9^d \cdot 2^{|l-1|_1}} = \left(\frac{2}{9}\right)^d \cdot 2^{-5|l|_1} \cdot |u|^2_{2,\infty} \tag{3.44}$$

as the local cost–benefit ratios. Note that we use bounds for the squared norms of u_l for reasons of simplicity, but without loss of generality. An optimal grid $I^{(opt)}$ will consist of all multi-indices l or their corresponding subspaces W_l where $cbr_\infty(l)$ or $cbr_2(l)$ is bigger than some prescribed threshold $\sigma_\infty(n)$ or $\sigma_2(n)$, respectively. We choose those thresholds to be of the order of $cbr_\infty(l)$ or $cbr_2(l)$ with $\bar{l} := (n, 1, ..., 1)$:

$$\sigma_\infty(n) := \mathrm{cbr}_\infty(\bar{l}) = \frac{1}{2^d} \cdot 2^{-5(n+d-1)} \cdot |u|^2_{2,\infty} \tag{3.45}$$

$$\sigma_2(n) := \mathrm{cbr}_2(\bar{l}) = \left(\frac{2}{9}\right)^d \cdot 2^{-5(n+d-1)} \cdot |u|^2_{2,\infty} \tag{3.46}$$

That is, we fix d subspaces on the axes and search for all W_l whose cost–benefit ratio is equal or better. Thus, applying the criterion $\mathrm{cbr}_\infty(l) \geq \sigma_\infty(n)$ or $cbr_2(l) \geq \sigma_2(n)$, respectively, we get the relation

$$|l|_1 \leq n + d - 1 \tag{3.47}$$

that qualifies a subspace W_l to be taken into account. This result leads us to the definition of a new discrete approximation space $V_n^{(1)}$,

$$V_n^{(1)} := \bigoplus_{|l|_1 \leq n+d-1} W_l, \tag{3.48}$$

which is L^∞- and L^2-optimal with respect to our cost–benefit setting. The grids that correspond to the spaces $V_n^{(1)}$ are just the standard sparse grids. In comparison with the standard full grid space $V_n^{(\infty)}$, we now have triangular or simplicial sectors of subspaces, as seen in Figure 3.4.

Before moving foward, let's stop and interpret what all this math means for our implementation. First, the spaces W_l gives us the functions which compose the numerical approximation. Then, equation (3.47) tells us which set of functions actually count, while equation (3.48) defines the Sparse Grid exactly as the space of (unique) linear combinations of the chosen subspaces. For our implementation purposes, we won't need the actual hat functions, but only the points in the grid they determine (Figure 3.4).

We are finally in position to show how well the Sparse Grid avoids the curse of dimensionality.

Theorem 3.4.1 *The dimension of the space $V_n^{(1)}$, i.e., the number of degrees of freedom or inner grid points, is given by*

