Sampling Methods

In this lecture note we discuss various sampling methods commonly used to propagate uncertainty in numerical simulations of nonlinear systems. Such samples can be randomly generated (using pseudo-random number generators) as in Monte Carlo methods [7], or can be part of deterministic sequences as in quasi-Monte Carlo methods [7, 4], sparse grids [3], or probabilistic collocation methods [10, 5].

The most appropriate sampling scheme depends on the problem at hand, in particular on the number and the nature of random variables driving the system. For instance, if we are interested in approximating the expectation of a quantity of interest in a system that depends only on one random variable $\xi(\omega)$ with bounded range then perhaps Monte-Carlo is not the most efficient method. Indeed, in this case, it is rather straightforward to derive a Gauss quadrature rule with a high degree of exactness approximate the expectation as

$$\mathbb{E}\left\{h(\xi)\right\} = \int_a^b h(x) p_\xi(x) dx \simeq \sum_{k=1}^N h(\xi_k) w_k. \tag{1}$$

If the function h of class C^∞, then the Gauss quadrature rule (1) converges exponentially fast with N (number of samples). On the other hand, the convergence rate of the Monte Carlo method is $1/\sqrt{N}$. If the system is driven by high-dimensional random input vector then Gauss quadrature on tensor product grids is not viable, we are are often left with no choice other than random sampling.

A distinctive advantage of sampling methods is that they are *non-intrusive*. This means that they do not require devising problem-dependent equations or writing new codes and algorithms from scratch perform UQ analyses, but rather simply run existing legacy algorithms and codes many times, eventually in a massively parallel way.

Monte Carlo (MC)

Monte Carlo methods are a broad class of computational algorithms that rely on repeated *random sampling* to obtain numerical results of various types, e.g., estimation of high-dimensional PDFs or approximation of high-dimensional integrals representing, e.g., expectation operators.

Example (PDF estimation): Suppose we are interested in estimating the PDF of a random variable Y depending on three random variables X_1, X_2 and X_3. We are given joint PDF of (X_1, X_2, X_3), i.e., $p(x_1, x_2, x_3)$ and the mapping

$$Y = g(\boldsymbol{X}). \tag{2}$$

In a (Markov-Chain) Monte-Carlo setting the estimation of the PDF Y proceeds as follows:

1. Determine N samples of $p(x_1, x_2, x_3)$, e.g., using Gibbs sampling (see Chapter 1). This yields $\left\{\boldsymbol{X}^{[1]}, \ldots, \boldsymbol{X}^{[N]}\right\}$;
2. Compute N samples of Y using (2), i.e., $Y^{[j]} = g\left(\boldsymbol{X}^{[j]}\right)$;
3. Estimate the joint PDF of $Y(\omega)$ using relative frequencies, or kernel density estimation [2].

Example (Expectation operator): Suppose we are interested in approximating the mean of scalar phase space function of interest $h(\boldsymbol{x})$ depending on the solution of the ODE system

$$\begin{cases} \dfrac{d\boldsymbol{x}}{dt} = \boldsymbol{G}(\boldsymbol{x}, \boldsymbol{\xi}(\omega), t) \\ \boldsymbol{x}(0;\omega) = \boldsymbol{x}_0 \end{cases} \tag{3}$$

where $\boldsymbol{x}_0$ is deterministic and $\boldsymbol{\xi}$ is a random vector. The expectation of $h(\boldsymbol{x}(t;\omega))$ can be written as

$$\mathbb{E}\left\{h(\boldsymbol{x}(t;\omega))\right\} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} h\left(\boldsymbol{x}\left(t;\boldsymbol{y}\right)\right) p_{\boldsymbol{\xi}}(\boldsymbol{y}) d\boldsymbol{y}, \tag{4}$$

i.e., as a high-dimensional integral over the PDF of $\boldsymbol{\xi}$. In a Monte-Carlo setting, such an integral is approximated by an *equal-weight* quadrature formula of the form

$$\mathbb{E}\left\{h(\boldsymbol{x}(t;\omega))\right\} \simeq \frac{1}{N}\sum_{k=1}^{N} h\left(\boldsymbol{x}\left(t;\boldsymbol{\xi}^{[k]}\right)\right), \tag{5}$$

where $\{\boldsymbol{\xi}^{[1]},\ldots,\boldsymbol{\xi}^{[N]}\}$ are independent random samples obtained from $p(\boldsymbol{\xi})$ using, e.g., Gibbs sampling.

Similarly, Monte Carlo can be used to obtain response samples of random eigenvalue problems (random eigenvalues and random eigenvectors), solution to PDE, etc.

Monte Carlo integration. Consider the following mapping between an n-dimensional random vector $\boldsymbol{X}$ and a random variable Y

$$Y = g(\boldsymbol{X}). \tag{6}$$

Here g is a measurable map from $\mathbb{R}^n$ into $\mathbb{R}$. We are interested in computing

$$\mathbb{E}\{Y\} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} g(\boldsymbol{x}) p_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x}. \tag{7}$$

To this end, we draw N independent random samples from the random vector $\boldsymbol{X}$, denoted as $\left\{\boldsymbol{X}^{[1]},\ldots,\boldsymbol{X}^{[N]}\right\}$, and approximate the integral at the right hand side of (7) as

$$\mathbb{E}\{Y\} \simeq \frac{1}{N}\sum_{k=1}^{N} g\left(\boldsymbol{X}^{[k]}\right). \tag{8}$$

The following error bound then holds true.

Theorem 1. For all functions $g \in L^2_{p_{\boldsymbol{X}}}(\mathbb{R}^n)$, i.e., for all random variables $Y = g(\boldsymbol{X})$ with finite second-order moment we have

$$\widehat{\mathbb{E}}\left\{\left|\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} g(\boldsymbol{x}) p_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x} - \frac{1}{N}\sum_{k=1}^{N} g\left(\boldsymbol{X}^{[k]}\right)\right|^2\right\} = \frac{\sigma^2(g)}{N}, \tag{9}$$

where $\widehat{\mathbb{E}}$ is an expectation of the joint PDF of $\left\{\boldsymbol{X}^{[1]},\ldots,\boldsymbol{X}^{[N]}\right\}$ (treated as independent random vectors), and

$$\sigma^2(g) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} g^2(\boldsymbol{x}) p_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x} - \left(\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} g(\boldsymbol{x}) p_{\boldsymbol{X}}(\boldsymbol{x}) d\boldsymbol{x}\right)^2 \tag{10}$$

is the variance of $g(\boldsymbol{X})$, i.e., the variance of Y in (6).

The proof of this theorem is available in [4, 7] and therefore omitted here. Note that if we approximate (7) using (8) and different sets of samples $\{\boldsymbol{X}^{[1]},\ldots,\boldsymbol{X}^{[N]}\}$ then we obtain different results. Hence, we should really think of (8) as a sum of independent random variables $\boldsymbol{X}^{[k]}$, each one of which is distributed as $p_{\boldsymbol{X}}(\boldsymbol{x})$. This means that the right hand side of (8) can be thought of as a sum of *independent random variables*.

By using the central limit theorem[1] it is straightforward to obtain the following probabilistic error bound on the MC approximation (8)

$$\lim_{N\to\infty} P\left(\left\{\omega : \left|\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} g(\boldsymbol{x})p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x} - \frac{1}{N}\sum_{k=1}^{N} g\left(\boldsymbol{X}^{[k]}(\omega)\right)\right| \leq c\frac{\sigma(g)}{\sqrt{N}}\right\}\right) = \frac{1}{\sqrt{2\pi}}\int_{-c}^{c} e^{-y^2/2}dy. \quad (14)$$

To this end, we simply substitute $Y^{[k]} = g\left(\boldsymbol{X}^{[k]}\right)$, $\sigma^2 = \sigma^2(g)$ and

$$m = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} g(\boldsymbol{x})p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x}. \quad (15)$$

into (13).

Equation (14) is an asymptotic probabilistic error bound stating that as we increase the number of samples the MC approximation goes to zero as[2] $1/\sqrt{N}$. Note that (14) is *independent of the dimension of the integral* (dimension of the vector $\boldsymbol{x}$), which is a great deal that makes MC suitable for high-dimensional integration. Similarly, by using the Markov inequality, it follows from (9) that

$$P\left(\left\{\omega : \left|\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} g(\boldsymbol{x})p_{\boldsymbol{X}}(\boldsymbol{x})d\boldsymbol{x} - \frac{1}{N}\sum_{k=1}^{N} g\left(\boldsymbol{X}^{[k]}(\omega)\right)\right| \geq \epsilon\right\}\right) \leq \frac{\sigma(g)}{\epsilon\sqrt{N}}. \quad (16)$$

Remark: While independent of the dimension of the integral, the convergence rate $O(N^{-1/2})$ of the Monte Carlo approximation (8) is not too great. Roughly speaking, to obtain a one digit increase in accuracy we need 100 times more samples! To show this Let E_1 be the integration error. We know that E_1 is proportional to $N_1^{-1/2}$, i.e.,

$$E_1 = CN_1^{-1/2}. \quad (17)$$

To obtain an error $E_2 = E_1/10$, i.e., gain one digit accuracy, we need

$$CN_2^{-1/2} = CN_1^{-1/2}/10 \quad \Rightarrow \quad N_2 = 100N_1. \quad (18)$$

Hence, if we get the first two digits of our integral right with an MC formula involving 5000 random samples, then we would need roughly 500000 samples to get third digit right!

