Non-Intrusive Approaches for Spectral UQ Methods

Olivier Le Maître[1]
with Colleague & Friend
Omar Knio

[1] LIMSI, CNRS
UPR-3251, Orsay, France
https://perso.limsi.fr/olm/

PhD course on UQ - DTU

Outline

- Reminder on PC expansions and notations
- Non-Intrusive *vs* Galerkin methods
- Least-squares and minimization formulations
- Non-Intrusive Spectral Projection
- Sparse Grids

Table of content

Model output and assumption

Consider the output of a model parametrized by a (finite) set of **independent** random variables $\boldsymbol{\xi} = (\xi_1 \dots \xi_N)$,

$$U(\boldsymbol{\xi}) \in \mathcal{V} \quad a.s.$$

The solution belongs almost surely to an Hilbert space $\mathcal{V}$, equipped with an inner product $(\cdot,\cdot)_{\mathcal{V}}$ and associated norm

$$\|U\|_{\mathcal{V}} = (U,U)_{\mathcal{V}}^{1/2}.$$

We assume that $U(\boldsymbol{\xi})$ is a second order random quantity in the sense that

$$(U(\boldsymbol{\xi}), \mathrm{U}(\boldsymbol{\xi}))_{\mathcal{V}}^{1/2} = \|U(\boldsymbol{\xi})\|_{\mathcal{V}} \in L_2(\Xi, p_{\boldsymbol{\xi}}),$$

and we write $U \in L_2(\mathcal{V}, \Xi, p_{\boldsymbol{\xi}})$, where Ξ is the domain of $\boldsymbol{\xi}$ and $p_{\boldsymbol{\xi}}$ the associated density:

$$p_{\boldsymbol{\xi}}(\boldsymbol{x}) = \prod_{i=1}^{\mathrm{N}} p_i(x_i), \quad \int \cdots \int_{\Xi} p_{\boldsymbol{\xi}}(\boldsymbol{x}) d\boldsymbol{x} = 1.$$

$$U \in L_2(\mathcal{V}, \Xi, p_{\boldsymbol{\xi}}) \Leftrightarrow \int \cdots \int_{\Xi} \|U(\boldsymbol{x})\|_{\mathcal{V}}^2 p_{\boldsymbol{\xi}}(\boldsymbol{x}) d\boldsymbol{x} < \infty.$$

PC expansion I

The model output $U \in L_2(\mathcal{V}, \Xi, p_{\boldsymbol{\xi}})$ in fact belongs to the tensored space $\mathcal{V} \otimes L_2(\Xi, p_{\boldsymbol{\xi}})$, and so has a **separable representation**

$$L_2(\mathcal{V}, \Xi, p_{\boldsymbol{\xi}}) \ni U(\boldsymbol{\xi}) = \sum_{l=0}^{\infty} \Phi_l(\boldsymbol{\xi}) u_l, \quad \Phi_l \in L_2(\Xi, p_{\boldsymbol{\xi}}),\ u_l \in \mathcal{V},\ l = 0, 1, 2, \ldots$$

The PC expansion of U rely on

1. the introduction of an **orthonormal polynomial basis** of $L_2(\Xi, p_{\boldsymbol{\xi}})$,

$$\text{span}\,\{\Psi_0, \Psi_1, \Psi_2, \ldots\} = L_2(\Xi, p_{\boldsymbol{\xi}}),$$

$$\langle \Psi_i, \Psi_j \rangle = \int \cdots \int_{\Xi} \Psi_i(\boldsymbol{x}) \Psi_j(\boldsymbol{x}) p_{\boldsymbol{\xi}}(\boldsymbol{x}) d\boldsymbol{x} = \delta_{ij} \langle \Psi_1, \Psi_i \rangle ,$$

2. such that U as for expression in the PC basis

$$U(\boldsymbol{\xi}) = \sum_{l=0}^{\infty} \Psi_l(\boldsymbol{\xi}) u_l, \quad u_l \in \mathcal{V},\ l = 0, 1, \ldots$$

where the equality stands in the L_2-sense.