$$|V_n^{(1)}| = O(h_n^{-1} \cdot |\log_2 h_n|^{d-1}). \tag{3.49}$$

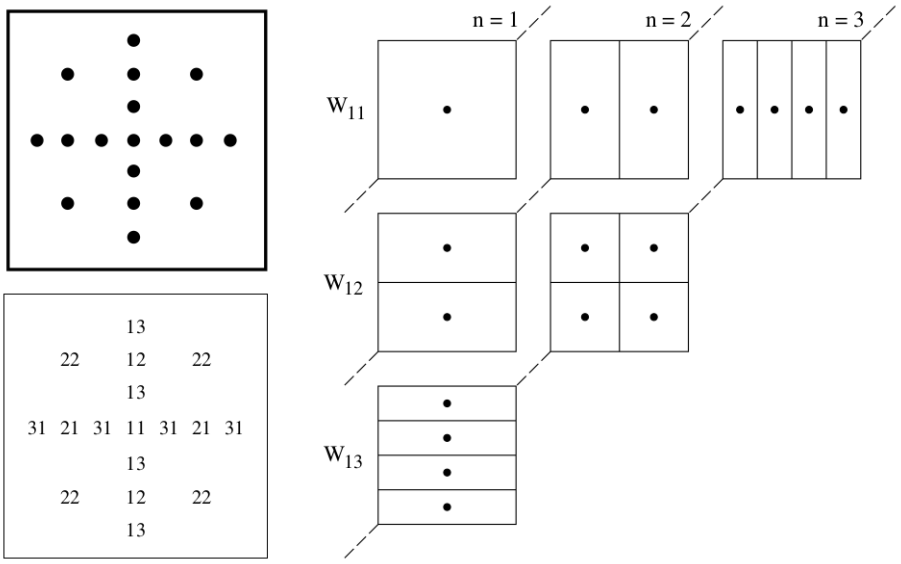

Figure 3.4: The sparse grid of V_3, $d = 2$, and the assignment of grid points to subspaces. Source: Bungartz-Griebel [1]

.

Proof With $|W_l| = 2^{|l-\mathbf{1}|_1}$ and (3.48), we get

$$\begin{aligned} |V_n^{(1)}| = \left| \bigoplus_{|l|_1 \le n+d-1} W_l \right| = \sum_{|l|_1 \le n+d-1} 2^{|l-\mathbf{1}|_1} &= \sum_{i=d}^{n+d-1} 2^{i-d} \cdot \sum_{|l|_1=1} 1 \\ &= \sum_{i=d}^{n+d-1} 2^{i-d} \cdot \binom{i-1}{d-1} \\ &= \sum_{i=0}^{n-1} 2^i \cdot \binom{d-1+i}{d-1} \end{aligned}$$

since there are $\binom{i-1}{d-1}$ ways to form the sum i with d nonnegative integers. Furthermore,

$$\begin{aligned} \sum_{i=0}^{n-1} 2^i \cdot \binom{d-1+i}{d-1} &= \frac{1}{(d-1)!} \cdot \sum_{i=0}^{n-1} (x^{i+d-1})^{(d-1)} \Big|_{x=2} \\ &= \frac{1}{(d-1)!} \cdot (x^{d-1} \cdot \frac{1-x^n}{1-x})^{(d-1)} \Big|_{x=2} \\ &= \frac{1}{(d-1)!} \cdot \sum_{i=0}^{d-1} \Big(d-1\Big) \cdot (x^{d-1} - x^{n+d-1})^{(i)} \cdot \left(\frac{1}{1-x}\right)^{(d-1-i)} \Big|_{x=2} \end{aligned}$$

$$\iff \sum_{i=0}^{n-1} 2^i \cdot \binom{d-1+i}{d-1} = (-1)^d + 2^n \cdot \sum_{i=1}^{d-1} \binom{n+d-1}{i} \cdot (-2)^{d-1-i}$$
$$= 2^n \left(\frac{n^{d-1}}{(d-1)!} + O(n^{d-2}) \right)$$

from which the result concerning the order and the leading coefficient follows. □

The above theorem shows the order $O(2^n \cdot n^{d-1})$ or, with $h_n = 2^{-n}$, $O(h_n^{-1} |\log_2 h_n|^{d-1})$, which is a significant reduction of the number of degrees of freedom and, thus, of the computational and storage requirement compared with the order $O(h_n^{-d})$ of the fullgrid $V_n^{(\infty)}$. This result will allow us to estimate the number of simulations required for the engine blades problem in Chapter 4 (that is, how many times do we call f).

The other question to be discussed concerns the interpolation accuracy that can be obtained on sparse grids. The following Theorem gives the appropriate estimates, and whose highly arithmetic proof can be found in [1].