As an example, in Figure 1 we compare the error in the numerical approximation of the integral

$$I(f) = \int_{-1}^{1} g(x)dx, \qquad g(x) = e^{-x}x^2\sin(10x)^2 \quad (19)$$

[1]The central limit theorem can be stated as follows: let $\{Y^{[1]},\ldots,Y^{[N]}\}$ be a sequence of i.i.d. random variables with mean m and variance σ^2. Define

$$Z_N = \sqrt{N}\left(\frac{1}{N}\sum_{k=1}^{N} Y^{[k]} - m\right). \quad (11)$$

Then the PDF of Z_N converges to a normal distribution with zero mean and variance σ^2, i.e.,

$$\lim_{N\to\infty} p_{Z_N}(x) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-x^2/(2\sigma^2)}. \quad (12)$$

This means that

$$\lim_{N\to\infty} P\left(\{\omega : |Z_N(\omega)| \leq c\sigma\}\right) = \frac{1}{\sqrt{2\pi\sigma^2}}\int_{-c\sigma}^{c\sigma} e^{-x^2/(2\sigma^2)}dx = \frac{1}{\sqrt{2\pi}}\int_{-c}^{c} e^{-y^2/2}dy. \quad (13)$$

[2]Simply set $c = 3$ in (14) to obtain the right hand side approximately equal to 1.

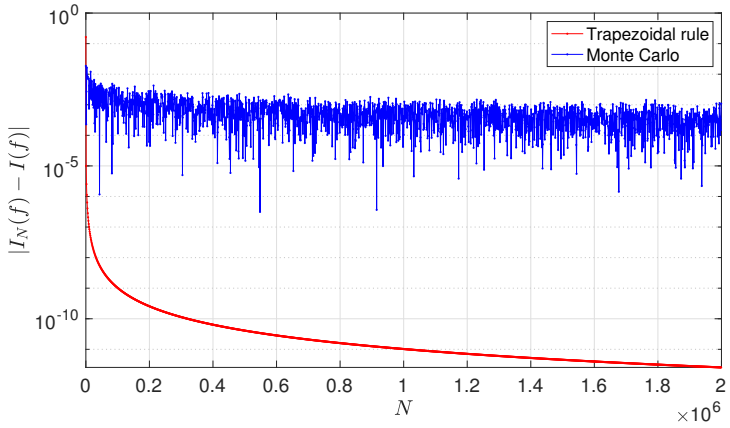


Figure 1: Error in the numerical approximation of the integral (19) using the Monte Carlo rule (8) and the trapezoidal rule versus the number of points N.

using Monte Carlo and the trapezoidal rule. For Monte Carlo, we simply compute N independent samples of a uniform random variable X in $[-1, 1]$ and compute the sum

$$I_N(f) = \frac{2}{N} \sum_{k=1}^{N} g(X^{[k]}). \tag{20}$$

The factor 2 accounts for the fact that the PDF of a uniform variable in $[-1, 1]$ is $1/2$. Note that on average the convergence rate of MC is $1/\sqrt{N}$ while the converge rate of the trapezoidal rule is $1/N^2$.

Quasi Monte Carlo (QMC)

Just like Monte Carlo, quasi-Monte Carlo methods [7, 4] aim at representing multidimensional integrals of the form (7) as equal-weights quadrature rules (8). However, in QMC the sequence of points $\boldsymbol{X}^{[k]}$ are not realizations of a random vector, but rather elements of a *deterministic sequence* called low-discrepancy sequence. The whole point such low-discrepancy sequences is to improve the (very) slow convergence rate of MC (i.e., $O(N^{-1/2})$) when evaluating multidimensional integrals of the form

$$I(g) = \int_{[0,1]^n} g(\boldsymbol{x}) d\boldsymbol{x}. \tag{21}$$

QMC methods are usually classified based on how the points in the low-discrepancy sequences are computed. In particular, we can have sequences of points that can be increased without recomputing the first few points (open QMC formulas), and sequences of points that require recalculation of all points if N changes (closed QMC formulas) [4]. Hereafter we provide a two examples of QMC rules leveraging the radical inverse function.

Radical inverse function. Let $b \geq 2$ be a natural number. As is well known, any integer number can be represented relative to the base b as

$$i = \sum_{k=1}^{\infty} i_k b^{k-1} \tag{22}$$

where i_k can take values in $\{0, 1, \ldots, b-1\}$. For example, the number 11 can be written in base 2 and

base 3 as c

$$11 = 1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 1 \times 2^3 = [\cdots 01011]_2, \tag{23}$$
$$= 2 \times 3^0 + 0 \times 3^1 + 1 \times 3^2 = [\cdots 0102]_3. \tag{24}$$

We define the radical inverse function corresponding to an integer number $i \in \mathbb{N}_0$

$$\phi_b(i) = \sum_{k=1}^{\infty} \frac{i_k}{b^k}. \tag{25}$$

The function (25) operates as follows:

$$i = [\cdots i_3 i_2 i_1]_b \qquad \Rightarrow \qquad \phi_b(i) = [0.i_1 i_2 i_3 \cdots]_b. \tag{26}$$

With reference to (23)-(24) we have, for example,

$$\phi_2(11) = 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-4} = \frac{1}{2} + \frac{1}{4} + \frac{1}{16} = \frac{13}{16}, \tag{27}$$
$$\phi_3(11) = 2 \times 3^{-1} + 1 \times 3^{-3} = \frac{2}{3} + 1\frac{1}{27} = \frac{19}{27}. \tag{28}$$

Halton's sequence. The Halton's sequence is a point set in the hypercube $[0,1]^n$ defined as

$$\boldsymbol{X}_{Hl}^{[i]} = (\phi_{p_1}(i), \ldots, \phi_{p_n}(i)) \qquad i = 1, 2, \ldots, N, \tag{29}$$

where $\{p_1, \ldots, p_n\}$ are the first n prime numbers, and $\phi_{p_j}(i)$ is the radical inverse function (25). For example, in dimension $n = 5$ we have

$$\boldsymbol{X}_{Hl}^{[i]} = (\phi_2(i), \phi_3(i), \phi_5(i), \phi_7(i), \phi_{11}(i)) \qquad i = 1, 2, \ldots, N. \tag{30}$$

By using the Halton's sequence we can approximate integrals relative to uniform PDFs in $[0,1]^n$ as

$$\int_{[0,1]^n} g(\boldsymbol{x}) d\boldsymbol{x} \simeq \frac{1}{N} \sum_{k=1}^{N} g\left(\boldsymbol{X}_{Hl}^{[k]}\right). \tag{31}$$

It can be shown that (see [4])

$$\left| \int_{[0,1]^n} g(\boldsymbol{x}) d\boldsymbol{x} - \frac{1}{N} \sum_{k=1}^{N} g\left(\boldsymbol{X}_{Hl}^{[k]}\right) \right| \leq C_n \frac{(\log(N))^n}{N} V_{HK}(g), \tag{32}$$

where $V_{HK}(g)$ is the variation of $g(\boldsymbol{x})$ in the sense of Hardy and Krause (see [4]). For fixed g we have that $V_{HK}(g)$ is a number depending only on g. The function $(\log(N))^n / N$ defining the upper bound in (32) has an asymptote at $N = 0$, a minimum at $N = 1$ (equal to zero), a maximum at $N = e^n$ (equal to $(n/e)^n$), and goes to zero faster than $N^{-1/2}$ as N goes to infinity (see Figure 2. In dimension $n = 10$ we have $e^n = 22026$. Hence, to go past the "hump" in dimension $n = 10$ we need $N > 22026$ samples.

Hammersley's sequence. The Hammersley's sequence is a point set in the hypercube $[0,1]^n$ defined as

$$\boldsymbol{X}_{Hm}^{[i]} = \left(\frac{i}{N}, \phi_{p_1}(i), \ldots, \phi_{p_n}(i)\right) \qquad i = 1, 2, \ldots, N-1 \tag{33}$$

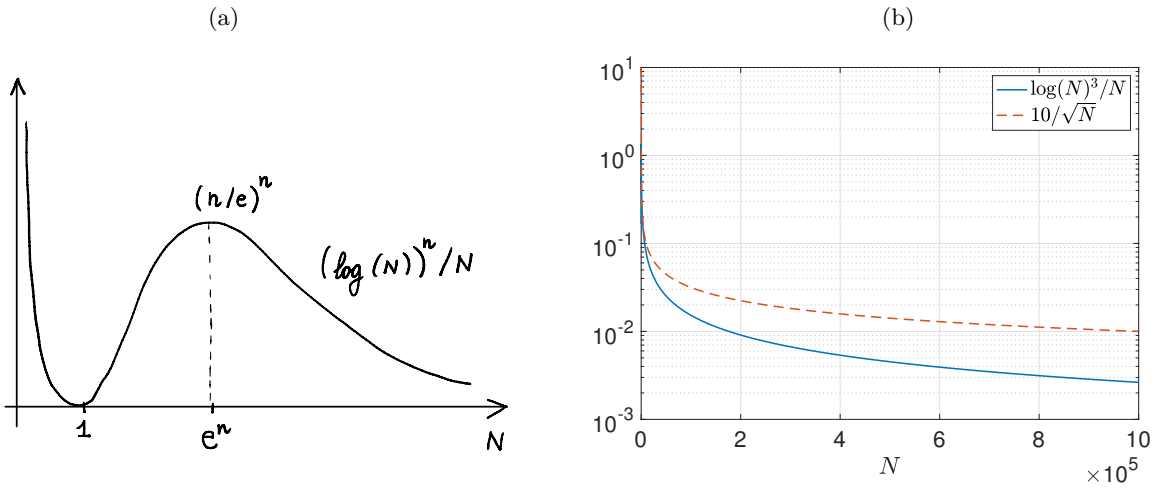


Figure 2: (a) Sketch of the upper bound of the approximation error of the quasi-Monte Carlo quadradure rule based on the Halton's sequence applied to an n-dimensional integral versus the number of samples N. (b) Comparison between the decay rate for of Halton QMC and MC for $n = 3$.

where $\{p_1, \ldots, p_n\}$ are the first n prime numbers, and $\phi_{p_j}(i)$ is the radical inverse function (25). Note that the first column in (33) needs to be recomputed if we change N. It can be shown (e.g., [4]) that

$$\left| \int_{[0,1]^n} g(\boldsymbol{x}) d\boldsymbol{x} - \frac{1}{N} \sum_{k=1}^{N} g\left(\boldsymbol{X}_{Hm}^{[i]} \right) \right| \leq C_n \frac{(\log(N))^{n-1}}{N} V_{HK}(g), \tag{34}$$

which represents a slight improvement over (32).