PC expansion II

Exploiting the **product structure** of the density, the PC functions Ψ_l can be constructed by tensorization of 1-d families of univariate orthogonal polynomial. For instance, let $\{\psi_0^i, \psi_1^i, \psi_2^i, \dots\}$ be the family of orthogonal polynomials for the density p_i, where $\psi_j^i \in \Pi^j(\Xi_i)$ has degree j,

$$\left\langle \psi_l^i, \psi_j^i \right\rangle_i = \int_{\Xi_i} \psi_l^i(\xi)\psi_j^i(x)p_i(x)dx = \delta_{lj} \left\langle \psi_l^i, \psi_j^i \right\rangle .$$

We introduce the **multi-index** $\alpha = (\alpha_1 \dots \alpha_N) \in \mathbb{N}^N$, and define the **multi-variate polynomial** as product of univariate ones:

$$\Psi_\alpha(\boldsymbol{\xi}) = \prod_{i=1}^{N} \psi_{\alpha_i}^i(\xi_i).$$

- The partial degree of Ψ_α is $q = \max_i \alpha_i = \|\alpha\|_\infty$
- The total degree of Ψ_α is $q = \sum_i \alpha_i = \|\alpha\|_{\ell_1}$

PC expansion III

The model output $U(\boldsymbol{\xi})$ is sought has the PC expansion

$$L_2(\mathcal{V}, \Xi, p_{\boldsymbol{\xi}}) \in U = \sum_{\alpha \in \mathbb{N}^N} \Psi_\alpha(\boldsymbol{\xi}) u_{\boldsymbol{\alpha}}.$$

For practical computation, the expansion needs be **truncated**. Consider the finite multi-index set $\mathcal{A}$ and the truncated expansion

$$U(\boldsymbol{\xi}) \approx U^{\mathcal{A}}(\boldsymbol{\xi}) = \sum_{\alpha \in \mathcal{A}} \Psi_\alpha(\boldsymbol{\xi}) u_\alpha.$$

The **truncation error** is measured as

$$\epsilon^2(\mathcal{A}) = \int \cdots \int_{\Xi} \|U - U^{\mathcal{A}}\|^2_{\mathcal{V}} p_{\boldsymbol{\xi}}(\boldsymbol{x}) d\boldsymbol{x} = \sum_{\alpha \in \mathbb{N}^N \setminus \mathcal{A}} \|U_\alpha\|^2_{\mathcal{V}} \langle \Psi_\alpha, \Psi_\alpha \rangle .$$

Classical truncation strategies are based on

- Partial degree truncation: $\mathcal{A}^{\mathrm{No}}_\infty = \{\alpha \in \mathbb{N}^N, \|\alpha\|_\infty \leq \mathrm{No}\}$
- Total degree truncation: $\mathcal{A}^{\mathrm{No}}_{\ell_1} = \{\alpha \in \mathbb{N}^N, \|\alpha\|_{\ell_1} \leq \mathrm{No}\}$
- Hyperbolic cross product truncation: $\mathcal{A}^{\mathrm{No}}_{HC}(q < 1) = \{\alpha \in \mathbb{N}^N, \sum_i \alpha^q \leq \mathrm{No}^q\}$

All of these strategies are isotropic and converge as $\mathrm{No} \to \infty$. The dimension of the PC basis is $|\mathcal{A}|$.

Non-Intrusive methods

Given the truncated PC basis, defined from its index set $\mathcal{A}$, it remains to compute the PC coefficients u_α in the approximation $U^{\mathcal{A}}(\boldsymbol{\xi})$ of the model output:

$$U(\boldsymbol{\xi}) \approx U^{\mathcal{A}}(\boldsymbol{\xi}) = \sum_{\alpha \in \mathcal{A}} \Psi_\alpha(\boldsymbol{\xi}) u_\alpha.$$

In other words, the approximation is sought in the subspace $\mathcal{S}_{\mathcal{A}} \otimes \mathcal{V}$ of $L_2(\mathcal{V}, \Xi, p_{\boldsymbol{\xi}})$, where $\mathcal{S}_{\mathcal{A}}$ is defined as

$$\mathcal{S}_{\mathcal{A}} \doteq \operatorname{span}\{\Psi_\alpha, \alpha \in \mathcal{A}\} \subset L_2(\Xi, p_{\boldsymbol{\xi}}), \quad \dim \mathcal{S}_{\mathcal{A}} = |\mathcal{A}|.$$

- **Different methods** can be considered for the **determination** of the PC coefficients u_α.
- These methods **differ in the definition of the error** that approximation minimizes.
- These methods should however **converge to the unique solution as** $|\mathcal{A}| \to \infty$.
- These methods correspond to **different computational strategies** which are more or less suited to a given context. Classical considerations are computational complexity, available tools and softwares, ...