Theorem 3.4.2 *For the L^∞- and the L^2-norm, we have the following upper bounds for the interpolation error of a function $u \in X_0^{q,2}(\bar{\Omega})$ in the sparse grid space $V_n^{(1)}$:*

$$\|u - u_n^{(1)}\|_\infty = O(h_n^2 \cdot n^{d-1}), \|u - u_n^{(1)}\|_2 = O(h_n^2 \cdot n^{d-1}). \tag{3.50}$$

Theorem 3.4.1 shows the crucial improvement of the sparse grid space $V_n^{(1)}$ in comparison with $V_n^{(\infty)}$ (lemma 3.3.1). The number of degrees of freedom is reduced significantly, whereas the accuracy is only slightly deteriorated (according to theorem 3.4.2). This lessens the curse of dimensionality, but it does not overcome it completely. Since this result is optimal with respect to both the L^∞- and the L^2-norm, a further improvement can only be expected if we change the setting, for instance the optimization process with respect to the energy norm which we do not pursue here and can naturally be found in [1].

Before moving on to the next section, we would like to reference the reader to [13] for an implementation of a generic sparse-grid algorithm in the context of function integration and which help the author developing his own algorithm in the context of the UT in the Octave environment.

3.5 Sparse-grid implementation of the Multidimensional UT

As we suppose the random variables have no correlation, calculating the multidimensional UT would resume simply to a cartesian product of unidimensional transforms. However,

as seen in the motivation for the Sparse Grid method, such approach is computationally unfeasible due to the curse of dimensionality, and we instead reduce the number of (grid) sigma-points necessary by applying such method and making the problem feasible for our application with 16 dimensions.

The implementation consists simply of taking tensorial products of the sigma-points and corresponding weights in each dimension and considering only the chosen sparse grid points instead of the full grid (with the aid of Eq.(3.47)), giving a more feasible computational selection of points while maintaining the probabilistic moments up to some order of precision. For the unidimensional UT, we use the algorithm given at the end of Chapter 2 to determine the sigma-points from a certain probability density function.

To illustrate the grid generated by such algorithm, we consider the probability density function given by a Gaussian distribution, which will be used in the next chapter to model the variability of the blades angles. Inserting two equal bell curves into the UT and applying the sparse grid method for selecting the sigma-points in the plane, we obtain the grid sequence (one for each level, with 5 levels) displayed in Figure 3.5.

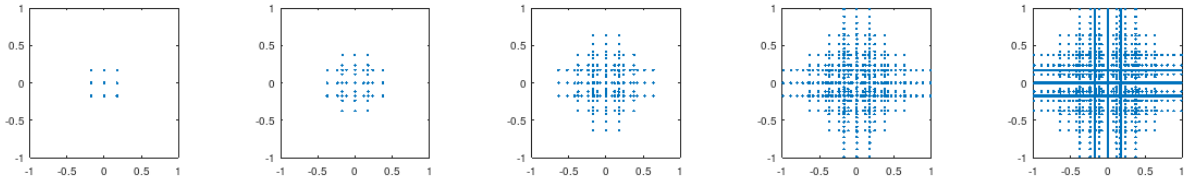

Figure 3.5: Sequence of grid points for the Gaussian distribution in each axis given by the Sparse Grid method. Levels increase from left to right.

The first level was chosen with 3 sigma-points (thus we end up with $3^2 = 9$ points), the second level with 7, and so on with $2^{level} - 1$ sigma-points, stopping at 63 points (5th level).

Recalling Figure 1.1, we were able to input probability density functions and the number of sigma-points desired per level in the Unscented Transform and generate the corresponding set of sigma-points and weights at the output (although we did not plot the weights). We are thus in position to generate the inputs to a model f and obtain information about the system under uncertainty. That is exactly what we are going to do in the next chapter with the case study of supersonic rotor cascades.

Chapter 4

Noise signature evaluation

4.1 Noise generated by supersonic rotor cascades

In nominal operation and under perfect manufacturing conditions, the noise signature produced by supersonic blades is characterized by a single tone at the blade-passing frequency and its harmonics. It is associated with a circumferential pressure profile that resembles a sawtooth pattern, shown in Figs. 4.1.a and 4.1.c [2]. Figure 4.1.a shows the expansion waves as light gray lines while the shock wave is colored dark gray. Notice that the expansion waves for all three blades are identical, as the blades poses the same geometry and same relative angle. Accordingly, Figure 4.1.c displays the pressure sawtooth pressure profile with exact periodicity (that is, $f(t+T) = f(t)$ for all t, where T is the period).