Remark: To further improve the convergence rate of quasi-Monte Carlo one can introduce *randomizations* of the QMC point sets, e.g., in the form of *random shifts* or *point scrambling*. The randomization allows us derive probabilistic error bounds similar to MC, and at the same time can improve the convergence rate of QMC (see [7, 4]).

Probabilistic collocation method (PCM)

The probabilistic collocation method is a high-order method based on deterministic point sets that allows us to compute expectation operators involving low-dimensional integrals. The method leverages high-order interpolatory quadrature rules [9], in particular, Gaussian quadrature.

We have seen in previous lecture that orthogonal polynomials play a fundamental role in the approximation of smooth functions. As we shall see hereafter, orthogonal polynomials play also a crucial role in devising interpolants and quadrature formulae with maximal *degrees of exactness*[3]. These formulae are known as Gaussian quadrature formulae [9, §10.2].

To introduce Gaussian quadrature in the context of UQ, suppose we are given a random variable X with range $[a, b]$ and PDF $p_X(x)$. We have seen that for every measurable function $g : [a, b] \to \mathbb{R}$ the expectation of $g(X)$ is defined as

$$\mathbb{E}\{g(X)\} = \int_a^b g(x) p_X(x) dx. \tag{35}$$

[3]The degree of exactness of a quadrature formula is the maximum degree of the polynomial that can be integrated exactly by the formula. In other words, we say that a quadrature formula has degree of exactness p if it can integrate exactly polynomials of degree p or less.

By using the coordinate transformation

$$x = \frac{b-a}{2}z + \frac{b+a}{2} \qquad z = \frac{2}{b-a}\left(x - \frac{b+a}{2}\right) \qquad z \in [-1,1] \tag{36}$$

we can rewrite the expectation in (35) as

$$\int_a^b g(x)p_X(x)dx = \frac{b-a}{2}\int_{-1}^1 f(z)\mu(z)dz, \tag{37}$$

where

$$f(z) = g\left(\frac{b-a}{2}z + \frac{b+a}{2}\right), \qquad \mu(z) = p_X\left(\frac{b-a}{2}z + \frac{b+a}{2}\right). \tag{38}$$

For the approximation of the weighted integral at the right hand side of (37), we consider the quadrature rule

$$\int_{-1}^1 f(z)\mu(z)dz \simeq \sum_{k=0}^M f(z_k)w_k, \tag{39}$$

where $\{z_0,\ldots,z_M\}$ are quadrature points in $[-1,1]$ while $\{w_0,\ldots,w_M\}$ are quadrature weights.

If we approximate $f(z)$ by the Lagrange interpolation polynomial $\Pi_M f(z)$ at the $M+1$ nodes $\{z_0,\ldots,z_M\}$ then (39) is a quadrature formula that has degrees of exactness at least equal to M, and explicit expression for the quadrature weights w_k. This follows from

$$\int_{-1}^1 f(z)\mu(z)dz \simeq \int_{-1}^1 \Pi_M f(z)\mu(z)dz = \sum_{k=0}^M f(z_k)\underbrace{\int_{-1}^1 l_k(z)\mu(z)dz}_{w_k}, \tag{40}$$

where

$$l_k(z) = \prod_{\substack{j=0\\ j\neq k}}^M \frac{z-z_j}{z_k-z_j} \tag{41}$$

are the Lagrange characteristic polynomial associated with the grid $\{z_0,\ldots,z_M\}$.

At this point the question is whether suitable choices of the nodes exist such that the degree of exactness is greater than M, say, equal to $r = M+m$ for some $m > 0$. The answer is given by the following theorem

Theorem 2 (Gaussian quadrature - Jacobi's theorem)**.** For any given $m > 0$ the interpolatory quadrature rule (39) has degree of exactness $M+m$ if and only if the polynomial

$$q_{M+1}(z) = \prod_{j=0}^M (z - z_j) \tag{42}$$

associated with the nodes $\{z_0,\ldots,z_M\}$ satisfies the orthogonality conditions

$$\int_{-1}^1 q_{M+1}(z)b(z)\mu(z)dz = 0 \tag{43}$$

for all polynomial $b(z)$ of degree at most $m-1$.

In other words, if we can find a set of nodes $\{z_0,\ldots,z_M\}$ such that $q_{M+1}(z)$ is orthogonal in $L^2_\mu([-1,1])$ to any polynomial of degree $m-1$ then the quadrature rule (39) has degree of exactness $M+m$.

Proof. Suppose that $f(z)$ in (39) is a polynomial of degree $m+M$. Divide $f(z)$ by (42) to obtain[4]

$$f(z) = \underbrace{q_{M+1}(z)}_{\text{divisor}} \underbrace{d_{m-1}(z)}_{\text{quotient}} + \underbrace{r_M(z)}_{\text{reminder}} \tag{45}$$

Note that the degree of the quotient is $(m+M)-(M+1)=m-1$ while the degree of the remainder is $(M+1)-1=M$ (i.e., a polynomial that cannot be divided by $q_{M+1}(z)$). Since $r_M(z)$ is a polynomial of degree M it can be integrated exactly by the quadrature rule with $M+1$ nodes. This yields,

$$\sum_{k=1}^{M} w_k r_M(z_k) = \int_{-1}^{1} r_M(z)\mu(z)dz = \int_{-1}^{1} f(z)\mu(z)dz - \int_{-1}^{1} q_{M+1}(z)d_{m-1}(z)\mu(z)dz. \tag{46}$$

If hypothesis (43) holds true then the last term at the right hand side vanishes. This allows us to conclude that

$$\int_{-1}^{1} f(z)\mu(z)dz = \sum_{k=1}^{M} w_k r_M(z_k), \tag{47}$$

i.e., that the polynomial $f(z)$ of degree $M+m$ can be integrated exactly on the grid with $M+1$ points $\{z_0,\ldots,z_M\}$ satisfying the condition (43).

□

Example (Gauss-Legendre quadrature): Let $\{z_0,\ldots,z_M\}$ the zeros of the Legendre orthogonal polynomial $L_{M+1}(z)$, i.e., $L_{M+1}(z_j)=0$. Clearly, the nodal polynomial $q_{M+1}(z)$ in theorem (2) coincides (modulus sign) with $L_{M+1}(z)$. In fact, $q_{M+1}(z)$ and $L_{M+1}(z)$ have the same zeros. Setting $\mu(z)=1$ in (43) (Legendre polynomials are orthogonal in $[-1,1]$ with respect to $\mu(z)=1$) yields

$$\int_{-1}^{1} L_{M+1}(z)b(z)dz = 0. \tag{48}$$

At this point we write the polynomial $b(z)$ (of degree $m-1$) in terms of a linear combination of Legendre polynomials

$$b(z) = \sum_{j=0}^{m-1} b_j L_j(z). \tag{49}$$

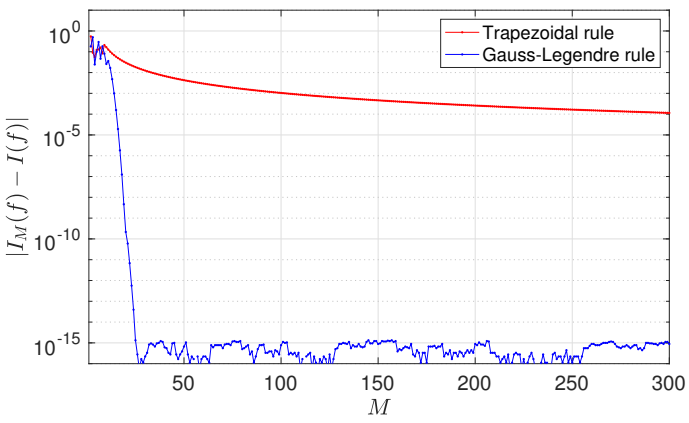
Next, substitute (49) into (48) to obtain

$$\sum_{j=0}^{m-1} b_j \int_{-1}^{1} L_{M+1}(z)L_j(z)dz = 0. \tag{50}$$

By using orthogonality of the Legendre polynomials we see that the maximum degree $m-1$ of the polynomial $b(z)$ that satisfies equation (50) is $m-1=M$ (i.e., $m=M+1$). Hence, the degree of exactness of Gauss-Legendre quadrature is $M+m=2M+1$. This means that with $M+1$ points we can integrate

[4] As an example, consider the polynomial division of $f(z)=z^3+z^2-3z+4$ by $q_2(z)=z^2-3z+2$. To this end, we firt multiply $q_2(z)$ by z to obtain z^3-3z^2+2z. Subtracting this from $f(z)$ yields the reminder $4z^2-5z+4$. At this point we multiply $q_2(z)$ by 4, i.e., $4q_2(z)=4z^2-12z+8$ and subtract it from $4z^2-5z+4$ to obtain the final remainder $7z-4$. Hence, we obtained the factorization

$$z^3+z^2-3z+4 = \underbrace{(z^2-3z+2)}_{\text{divisor}} \underbrace{(z+4)}_{\text{quotient}} + \underbrace{(7z-4)}_{\text{reminder}}. \tag{44}$$