Galerkin *vs* Non-Intrusive methods I

The stochastic Galerkin projection

- It uses the model equations to derive an associated problem for the Galerkin modes u_α of the output. For this methods, the Galerkin modes are defined as to cancel the equations residual within the subspace $\mathcal{S}_\mathcal{A}$ spanned by the truncated PC basis. It is a method of weighted residual. It aims at **minimizing the error measured by the equation residual**.
- It assumes a **complete knowledge of the model equations**, and the **PC expansion of all model unknowns***.
- The formulation of the Galerkin problem **can be challenging** in particular in presence of strong model non-linearities.†
- Derivation and coding of efficient Galerkin solvers **can be time-consuming when it cannot reuse effectively deterministic code components**. It can also require the development of specific numerical methods (stabilization schemes, new preconditionners, . . .)
- Code verification and certification can also be an issue.‡

---

* May require significant memory requirement for large models.
† For instance inequalities, model branching, . . .
‡ Not to talk of UQ using legacy codes!

Galerkin *vs* Non-Intrusive methods II

Non-Intrusive methods

- Non-Intrusive methods refer to the set of approaches that reuse **deterministic codes as black-boxes**. By this, we mean that we have to our disposal a numerical code[§] that given the value of the input parameters ξ evaluate the corresponding value of the quantity of interest $U(\xi)$.
  We are able to observe **the mapping from $U : \Xi \mapsto \mathcal{V}$ at selected values of $\xi \in \Xi$**[¶].
- Contrary to the Galerkin projection, Non-Intrusive methods can focus on the approximation of the QoI only.
- We do not need the full knowledge of the model equations, nor of all model unknowns.
- The Non-Intrusive approaches focus on observations of the mapping to construct the "best" approximation $U^{\mathcal{A}}(\xi)$.
- Classically, they are based on the minimization of the L_2-distance $U - U^{\mathcal{A}}$,

$$U^{\mathcal{A}} = \arg \min_{V \in \mathcal{V} \otimes \mathcal{S}_{\mathcal{A}}} \mathbb{E}\left[\|U - V\|_{\mathcal{V}}^2\right].$$

- They differ in the way this minimization problem is approximated.

[§] Could actually be an experimental device.

[¶] For this reason, the black-box is sometime called the oracle in machine learning theory.

Table of content

Basic setting

Consider a sample set of M realizations of the input parameter,

$$S_M = \left\{ \boldsymbol{\xi}^{(i)}, i = 1, \ldots, M \right\},$$

and the corresponding sample set of observations of the mapping $\Xi \mapsto \mathcal{V}$,

$$Y_M = \left\{ y^{(i)} \doteq U(\boldsymbol{\xi}^{(i)}), i = 1, \ldots, M \right\}.$$

The original minimization problem (for the L_2-distance)

$$U^{\mathcal{A}} = \arg \min_{V \in \mathcal{V} \otimes \mathcal{S}_{\mathcal{A}}} \mathbb{E}\left[\|U - V\|_{\mathcal{V}}^2 \right],$$

can be substituted for the following least-squares problem:

$$\hat{U}^{\mathcal{A}} = \arg \min_{V \in \mathcal{V} \otimes \mathcal{S}_{\mathcal{A}}} \frac{1}{M} \sum_{i=1}^{M} \|y^{(i)} - V(\boldsymbol{\xi}^{(i)})\|_{\mathcal{V}}^2.$$

In other words, we estimate the L_2 distance from the sample sets using the averaged sum of squared residuals.
Introducing the PC expansions, the problem can be recast in terms of the coefficients $\{u_\alpha \in \mathcal{V}, \alpha \in \mathcal{A}\}$:

$$\{u_\alpha, \alpha \in \mathcal{A}\} = \arg \min_{\{v_\alpha \in \mathcal{V}, \alpha \in \mathcal{A}\}} \frac{1}{M} \sum_{i=1}^{M} \left\| y^{(i)} - \sum_{\alpha \in \mathcal{A}} v_\alpha \Psi_\alpha(\boldsymbol{\xi}^{(i)}) \right\|_{\mathcal{V}}^2.$$