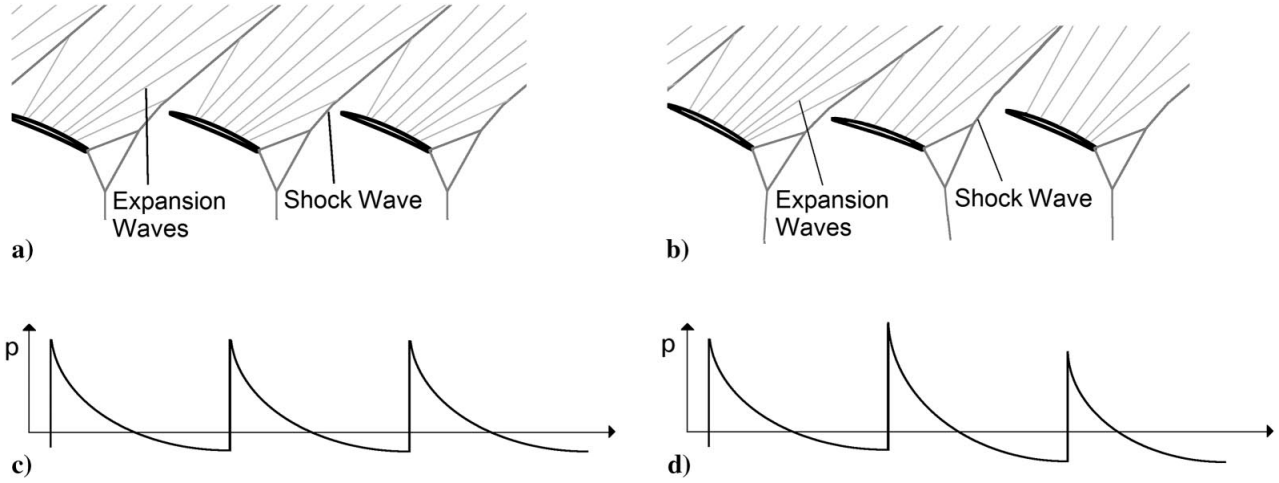

Figure 4.1: Uniform (left) and distorted (right) systems of shock and expansion waves at the rotor tips. Source: Pimenta-Miserda [2].

However, under more realistic conditions, due to manufacturing processes and extended usage, minimal blade-to-blade differences are produced in the stagger angle, camber, thickness, and blade spacing, slightly affecting the system of shock and expansion waves in the rotor blades region, as shown in Figs. 4.1.b and 4.1.d. The upstream propagation of this irregular shock waves system results in the generation of multiple pure tone noise, characterized by spectra clearly dominated by the shaft order and its harmonics [2]. Figure 4.1.b shows divergence of the expansion waves compared to that of Figure 4.1.a due to changes in the blades angles, manifested also in the changes of local maximum and minimum pressures in the profile of Figure 4.1.d compared to 4.1.b.

With the break of the uniform pressure pattern at each blade stage, the fundamental tonal frequency changes from the blade-passing frequency to the shaft rotation frequency, where the resultant pressure pattern repeats only after a full shaft turn. The perceived noise resultant from this interaction is commonly referred to as multiple pure tone (MPT) noise, or buzz-saw noise, for its resemblance to the sound produced by a circular buzz saw. These effects vary even from engines of the same model due to these slight rotor-to-rotor blades differences, making it difficult to obtain a general method to predict this kind of noise for a given turbofan engine [2].

With this in mind, remember that we want to determine as precisely as possible how much variations in the blade angles affect the noise signature of the engine. In this chapter we'll apply the multidimensional UT to generate sigma-points from given continuous noise distributions (e.g. Gaussian noise) and from this generate noise responses from the corresponding modified blade angles of the turbine at specific measuring points. The following illustrates the general algorithm applied in the computations.