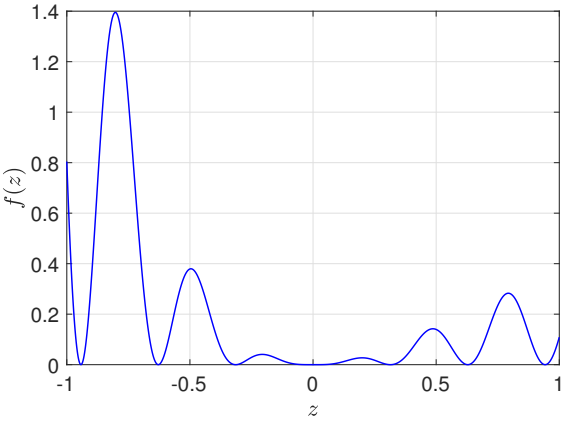


Figure 3: Error in the numerical approximation of the integral (53) using the Gauss-Legendre quadrature rule and the trapezoidal rule versus the number of collocation points M. Note that the Gauss-Legendre rule converges exponentially fast. In particular, with only 25 points the Gauss-Legendre rule achieves error 1.3×10^{-15}. On the other hand, the trapezoidal rule with 300 points achieves error 10^{-4}.

exactly polynomials up to degree $2M+1$! Not really intuitive, huh? Regarding the integration weights for the Gauss-Legendre quadrature, it can be shown that

$$w_j = \frac{2}{(1-z_j^2)\left[L'_{M+1}(z_j)\right]^2} \qquad j = 0, \ldots, n. \tag{51}$$

Moreover, for every $f \in H^s([-1,1])$ we have the following spectral convergence result[5] [9, p. 437]

$$\left| \int_{-1}^{1} f(z)dz - \sum_{k=0}^{M} f(z_k) w_k \right| \leq C M^{-s} \left\| f \right\|_{H^s([-1,1])}. \tag{52}$$

In Appendix A, we discuss similar results for Chebyshev-Gauss-Lobatto quadrature. If f is infinitely differentiable then convergence is exponential. As an example, in Figure 3 we compare the error in the numerical approximation of the integral

$$I(f) = \int_{-1}^{1} f(z)dz \qquad f(z) = e^{-z} z^2 \sin(10z)^2 \tag{53}$$

using the Gauss-Lobatto rule and the trapezoidal rule.

Lemma 1. The *maximum degree of exactness* of the interpolatory quadrature formula (39) is $2M+1$.

Proof. The proof is very simple. Suppose we could choose the max degree of $b(z)$ to be $m = M+2$. Following what we just said for the Gauss-Legendre quadrature this would imply

$$\int_{-1}^{1} q_{M+1}^2(z) \mu(z) dz = 0, \tag{54}$$

i.e., $q_{M+1}^2(z) = 0$ which is impossible (see [9, Corollary 10.2]).

□

[5] The error estimate holds for Gauss-Legendre-Lobatto quadrature (see Table 1), which has degree of exactness $2M-1$.

	Gauss-Legendre	Gauss-Legendre-Lobatto
nodes $\{z_0,\dots,z_M\}$	$L_{M+1}(z)=0$	$(1-x^2)L'_M(z)=0$
Lagrange polynomials	$l_i(z)=\dfrac{L_{M+1}(z)}{(z-z_i)L'_{M+1}(z)}$	$l_i(z)=-\dfrac{1}{M(M+1)}\dfrac{(1-z^2)}{(z-z_i)}\dfrac{L'_M(z)}{L'_M(z_i)}$
Integration weights	$w_i(z)=\dfrac{2}{(1-z_i^2)\left[L'_{M+1}(z_i)\right]^2}$	$w_i(z)=\dfrac{1}{M(M+1)L_M(z_i)^2}$

Table 1: Gauss-Legendre (GL) and Gauss-Lobatto-Legendre (GLL) quadrature and interpolation rules. The GL rule has degree of exactness $2M+1$, while the GLL rule has degree of exactness $2M-1$.

Gauss and Gauss-Lobatto points have also excellent properties when used for polynomial interpolation (see Appendix B).

Computation of Gaussian quadrature points and weights. With the exception of a few special cases, like the Chebyshev polynomials, no closed form expressions for the quadrature nodes and weights are known (see, e.g., Table 1). Nevertheless, there is a simple and elegant way of computing these nodes as well as the corresponding weights based on the eigenvalues suitable tridiagonal matrices [6, §11.2]. The method relies on the three-term recurrence relation for orthogonal polynomials, written hereafter for monic orthogonal polynomials

$$\pi_{n+1}(z)=(z-\alpha_n)\pi_n(x)-\beta_n\pi_{n-1}(x). \tag{55}$$

We have seen that the coefficients α_n and β_n can be computed for every measure $\mu(z)$ using the Stieltjes algorithm (see Appendix B of Chapter 4). Equation (55) can be rewritten as

$$z\pi_n(z)=\pi_{n+1}(z)+\alpha_n\pi_n(x)+\beta_n\pi_{n-1}(x), \tag{56}$$

or in the following convenient matrix-vector form

$$z\underbrace{\begin{bmatrix}\pi_0(z)\\ \pi_1(z)\\ \pi_2(z)\\ \vdots\\ \pi_{n-1}(z)\\ \pi_n(z)\end{bmatrix}}_{\boldsymbol{\pi}(z)}=\underbrace{\begin{bmatrix}\alpha_0 & 1 & 0 & 0 & \cdots & 0\\ \beta_1 & \alpha_1 & 1 & 0 & \cdots & 0\\ 0 & \beta_2 & \alpha_2 & 1 & \cdots & 0\\ \vdots & & \ddots & \ddots & \ddots & \vdots\\ 0 & \cdots & 0 & \beta_{n-1} & \alpha_{n-1} & 1\\ 0 & \cdots & 0 & 0 & \beta_n & \alpha_n\end{bmatrix}}_{\text{Jacobi matrix }\boldsymbol{J}}\underbrace{\begin{bmatrix}\pi_0(z)\\ \pi_1(z)\\ \pi_2(z)\\ \vdots\\ \pi_{n-1}(z)\\ \pi_n(z)\end{bmatrix}}_{\boldsymbol{\pi}(z)}+\begin{bmatrix}0\\ 0\\ 0\\ \vdots\\ 0\\ \pi_{n+1}(z)\end{bmatrix}. \tag{57}$$

At this point, it is clear that the zeros of $\pi_{n+1}(z)$ are eigenvalues of the Jacobi matrix $\boldsymbol{J}$. In fact, if z_j is such that $\pi_{n+1}(z_j)=0$ then

$$\boldsymbol{J}\boldsymbol{\pi}(z_j)=z_j\boldsymbol{\pi}(z_j). \tag{58}$$

This eigenvalue problem may be solved using the QR algorithm. This yields the Gauss quadrature points $\{z_0,\dots,z_n\}$. The corresponding quadrature weights can be computed by expanding each Lagrange polynomial $l_j(z)$ in (40) in terms of $\pi_j(z)$, and using orthogonality of $\pi_j(z)$ relative $\mu(z)$ to we obtain

$$w_k=\int_{-1}^1 l_k(z)\mu(z)dz=\sum_{j=0}^M a_{kj}\int_{-1}^1\pi_j(z)\mu(z)dz=a_{k0}\int_{-1}^1\mu(z)dz \qquad \text{(integration weights)}. \tag{59}$$

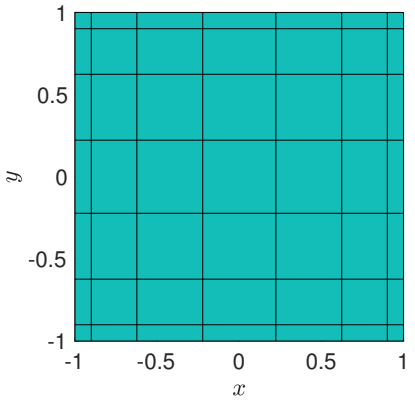


Figure 4: Tensor product of two one-dimensional Chebyshev grids (97) of 8 points.

Quadrature and interpolation on tensor product grids. Suppose we are interested in approximating the integral of a two-dimensional function $g(x,y)$ in $[-1,1]^2$ relative to the separable integration weight

$$\mu(x,y) = \mu_1(x)\mu_2(y), \tag{60}$$

i.e.,

$$\int_{[-1,1]^2} g(x,y)\mu_1(x)\mu_2(y)dxdy. \tag{61}$$

By leveraging the isomorphism

$$L^2_\mu([-1,1]^2) = L^2_{\mu_1}([-1,1]) \otimes L^2_{\mu_2}([-1,1]) \tag{62}$$

we see that we can represent $g(x,y)$ in terms of a tensor product of one-dimensional bases involving functions of x alone and y alone. In particular, such bases could be Lagrange characteristic polynomials corresponding to appropriate one-dimensional grids in $x \in [-1,1]$ and $y \in [-1,1]$. Let us denote by

$$\{x_0,\ldots,x_M\} \quad \text{and} \quad \{y_0,\ldots,y_N\} \tag{63}$$

the aforementioned one-dimensional grids, and by $\{l_j(x)\}$ and $h_i(y)$ the corresponding Lagrange polynomials. Then 2D polynomial interpolant of the dataset $\{g(x_i,y_j)\}$ can be written as

$$\Pi g(x,y) = \sum_{i=0}^{M}\sum_{j=0}^{N} g(x_i,y_j)\, l_i(x)h_j(y). \tag{64}$$

Clearly, $\Pi g(x,y)$ is a polynomial of total degree $M+N$. In Figure 4 we show a tensor product of two Gauss-Chebyshev-Lobatto one-dimensional grids (97). In Figure 5 we plot a few 2D Lagrange characteristic polynomials $l_i(x)h_j(y)$ associated with the Chebyshev grid shown in Figure 4. The convergence rate of the interpolant (64) is determined by the tensor product interpolation grid. In particular, for each fixed $x = x_j$ or $y = y_k$ it is clear that the spectral convergence results summarized in Appendix B hold. With the 2D interpolant (64) available, it is straightforward to derive a 2D interpolatory quadrature rule. In