Least-squares problem

For simplicity, let us take $\mathcal{V} = \mathbb{R}^{\|}$, so the we have to minimize the LS functional

$$LS(v_\alpha, \alpha \in \mathcal{A}) = \frac{1}{M} \sum_{i=1}^{M} \left| y^{(i)} - \sum_{\alpha \in \mathcal{A}} v_\alpha \Psi_\alpha^{(i)} \right|^2, \quad \Psi_\alpha^{(i)} \doteq \Psi_\alpha(\boldsymbol{\xi}^{(i)}).$$

The optimality conditions, $\partial LS / \partial v_\alpha = 0$, yield the linear problem satisfied by the solution

$$\frac{1}{M} \sum_{i=1}^{M} \left[\Psi_\beta^{(i)} \left(y^{(i)} - \sum_{\alpha \in \mathcal{A}} u_\alpha \Psi_\alpha^{(i)} \right) \right] = 0, \quad \forall \beta \in \mathcal{A}.$$

Denoting $[Z] \in \mathbb{R}^{M \times |\mathcal{A}|}$ the matrix with entries $Z_{i,\alpha} = \Psi_\alpha^{(i)}$, the optimization problem can be rewritten as a linear system:

$$\frac{1}{M}[Z]^T[Z]\boldsymbol{u} = \frac{1}{M}[Z]^T\boldsymbol{y}, \quad \boldsymbol{u} = (u_\alpha)^T \in \mathbb{R}^{|\mathcal{A}|}, \quad \boldsymbol{y} = (y^{(0)} \cdots v^{(M)})^T \in \mathbb{R}^M.$$

The Fisher matrix $[F] = \frac{1}{M}[Z]^T[Z] \in \mathbb{R}^{|\mathcal{A}| \times |\mathcal{A}|}$ plays a crucial role in the conditioning of the least-squares problem. Clearly $[F]$ **must be invertible**.

Limsi

[‖] And for $\mathcal{V} = \mathbb{R}^n$?

Least-squares projection operator

We the haven to solve

$$[F]\boldsymbol{u} = \frac{1}{M}[Z]^T\boldsymbol{y}, \quad [F] = \frac{1}{M}[Z]^T[Z],$$

for the vector $\boldsymbol{u} \in \mathbb{R}^{|\mathcal{A}|}$ of PC coefficients.
The conditioning of the problem depends on the spectrum of the Fisher matrix, through the matrix $[Z]$.
In fact $[Z]$ defines **an orthogonal projection operator** Π from $\mathbb{R}^M$ to the subspace spanned by the $|\mathcal{A}|$ columns of $[Z]$:

$$\Pi = [Z]([Z]T[Z])^{-1}[Z]^T.$$

The projector Π is **symmetric, idempotent ($\Pi\Pi = \Pi$),** and columns of $[Z]$ are Π-stable ($\Pi[Z] = [Z]$).
It follows that the solution $\boldsymbol{u}$ belongs to the subspace of $\mathbb{R}^{|\mathcal{A}|}$ spanned by the columns of $[Z]$.
Therefore, the approximation error, $R(\xi) = U(\xi) - \hat{U}^{\mathcal{A}}(\xi)$, will be orthogonal to $\mathcal{S}_{\mathcal{A}}$ only for $M \to \infty$ and *appropriate* selection of the sample points.

Typically, $M = k|\mathcal{A}|$ with $k = 3 - 5$ is used in practice, for degree based polynomial basis.

Sample points selection

The Fisher (or information) matrix has for entries

$$F_{\alpha\beta} = \frac{1}{M}\sum_{i=1}^{M} \Psi_\alpha(\boldsymbol{\xi}^{(i)})\Psi_\beta(\boldsymbol{\xi}^{(i)}).$$

It shows that the conditioning of the problem depends on the sample set (its dimension M and selected points) and the basis through the definition of $\mathcal{A}$.