First, take as input a probability distribution (P) and the number of sigma-points required (N), one pair (P,N) for each engine blade (16 in total), and the number of levels L that will be constructed by the Sparse Grid. Apply the UT (modified with the Sparse Grid for the multidimensional computation) to this input to obtain as output the sigma-points (s_i, w_i). Now take the sigma-points and modify the initial value pressures $P_{\min}$ and $P_{\max}$ related to the blade angles. Apply the model for each of the modified initial values and obtain a sequence of responses from the model f. Apply the Fast-Fourier Transform (FFT) to obtain a frequency response and detect the dominant tones. Finally, calculate the average of all responses and plot such curve together with error curves corresponding to a number of deviations σ. This algorithm is illustrated in Figure 4.4.

As mentioned throughout the text, we interpret the probability distribution as the possible variation of the blades angles from a certain ideal position (Figure 4.3). However, we do not manipulate such values directly. Instead, we simplify the model and consider only the shock wave pressures at the tip of the blades as indicative of the blade angles,

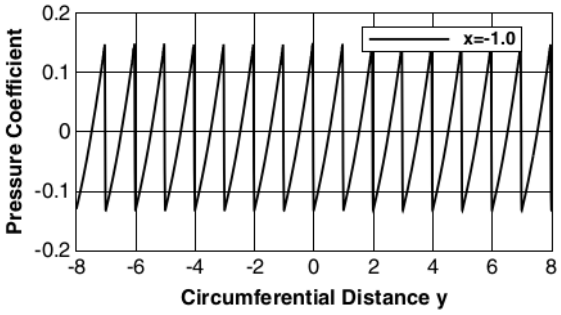

Figure 4.2: Circumferential pressure-coeficient profiles measured 1m from the blades. Source: Pimenta-Miserda [2].

which in [2] resulted from extensive and costly computations (see Figure 4.2). Then, to transform the sigma-points to pressure, we take the formula $\Delta p = (1 - s)\Delta^{(max)}p$, where we take $\Delta^{(max)}p := 25612\text{Pa}$ as the maximum amplitude $p_{max} - p_{min}$ in Braulio's [2] code (so in our model we actually get twice the amplitude with null probability).

Remember that, although the input involves a random variable, the algorithm is **deterministic**. Thus, to obtain different results (for the purpose of testing, for example), alterations must be made to the probabilistic distribution which is given as input, as for each such distribution (and set of parameters) the sigma points will be uniquely determined.

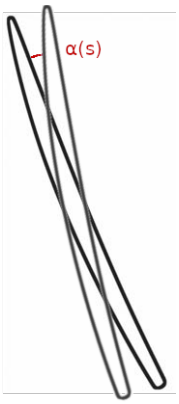

Figure 4.3: Alteration of the blade angle α in function of the sigma-point s.

4.2 Simulation

Programmed and executed in Octave 6.0.1, the simulation presented here was conducted in a sixteen dimensional space (one dimension for each blade of the engine), received as input 3 levels for the sparse grid computation, and $N = 1, 3, 5$ for the number of sigma-points in each dimension (one N for each level). All probability density functions were