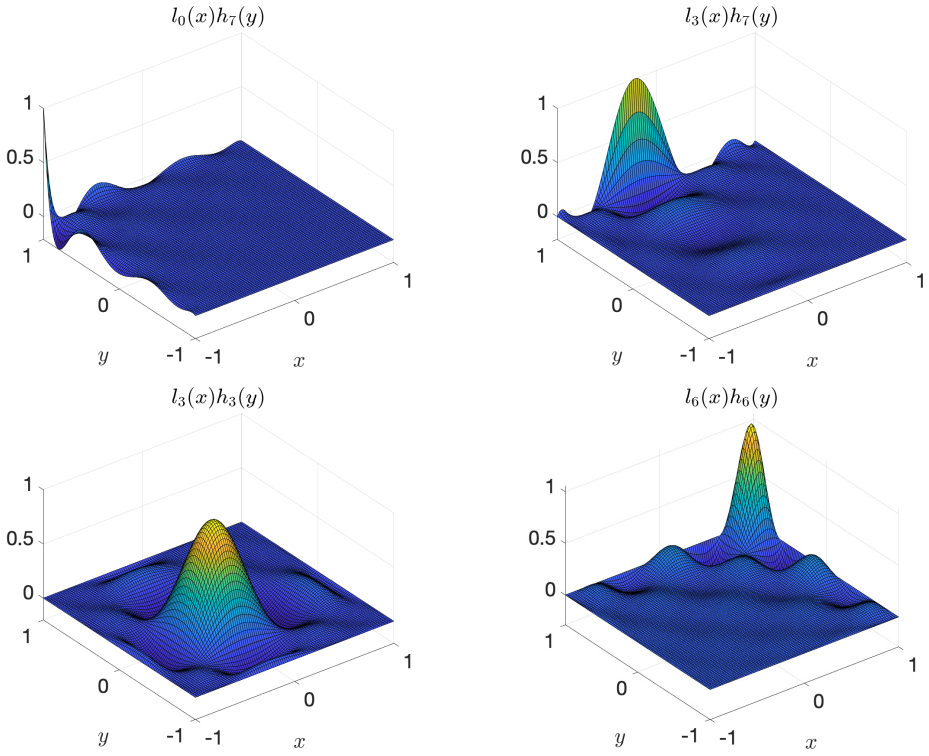


Figure 5: 2D Lagrange characteristic polynomials $l_i(x)h_j(y)$ associated with the 2D Chebyshev grid shown in Figure 4.

fact,

$$\begin{aligned}\int_{-1}^{1}\int_{-1}^{1} g(x,y)\mu_1(x)\mu_2(y)dxdy &\simeq \int_{-1}^{1}\int_{-1}^{1} \Pi g(x,y)\mu_1(x)\mu_2(y)dxdy \\ &= \sum_{i=0}^{M}\sum_{j=0}^{N} g\left(x_i,y_j\right) \underbrace{\int_{-1}^{1} l_i(x)\mu_1(x)dx}_{w_i} \underbrace{\int_{-1}^{1} h_j(y)\mu_2(y)dy}_{q_j} \\ &= \sum_{i=0}^{M}\sum_{j=0}^{N} g\left(x_i,y_j\right) w_i q_j. \end{aligned} \tag{65}$$

Next, consider the random variable

$$\eta(\omega) = g(\xi_1,\ldots,\xi_n) \tag{66}$$

and assume that all random variables $\{\xi_1,\ldots,\xi_n\}$ are i.i.d. with PDF $p_\xi(x)$ supported in $[-1,1]$. We have

$$\mathbb{E}\{\eta(\omega)\} = \int_{-1}^{1}\cdots\int_{-1}^{1} g(x_1,\ldots,x_n)p_\xi(x_1)\cdots p_\xi(x_n)dx_1\cdots dx_n. \tag{67}$$

We approximate this integral using tensor product PCM. To this end, we first construct a one-dimensional quadrature rule with high-degree of exactness (i.e., Gauss or Gauss-Lobatto) using the methods described in previous sections. With the one-dimensional quadrature points $\{x_0,\ldots,x_M\}$ and quadrature weights $\{w_0,\ldots,w_M\}$ available we approximate the integral in (67) as

$$\mathbb{E}\{\eta(\omega)\} = \sum_{j_1=0}^{N}\cdots\sum_{j_n=0}^{N} g\left(x_{j_1},\ldots,x_{j_n}\right) w_{j_1}\cdots w_{j_n}. \tag{68}$$

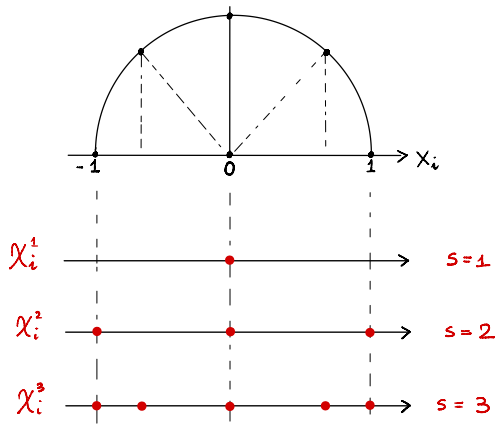


Figure 6: Chebyshev-Gauss-Lobatto (GCL) nested point sets χ_i^1, χ_i^2, and χ_i^3.

Computational cost: To compute all the sums we essentially need to evaluate all $g(\xi_1, \ldots, x_n)$ at a number of points that grows as $(N+1)^n$, i.e., polynomially in N (number of points) and *exponentially* in n (dimension). In dimension $n = 10$ with just $N + 1 = 10$ points per dimension this yields 10^{10} collocation points! Each point has n coordinates. Hence, to store the grid in double precision floating point (64 bits, 8 Bytes per floating point number) we need

$$10^{10} \times 10 \times 8 = 800\,\text{GB}. \tag{69}$$

To store $g\left(x_{j_1}, \ldots, x_{j_n}\right)$ we need an extra 80 GB, and 1.25 Bytes for the vector of weights. Hence, similarly to polynomial chaos, tensor product PCM undergoes an exponential growth of degrees of freedom with the dimension of the problem. However, differently than polynomial chaos, PCM is a *non-intrusive* method that allows us to perform UQ calculations on legacy codes in a straightforward way, without coding polynomial chaos propagators or PDF equation solvers from scratch.

Sparse grids

Sparse grids are numerical techniques to represent, integrate or interpolate high dimensional functions. They were originally developed by the Russian mathematician Sergey A. Smolyak, and are based on a *sparse tensor product* construction [3]. The fundamental building block of sparse grids is a one-dimensional nested points set, e.g., the Gauss-Chebyshev-Lobatto grid (97) for $M = 2, 4, 8, \ldots, 2^s$, or any other nested point set. To describe how sparse grids are constructed, let

$$\chi_i^s = \{x_i^1 \ldots, x_i^{n_s}\} \tag{70}$$

be the nested points set in the variable x^i where

$$n_1 = 1, \qquad n_s = 2^{s-1} + 1 \quad (s \geq 2), \tag{71}$$

is the total number of points in the nested point set, e.g., in the Gauss-Chebyshev-Lobatto grid. In Figure 6 we provide a graphical representation of the GCL nested point set.

The *level* q sparse grids in d dimensions is defined as the multidimensional point set

$$H_q^d = \bigcup_{q+1-d \leq i_1 + \cdots + i_d \leq q} \chi_1^{i_1} \times \cdots \times \chi_d^{i_d}, \tag{72}$$

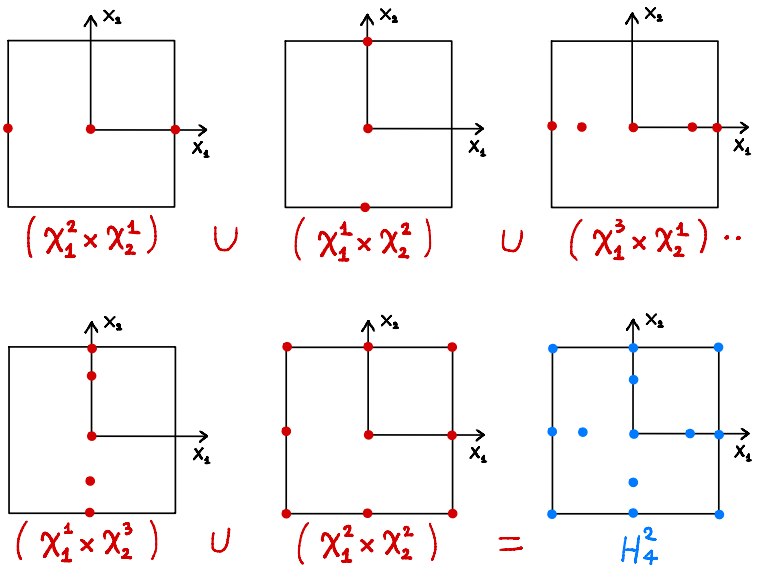


Figure 7: Construction of two-dimensional Chebyshev-Gauss-Lobatto (GCL) sparse grids of level 4 (see Eq. (74)). The final point set is denoted by H_4^2.

i.e., the union of suitable Cartesian products of one-dimensional grids. As we will see sparse grids follow naturally from the definition of Smolyak interpolant of a multivartiate function, which we will discuss in detail in the next section. For now, we simply notice that the point set (72) is nested in the sense that

$$H_{q-1}^d \subset H_q^d \tag{73}$$

if χ_i^s is a nested point set.

Example (Level 4 Gauss-Chebyshev-Lobatto (GCL) sparse grids in two-dimensions): To derive the level 4 GCL sparse grids in two dimensions we set $q = 4$ and $d = 2$ in (72). This yields

$$\begin{aligned} H_4^2 &= \bigcup_{3 \leq i_1 + i_2 \leq 4} \chi_1^{i_1} \times \chi_2^{i_2} \\ &= (\chi_1^2 \times \chi_2^1) \cup (\chi_1^1 \times \chi_2^2) \cup (\chi_1^3 \times \chi_2^1) \cup (\chi_1^1 \times \chi_2^3) \cup (\chi_1^2 \times \chi_2^2)\,. \end{aligned} \tag{74}$$

The Cartesian product grids appearing in this expression can be easily derived by taking Cartesian products of the elementary 1D grids shown in Figure 6. Such product grids are shown in Figure 7 In Figure 8 we plot CGL sparse grids of level 5 and 6 in 2D and 3D.

Interpolation on sparse grids. Let Π_i^s be the interpolation operator in the variable x_i corresponding to the 1D point set (70). Note that for $s = 1$ we have that χ_i^1 has only one point (see Figure 6). Therefore that Π_i^1 is an interpolant at one point only (for polynomial this is the constant function). Define the difference between two interpolation operators as[6]

$$\Delta_i^0 = 0 \qquad \Delta_i^s = \Pi_i^s - \Pi_i^{s-1}. \tag{75}$$

The *Smolyak interpolant* of a multivariate function $f(x_1, \ldots, x_d)$ is defined as

$$S_q^d(f) = \sum_{i_1 + \cdots + i_d \leq q} \Delta_1^{i_1} \otimes \cdots \otimes \Delta_d^{i_d}, \tag{76}$$

[6]In equation (75) Π_i^s denotes the one-dimensional interpolant of a function $f(\boldsymbol{x})$ on a grid in the variable x_i.

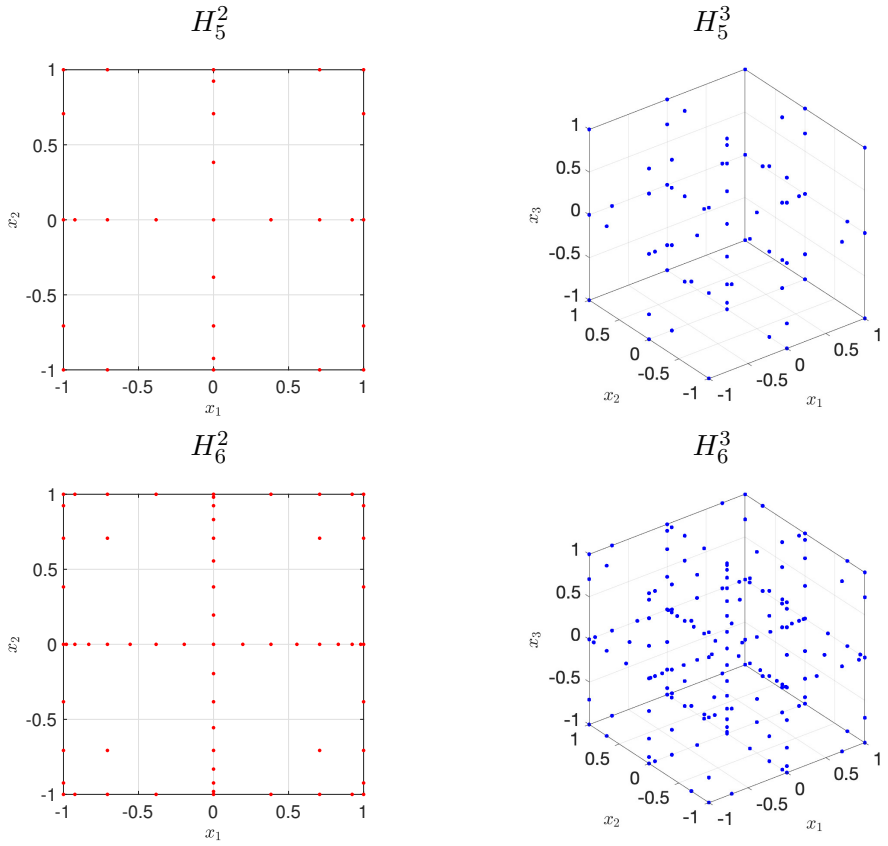


Figure 8: Chebyshev-Gauss-Lobatto (GCL) sparse grids of level 5 (first row) and 6 (second row) in dimension 2 and 3.

where $i_j \geq 1$, and $q \geq d$ is a parameter called sparse grids level. To clarify the meaning of (76), let us compute the two-dimensional Smolyak interpolant of level 3 of a two-dimensional function $f(x_1, x_2)$. By definition,

$$\begin{aligned} S_3^2(f) = &\sum_{i_1+i_2\leq 3} \Delta_1^{i_1} \otimes \Delta_2^{i_2} \\ =&\Delta_1^1 \otimes \Delta_2^1 + \Delta_1^2 \otimes \Delta_2^1 + \Delta_1^1 \otimes \Delta_2^2 \\ =&\Pi_1^1 \otimes \Pi_2^1 + \left(\Pi_1^2 - \Pi_1^1\right) \otimes \Pi_2^1 + \Pi_1^1 \otimes \left(\Pi_2^2 - \Pi_2^1\right) \\ =&\Pi_1^2 \otimes \Pi_2^1 + \Pi_1^1 \otimes \Pi_2^2 - \Pi_1^1 \otimes \Pi_2^1. \end{aligned} \tag{77}$$

The interpolant (77) is built upon the sparse grid H_3^2 as shown in Figure 9. Note that each point is accounted for only once in the final grid H_3^2 (the origin is summed up twice and subtracted once). Specifically, we have

$$\begin{aligned} S_3^2(f) = &\left[f(-1,0)l_1^{(1)}(x_1) + f(0,0)l_2^{(1)}(x_1) + f(1,0)l_3^{(1)}(x_1)\right] l_1^{(2)}(x_2) + \\ &\left[f(0,-1)l_1^{(1)}(x_2) + f(0,0)l_2^{(1)}(x_2) + f(0,1)l_3^{(1)}(x_2)\right] l_1^{(2)}(x_1) - \\ &f(0,0)l_1^{(2)}(x_1)l_1^{(2)}(x_2), \end{aligned} \tag{78}$$

where $l_j^{(i)}$ are the Lagrange polynomials shown in Figure 9. By substituting (75) into (76) we can rewrite the Smolyak interpolant in terms of elementary one-dimensional interpolants as

$$S_q^d(f) = \sum_{q+1-d\leq i_1+\cdots+i_d\leq q} (-1)^{q-i_1-\cdots-i_d} \binom{d-1}{q-i_1-\cdots-i_d} \Pi_1^{i_1} \otimes \cdots \otimes \Pi_d^{i_d}. \tag{79}$$

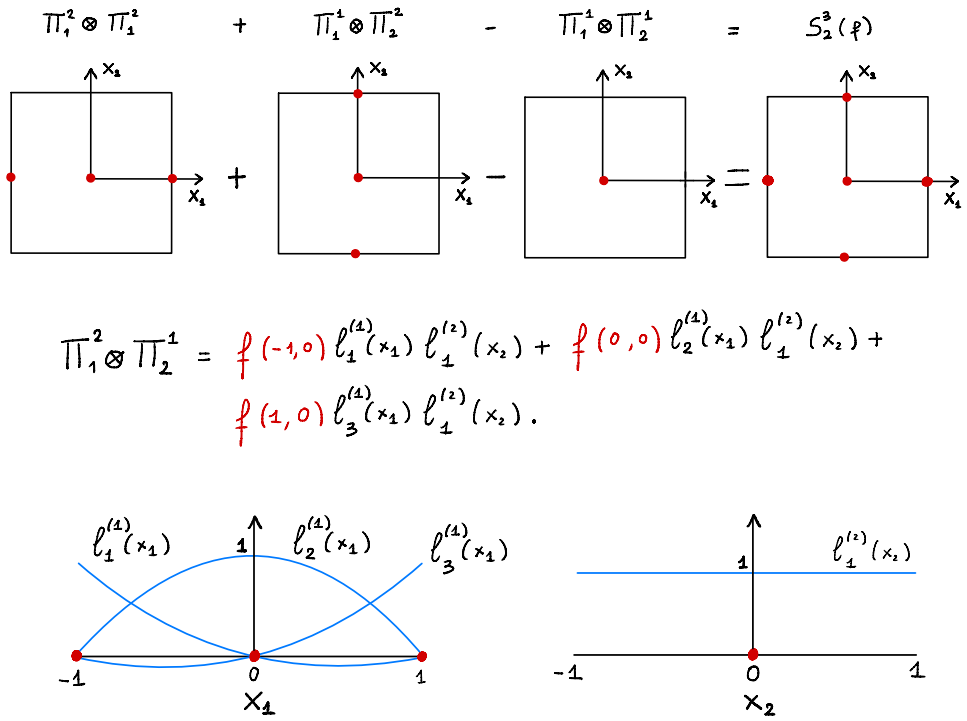


Figure 9: Construction of the Smolyak interpolant (77) and corresponding grids. Similar expression can be derived for $\Pi_1^1 \otimes \Pi_2^2$ and $\Pi_1^1 \otimes \Pi_2^1$.

Hereafter we summarize an error estimate obtained in [1, Remark 11].

Theorem 3. Let $H_\mu^s([-1,1]^d)$ be the weighted Sobolev space of order s, with weight[7]

$$\mu(x_1,\ldots,x_d) = \left[(1-x_1^2)\cdots(1-x_d^2)\right]^{-1/2}. \tag{80}$$

Then,

$$\left\| f - S_q^d(f) \right\|_{L_\mu^2([-1,1]^d)} \leq C(s,d) n^{-s} \log(n)^{(s+1)(d-1)} \left\| f \right\|_{H_\mu^s([-1,1]^d)}, \tag{81}$$

where $C(s,d)$ is a constant that depends on s and d, and $n = n$ is the total number of sparse grids points (which depends on d and q).

As easily seen, convergence is no longer spectral (unless $d = 1$) because of the factor $\log(n)^{(s+1)(d-1)}$.