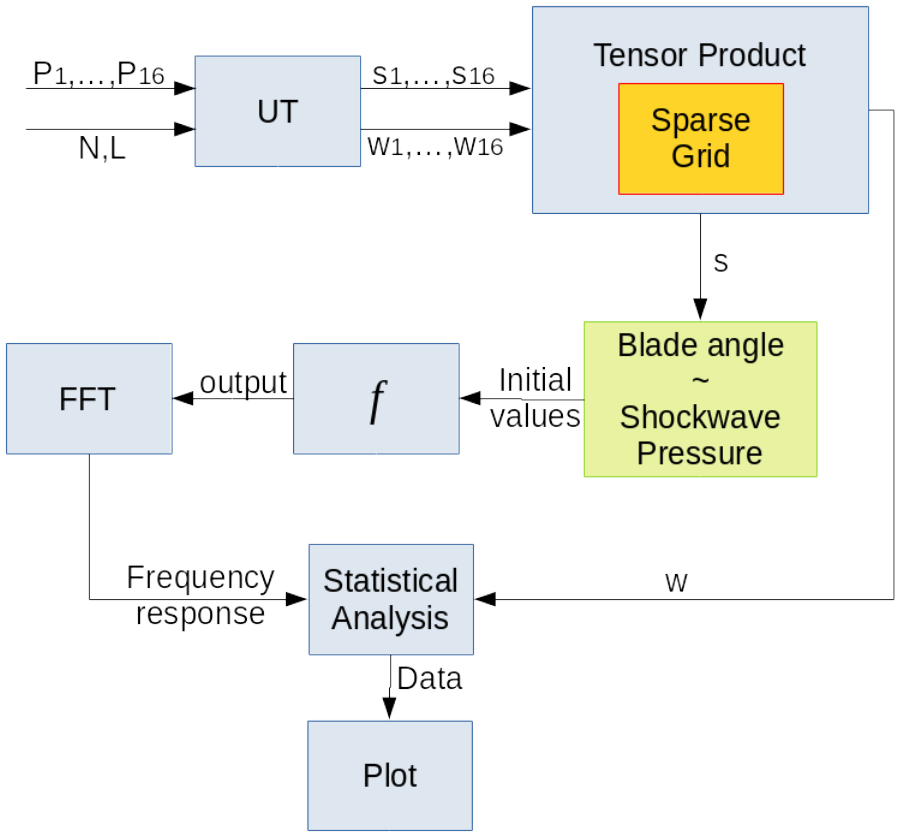

Figure 4.4: Schematic depiction of the simulation algorithm for determination of the noise signature. N in the input is the number of sigma-points desired and L is the number of levels in the sparse grid approximation.

normal distributions [9]

$$x \mapsto \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{4.1}$$

with $\sigma = 0.1$ and $\mu = 0$ and a code generously supplied by professor Braulio Pimenta (ENM-Universidade de Brasília) was used as the model f for the pressure computations. The whole simulation took approximately 30 minutes in a GNU/Linux machine with an Intel i5 processor @ 2.5Ghz and 8GB of RAM. A total of sigma-points (16-tuples) in the order of $8 \cdot 10^4$ were evaluated after the Sparse-Grid computation, which is close to the estimated value given by Theorem 3.1.6 of 69 121 grid points (a difference which the author believes is justified, at least in part, by the inclusion of boundary points which were not taken into account in the mentioned theorem). Figure 4.5 shows the mean output of f before entering the Fast-Fourier Transform.

Figure 4.6 shows in blue the modulus of the mean response obtained after the Fast-Fourier Transform according to a given frequency, while the grey lines show the limits of error according the two standard deviations, i.e. 2σ. A clear peak can be noticed at the left of the graph, while the remaining plot has a clear central symmetry. We

believe this geometric characteristic is due to the way the pressures were calculated from the sigma-points, where the expression for $p_{\max}$ and $p_{\min}$ are highly symmetrical (one simply adds or subtracsts Δp from a fixed mean value p_0). The probability weights were applied pointwise for the graph construction, that is, the median m was calculated by $m = \sum_i w_i \times (\text{corresponding } s_i \text{ FFT output}) / \sum_i w_i$ in each point.

From this simulation, one observes that the techniques work and are computationally feasible, specially in a well equipped computational environment (which was not the case), while a high level sparse grid still demands a lot of memory and computational time. In fact, the case $L = 4$ levels showed clearly that the curse of dimensionality was not completely overcome, consuming more than 10GB of memory and taking more than an hour before being manually stopped (the machine only had 8GB of memory, so a lot of disk space was being used as virtual memory and constantly being accessed, rendering the machine unresponsive).