Integration on sparse grids. The Smolyak algorithm can be used to construct cubature formulas to integrate high-dimensional functions. The key idea is very simple: replace the function with the Smolyak interpolant on a sparse grid and then integrate. Assuming that the integration weight is separable as in Theorem 3 we

$$\int_{[-1,1]^d} f(\boldsymbol{x}) \prod_{j=1}^d \mu_j(x_j) d\boldsymbol{x} \simeq \underbrace{\int_{[-1,1]^d} S_q^d(f)(\boldsymbol{x}) \prod_{j=1}^d \mu_j(x_j) d\boldsymbol{x}}_{I_q^d(f)}. \tag{82}$$

[7] Note that the weight (80) corresponds to a tensor product of Chebyshev polynomials

This yields an interpolatory quadrature rule with high degree of exactness [8]. In particular, by substituting (79) into (82) we obtain

$$\begin{aligned} I_q^d(f) &= \int_{[-1,1]^d} S_q^d(f)(\boldsymbol{x}) \prod_{j=1}^d \mu_j(x_j) d\boldsymbol{x} \\ &= \sum_{q+1-d \leq i_1+\cdots+i_d \leq q} (-1)^{q-i_1-\cdots-i_d} \binom{d-1}{q-i_1-\cdots-i_d} U_1^{i_1} \otimes \cdots \otimes U_d^{i_d}, \end{aligned} \tag{83}$$

where

$$U_j^{i_j} = \int_{-1}^1 \Pi_j^{i_j} \mu_j(x_j) dx_j. \tag{84}$$

Example: To illustrate (83), let us integrate the two-dimensional interpolant $S_3^2(f)$ defined in (78) on $[-1,1]^2$. This yields the interpolatory quadrature formula

$$\begin{aligned} I_3^2(f) =& f(-1,0) w_{11}^1 w_{12}^2 + f(0,0) \left[w_{21}^1 w_{12}^2 + w_{22}^1 w_{11}^2 - w_{11}^1 w_{12}^1 \right] + \\ & f(0,-1) w_{12}^1 w_{11}^2 + f(1,0) w_{31}^1 w_{12}^2 + f(0,1) w_{32}^1 w_{11}^2, \end{aligned} \tag{85}$$

where

$$w_{ip}^j = \int_{-1}^1 l_i^{(j)}(x_p) \mu(x_p) dx_p. \tag{86}$$

Note that the integration weights in sparse grids are not necessarily all positive. For example, the weight multiplying $f(0,0)$, i.e., $w_{21}^1 w_{12}^2 + w_{22}^1 w_{11}^2 - w_{11}^1 w_{12}^1$, could be negative. Regarding the degree of exactness, for CGL sparse grids we have the following result (see [8, Corollary 3])

Theorem 4. Let $q = \sigma d + \tau$, $\sigma \in \mathbb{N}$, $\tau \in \{0, \ldots, d-1\}$ Then $I_q^d(f)$ has degree of exactness

$$\begin{cases} 2(q-d)+1 & \text{if} \quad q < 4d \\ 2^{\sigma-2}(d+1+\tau) + 2d - 1 & \text{if} \quad q \geq 4d \end{cases} \tag{87}$$

Other convergence estimates for interpolatory quadrature rules on sparse grids can be derived based on convergence estimates of one-dimensional quadrature (see [1, Remark 11]).

Appendix A: Chebyshev-Gauss-Lobatto quadrature

Let briefly review the main ingredients of the Gauss-Lobatto Chebyshev expansion. For more details we refer to [6]. We first recall that the Chebyshev polynomials of the first kind are defined as[8]

$$T_k(x) = \cos(k \arccos(x)) \qquad x \in [-1,1] \qquad \text{(trigonometric representation)}. \tag{91}$$

It can be shown that $T_k(x)$ (like any other orthogonal polynomial) satisfy the three-term recurrence relation

$$\begin{aligned} T_0(x) &= 1 \\ T_1(x) &= x \\ T_{n+1}(x) &= 2x\, T_n(x) - T_{n-1}(x). \end{aligned} \tag{92}$$

and the orthogonality conditions

$$\int_{-1}^{1} T_k(x) T_j(x) \underbrace{\frac{1}{\sqrt{1-x^2}}}_{\mu(x)} dx = \delta_{kj} \left\| T_k \right\|^2_{L^2_\mu}. \tag{93}$$

Note that the first polynomials which gives

$$T_2(x) = 2x^2 - 1, \qquad T_3(x) = 4x^3 - 3x \qquad T_4(x) = 8x^4 - 8x^2 + 1, \ldots. \tag{94}$$

The Chebyshev-Gauss-Lobatto nodes are zeros of the polynomial

$$Q_{M+1}(x) = (1-x^2)\frac{dT_M(x)}{dx}, \tag{95}$$

i.e., $x_0 = -1$, $x_M = 1$ and all maxima and minima of $T_M(x)$. By differentiating (91) with respect to x we obtain

$$\frac{dT_M(x)}{dx} = \frac{\sin(M \arccos(x))}{\sqrt{1-x^2}}. \tag{96}$$

Hence $Q_{M+1}(x) = 0$ implies that

$$x_j = -\cos\left(\frac{k\pi}{M}\right) \qquad j = 0, \ldots, M \qquad \text{(Chebyshev-Gauss-Lobatto points)}. \tag{97}$$

These points are obtained by dividing half unit circle in evenly-spaced parts and projecting them onto the x-axis. Note also that Chebyshev grid points are *nested* for $M = 2, 4, 8, \ldots, 2^s$.

It can be shown that the Lagrange characteristic polynomials associated with the Gauss-Chebyshev-Lobatto nodes are

$$l_j(x) = \frac{(-1)^{M+j+1}(1-x^2)}{d_j M^2 (x - x_j)} \frac{dT_M(x)}{dx} = \frac{(-1)^{M+j+1}\sqrt{(1-x^2)}}{d_j M^2 (x - x_j)} \sin(M \arccos(x)), \tag{98}$$

where x_j is given in (97) and

$$d_0 = d_M = 2 \qquad d_1 = d_2 = \cdots = d_{M-1} = 1. \tag{99}$$

[8] Note that (91) are indeed polynomials. For example,

$$\cos(\arccos(x)) = x, \tag{88}$$

$$\cos(2\arccos(x)) = 2\left(\cos(\arccos(x))\right)^2 - 1 = 2x^2 - 1, \tag{89}$$

$$\cos(3\arccos(x)) = 4\left(\cos(\arccos(x))\right)^3 - 3\cos(\arccos(x)) = 4x^3 - 3x. \tag{90}$$

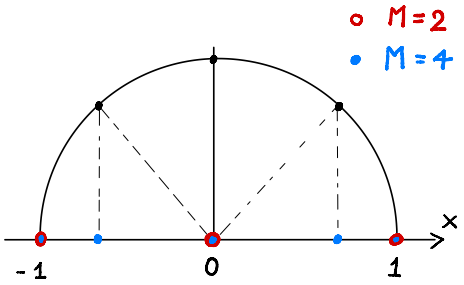


Figure 10: Nested property of Chebyshev grids for $M = 2^s$ $(s = 1, 2, 3, \ldots)$.

For any function $f(x)$ defined in $[-1, 1]$ we have the following Lagrangian interpolant

$$\Pi_M f(x) = \sum_{k=0}^{M} f(x_k) l_k(x), \qquad x \in [-1, 1]. \tag{100}$$

At this point we integrate (100) to obtain the quadrature formula

$$\int_{-1}^{1} f(x) \frac{1}{\sqrt{1-x^2}} dx \simeq \sum_{k=0}^{M} f(x_k) w_k, \tag{101}$$

where

$$w_k = \int_{-1}^{1} \frac{l_k(x)}{\sqrt{1-x^2}} dx = \frac{\pi}{M d_j} \tag{102}$$

and d_j is defined in (99).

Appendix B: Lagrangian interpolation at Gauss points

The quadrature rule (39) induces a discrete inner product that establishes a correspondence between series expansions in terms of orthogonal polynomials[9] and Lagrangian interpolation formulas. To show this, let

$$f(z) \simeq \sum_{k=0}^{M} a_k P_k(z) \qquad a_k = \frac{(f,P_k)_{L^2_\mu([-1,1])}}{(P_k,P_k)_{L^2_\mu([-1,1])}} \tag{103}$$

be a polynomial expansion of $f(z)$ in $[-1,1]$, where $\{P_0,\ldots,P_M\}$ is a set of polynomials orthogonal relative to the weight function $\mu(z)$. Consider the following Gauss approximation the inner product

$$\begin{aligned}(f,P_k)_{L^2_\mu([-1,1])} &= \int_{-1}^{1} f(z)P_k(z)\mu(z)dz \\ &\simeq \sum_{j=0}^{M} f(z_j)P_k(z_j)w_j, \qquad \text{(discrete inner product)}\end{aligned} \tag{104}$$

where

$$\{z_0,\ldots,z_M\} \quad \text{and} \quad \{w_0,\ldots,w_M\} \tag{105}$$

are $M+1$ Gauss quadrature points and quadrature weights, respectively. Recall that the Gauss rule (104) has degree of exactness $2M+1$ and therefore it can be used to compute

$$\gamma_k = (P_k,P_k)_{L^2_\mu} \tag{106}$$

exactly up to $k=M$. A substitution of (104) into (103) yields

$$f(z) \simeq \sum_{j=0}^{M} f(z_j) \underbrace{\sum_{k=0}^{M} \frac{w_j}{\gamma_k} P_k(z_j)P_k(z)}_{l_j(z)}. \tag{107}$$

In this form, we recognize that the Lagrangian interpolation formula, where

$$l_j(z) = \sum_{k=0}^{M} \frac{w_j}{\gamma_k} P_k(z_j)P_k(z) \tag{108}$$

are Lagrange characteristic polynomials associated with the Gauss nodes (105).