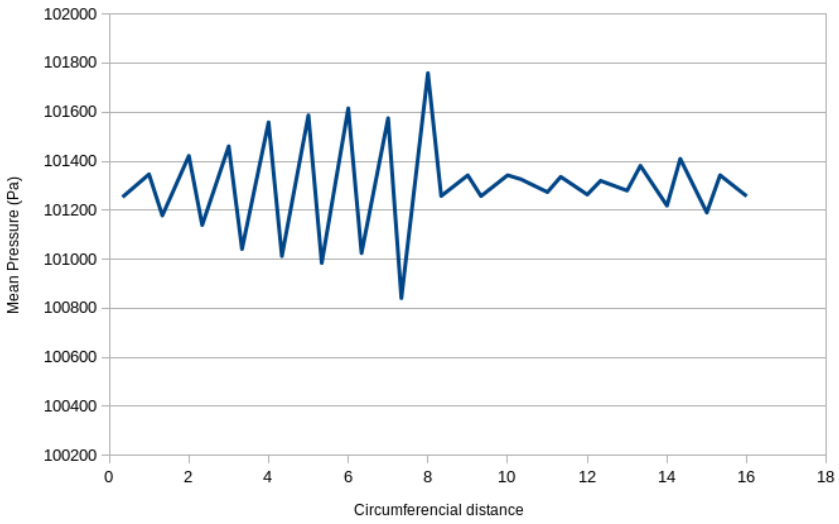

Figure 4.5: Mean pressure output of the model f for the determined sigma-points.

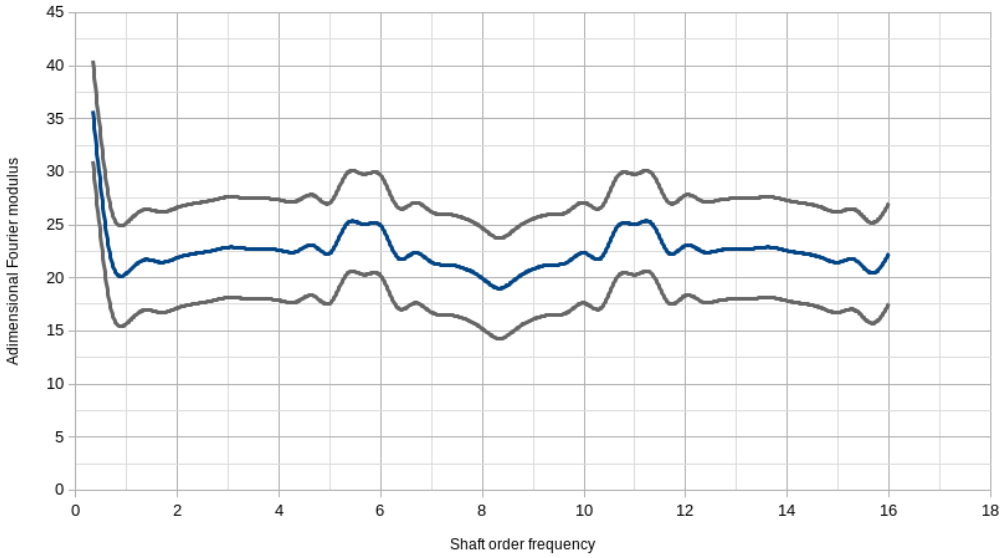

Figure 4.6: Graph of Pressure vs. Frequency giving the mean noise signature in blue and corresponding tolerance band in gray according to two standard deviations.

Chapter 5

Conclusion

This work systematically exposed the fundamentals of the Unscented Transform and the Sparse Grid approach to numerical integration, which was then applied to the characterization of the noise signature of supersonic rotor cascades under simplifying assumptions. The algorithm ran as expected in a reasonable time (relatively speaking compared to typical simulations in the area) and resulted in a response within the expected, however unrealistic from a modeling point of view. As far as the author knows, this is the first time such technique is used in this context. This work thus gives a step foward from the work done by de Medeiros in [3], where the domain of applicability of the UT did not go beyond the third dimension and the calculations were done with full grid, with the UT being an alternative to Monte Carlo methods.

As surely grasped by the reader, this work is but a initial step in the application of the Unscented Transform in the aeroacoustic realm of applications. This axiomatically implies a great number of possible paths from here, which we specify a few in the section.