The identification of the approximation (103) with the Lagrangian interpolant (107) at Gauss nodes (105) suggests a mathematically equivalent but computationally different way of representing the function $f(z)$. Regarding the approximation error in (103), the following general estimate in terms of the uniform norm holds true.

Theorem 5. Let $f \in C^0([-1,1])$ and $\Pi_M f(z)$ the polynomial of degree M interpolating $f(z)$ at $\{z_0,\ldots,z_M\}$. Then

$$\|f(z) - \Pi_M f(z)\|_\infty \leq (1+\Lambda_M) \inf_{\Psi\in\mathbb{P}_M} \|f(z)-\Psi(z)\|_\infty \tag{109}$$

where

$$\Lambda_M = \max_{z\in[-1,1]} \lambda_M(z) \qquad \text{(Lebesgue constant)}, \tag{110}$$

$$\lambda_M(z) = \sum_{j=0}^{M} |l_j(z)| \qquad \text{(Lebesgue function)}. \tag{111}$$

[9] We have seen in Chapter 4 that orthogonal polynomial expansions exhibit spectral convergence.

Proof. The proof for the upper bound (109) is very simple. Let $\Psi \in \mathbb{P}_M$ be the best approximating polynomial

$$\|f(z) - \Pi_M f(z)\|_\infty \leq \|f(z) - \Psi(z)\|_\infty + \|\Psi(z) - \Pi_M f(z)\|_\infty . \tag{112}$$

At this point, we represent $\Psi(z)$ and $\Pi_f(z)$ in terms of the same set of Lagrange polynomials associated with the grid $\{z_0, \ldots, z_M\}$ to obtain

$$\begin{aligned} \|\Psi(z) - \Pi_M f(z)\|_\infty &= \left\| \sum_{j=0}^{M} [\Psi(z_j) - f(z_j)]\, l_j(z) \right\|_\infty \\ &\leq \|\Psi(z) - f(z)\|_\infty \underbrace{\max_{z\in[-1,1]} \sum_{j=0}^{M} |l_j(z)|}_{\Lambda_M} . \end{aligned} \tag{113}$$

A substitution of (113) into (112) yields (109).

□

Note that the Lebesgue constant depends only on the set of grid points. Clearly, the smaller the Lebesgue constant, the smaller the interpolation error in the uniform norm. It can be shown that, no matter how we choose the points, the Lebesgue constant grows at least logarithmically with M, i.e. (see [6, p.102]),

$$\Lambda_M \geq \frac{2}{\pi} \log(1 + M) + C \quad \text{as} \quad M \to \infty. \tag{114}$$

Note that this does not mean that the interpolation error necessarily grows with M. It just means that the upper bound in (109) diverges as $M \to \infty$, i.e., that we cannot grant uniform convergence of Lagrangian interpolation using (109). In other words, for any given set of grid points there exist continuous functions for which the polynomial interpolant will exhibit non-uniform convergence. On the other hand, one can also show that for any given continuous function one can always construct a set of grid points that will result in a uniformly convergent polynomial representation.

It is possible to bound the Lebesgue constant corresponding to various types grids. For instance, for evenly-spaced grids of $M + 1$ points in $[-1, 1]$ we have

$$\frac{2^{M-2}}{M^2} \leq \Lambda_M \leq \frac{2^{M+3}}{M}. \tag{115}$$

Similarly, for the Gauss-Chebyshev-Lobatto (GCL) grid (97) we have (e.g., [6, p. 105])

$$\Lambda_M \leq \frac{2}{\pi} \log(M) + B \qquad (\text{finite } M), \tag{116}$$

where B is a suitable constant independent of M.

Example: In Figure 11 we plot the Lagrangian interpolant of

$$f(z) = \frac{1}{1 + 10z^2} \tag{117}$$

computed at 17 evenly-spaced nodes or 17 Gauss-Chebyshev-Lobatto (GCL) nodes ($M = 16$). In the same Figure we plot the Lebesgue functions of both interpolation problems. The Lebesgue constants for the evenly-spaced grid and the GCL grid are obtained, respectively, as

$$\Lambda_M^{eq} = 934.532 \qquad \Lambda_M^{GCL} = 2.468. \tag{118}$$

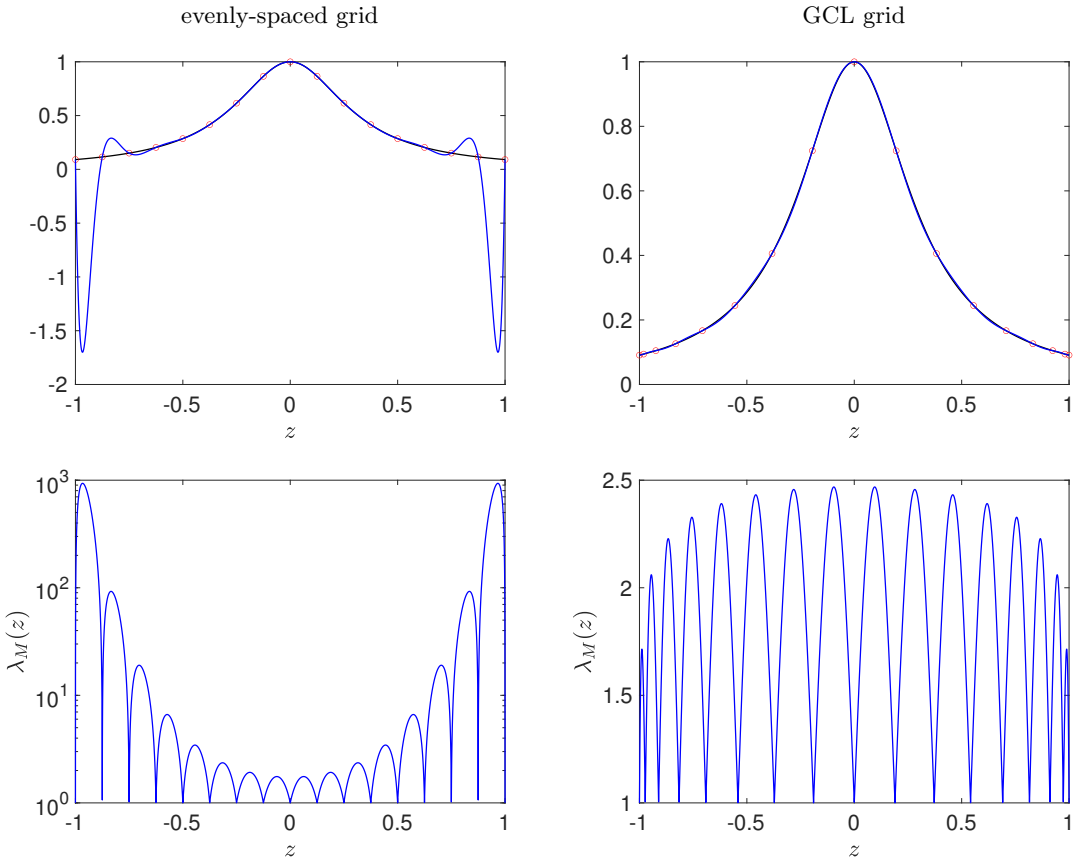


Figure 11: Lagrangian interpolation of $f(z) = (1 + 10z^2)^{-1}$ using 17 evenly-spaced nodes (left), and 17 GCL nodes (right). The Lebesgue functions $\lambda_M(z)$ associated with the evenly-spaced grid and the GCL grid have maxima $\Lambda_M^{eq} = 934.532$ and $\Lambda_M^{GCL} = 2.468$, respectively.

If we measure the interpolation error in terms of the $L^2_\mu([-1,1])$ (instead of the uniform norm we used in Theorem (5)) then by leveraging the correspondence between orthogonal polynomial expansions and the Lagrangian interpolant in Eq. (107) is possible to obtain spectral convergence results. For example, the following convergence result holds for Gauss-Legendre and Gauss-Legendre-Lobatto interpolation (see Table 1 and [6, p. 114]).

Theorem 6. Let $f(z) \in H^s([-1,1])$, $p \geq 1$. Then

$$\left\| f(z) - \sum_{k=0}^{M} f(z_k) l_k(z) \right\|_{L^2([-1,1])} \leq C M^{-s} \left\| f(z) \right\|_{H^s([-1,1])} \tag{119}$$

where $\{z_0, \ldots, z_M\}$ are either Gauss-Legendre points or Gauss-Legendre-Lobatto points (see Table 1).

References

[1] V. Barthelmann, E. Novak, and K. Ritter. High dimensional polynomial interpolation on sparse grids. *Advances in Computational Mechanics*, 12:273–288, 2000.

[2] Z. I. Botev, J. F. Grotowski, and D. P. Kroese. Kernel density estimation via diffusion. *Annals of Statistics*, 38(5):2916–2957, 2010.

[3] H. J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 13:147–269, 2004.

[4] J. Dick, F. Y. Kuo, and I. H. Sloan. High-dimensional integration: the quasi-Monte Carlo way. *Acta Numer.*, 22:133–288, 2013.

[5] J. Foo and G. E. Karniadakis. Multi-element probabilistic collocation method in high dimensions. *J. Comput. Phys.*, 229:1536–1557, 2010.

[6] J. S. Hesthaven, S. Gottlieb, and D. Gottlieb. *Spectral methods for time-dependent problems*, volume 21 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 2007.

[7] C. Lemieux. *Monte Carlo and Quasi-Monte Carlo Sampling*. Springer, 2009.

[8] E. Novak and K. Ritter. Simple cubature formulas with high polynomial exactness. *Constr. Approx.*, 15:499–522, 1999.

[9] A. Quarteroni, R. Sacco, and F. Salieri. *Numerical mathematics*. Springer, 2007.

[10] D. Xiu. *Numerical Methods for Stochastic Computations: A Spectral Method Approach*. Princeton University Press, 2010.