5.1 Future work

The weakest point of the overall simulation was the conversion of the sigma-points calculated after the tensor product to a pressure profile at blades, represented by the green block in Figure 4.2. Due to the shorter semester caused by the pandemic during the preparation of this work, the author was unable to learn and interact more with Mechanical Engineers so as to determine a faithful or realistic representation of the shockwave pressures, as well as establish the relation between the blade angles and such shockwave pattern. This is certainly a point to be worked on and that will prove invaluable for applications, as it will hopefully fill a gap in the Aeroacoustic toolbox.

As the technique described here is quite general, it surely can be used with few modifications to other applications and contexts, specially those afflicted by uncertainty in high-dimensions, for instance economic or physical models [5].

It should be pointed out that, besides the Sparse Grid method, no other optimization technique was explored, e.g. use of parallelism. Given the scope of application (high-dimensional problems), implementation of optimized algorithms seem mandatory to show the full potential of the techniques presented.

A number of simplifying hypothesis were used throughout this work, in particular our assumption that the random variables are independent. Our approach here is not directly applicable to the dependent case and poses interesting and challenging questions for future research.

The literature concerning the Unscented Transform focus mostly on conservation of at most 3 probabilistic moments [11, 4, 8], while our definition of the UT clearly indicates, at least on the mathematical side, that such bound is artificial. There seems to be room for both theoretical and practical developments concerning higher-order UTs, even at the foundational level.

Finally, in the introduction we mentioned a key motivation for the use of the UT as the possibility of obtaining a solution to the design problem (the inverse problem). Time did not permit us to pursue this quest further, but it is a highly ambitious goal with manifold applications in many areas of Engineering, if the proposal indeed works.

References

[1] Bungartz, Hans Joachim and Michael Griebel: *Sparse grids*. Acta Numerica, 13:147–269, 2004. vii, 11, 12, 14, 15, 16, 17, 19, 20, 24, 25

[2] Pimenta, Braulio Gutierrez and Roberto F. Bobenrieth Miserda: *Numerical simulation of multiple pure tone noise generated by supersonic rotor cascades*. Journal of propulsion and power, 35(3):1–14, 2019. vii, 1, 4, 27, 28, 29

[3] Medeiros, José Edil Guimarães de: *Unscented Transform Framework for Quantization Modeling in Data Conversion Systems*. PhD thesis, Faculdade de Tecnologia - Universidade de Brasília, 2017. 1, 2, 5, 6, 7, 9, 33

[4] Julier, Simon J. and Jeffrey K. Uhlmann: *Unscented filtering and nonlinear estimation*. Proceedings of the IEEE, 92(3):401–422, 2004. 4, 5, 34

[5] Kalos, Marvin H. and Paula A. Whitlock: *Monte Carlo Methods*. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, 2nd edition, 2008. 5, 34

[6] Menezes, L. D.: *Using unscented transform as alternative to monte carlo in bit error rate calculations*. Electronic Letters, 49(10):1, 2013. 6

[7] Hämmerlin, Günther and Karl Heinz Hoffmann: *Numerical Mathematics*. Springer-Verlag, New York, 1991. 6, 11

[8] Menegaz, Henrique M. T., João Y. Ishihara, Geovany A. Borges, and Alessandro N. Vargas: *A systematization of the unscented kalman filter theory*. IEEE transactions on automatic control, 60(10):2583–2598, 2015. 9, 10, 34

[9] James, Barry R.: *Probabilidade: um curso em nível intermediário*. IMPA, Rio de Janeiro, Brasil, 3rd edition, 2011. 9, 10, 30

[10] Ito, Kazufumi and Kaiqi Xiong: *Gaussian filters for nonlinear filtering problems*. IEEE Trans. Autom. Control, 45(5):910–927, 2000. 9

[11] Arasaratnam, Ienkaran and Simon Haykin: *Cubature kalman filters*. IEEE Trans. Autom. Control, 54(6):1254–1269, 2009. 10, 34

[12] Brezis, Haim: *Functional Analysis, Sobolev Spaces and Partial Differential Equations*. Springer, 2011. 11

[13] Kaarnioja, Vesa: *Smolyak quadrature*. Master's thesis, Faculty of Science - University of Helsinki, 2013. 25