- If the sample points $\boldsymbol{\xi}^{(i)}$ are drawn at random from the distribution $p_{\boldsymbol{\xi}}$, then

$$\lim_{M\to\infty} F_{\alpha\beta} = \langle \Psi_\alpha, \Psi_\beta \rangle \Rightarrow \lim_{M\to\infty} [F] = \mathrm{Diag}\left(\langle \Psi_\alpha, \Psi_\beta \rangle\right),$$

  so $[F]$ is invertible for sufficiently large M.
- If the sampling does not follow $p_{\boldsymbol{\xi}}$ the LS problem can be modified to consider the weighted sum of squared residuals:

$$\{u_\alpha, \alpha \in \mathcal{A}\} = \arg \min_{\{v_\alpha, \alpha\in\mathcal{A}\}} \sum_{i=1}^{M} \omega_i \left| y^{(i)} - \sum_{\alpha\in\mathcal{A}} v_\alpha \Psi_\alpha(\boldsymbol{\xi}^{(i)}) \right|^2 .$$

  This latter form has connections with the NISP method (yet to be introduced), for appropriate selection of the sampling points and associated weights**.

** What would be appropriate?

Design of Experiments

The convergence of $\lim_{M\to\infty}[F] = \mathrm{Diag}\left(\left\langle \Psi_\alpha, \Psi_\beta \right\rangle\right)$ is however slow for a random sampling ($\mathcal{O}(1/\sqrt{M})$). It suggests that other types of sampling strategies (*e.g.* deterministic ones) can be more efficient.

Optimal Design of Experiments aims at optimizing the spectral properties of $[F]$ (or Π), for a fixed sample set dimension M and set of basis functions $\{\Psi_\alpha, \alpha \in \mathcal{A}\}$. Classically, it is based on the optimization with respect to S_M:

Name	Objective	Object
A-optimality	minimize the trace	$(Z^t Z)^{-1}$
D-optimality	maximize determinant	$Z^t Z$
E-optimality	maximize lower singular value	$Z^t Z$
G-optimality	minimize largest diagonal term	Π

Such optimization problems are very hard and are usually solved using stochastic tools: a large number of sample sets S^M are generated, optimize by moving the points individually until a local optimum is reached, and the best set is retained. This sample set can then be reused since its optimality does not depend on $U(\xi)$.

Pulkelsheim, F.: Optimal Design of Experiments. Classics in Applied Mathematics, vol. 50. SIAM, Philadelphia (2006)
Hardin, R., Sloane, N.: A new approach to the construction of optimal designs. J. Stat. Plann. Inference 37, 339–369 (1993)

Over-fitting

- **Over-fitting** occurs when a too low number of samples is used with respect to the polynomial degree of the basis††, in particular in presence of noise in the observed mapping.
  The LS solution $\hat{U}^{\mathcal{A}}$ effectively reduces the residual,

$$LS\left(v_\alpha, \alpha \in \mathcal{A}\right) = \frac{1}{M} \sum_{i=1}^{M} \left| y^{(i)} - \sum_{\alpha \in \mathcal{A}} v_\alpha \Psi_\alpha^{(i)} \right|^2$$

  but is far from the optimum of the L_2-distance problem

$$\mathbb{E}\left[|U - \hat{U}^{\mathcal{A}}|^2 \right] >> \min_{V \in \mathcal{S}_{\mathcal{A}}} \mathbb{E}\left[|U - V|^2 \right].$$

- The empirical error $LS(\hat{U}^{\mathcal{A}})$ is not a safe indicator of the approximation quality.
- The later can be estimated using a second sample set: **cross-validation**.
- Alternatively, over-fitting can be detected using resampling (bagging) technics, such as the **Leave-One-Out** (LOO), where the stability of the approximation is verified. If not, M must be increased, $\mathcal{A}$ reduced or the LS problem regularized.

Picard, R., Cook, D., Cross-Validation of Regression Models. Journal of the American Statistical Association 79 (387): 575-583 (1994)
Devijver, P.A., Kittler, J., Pattern Recognition: A Statistical Approach. Prentice-Hall, London, GB, (1982)

†† Pretty much similar to aliasing error in spectral methods.

Regularization of LS problem

If only a low number M of sampling points are available, compared to $|\mathcal{A}|$ a regularization of the LS problem may be necessary.

- L_2 Tikhonov regularization: the LS problem is completed by a regularization term:

$$\boldsymbol{u} = \arg\min_{\boldsymbol{v}} \|[Z]\boldsymbol{v} - \boldsymbol{y}\|^2 + \|[\Gamma]\boldsymbol{v}\|^2 ,$$

  with now the regularized solution

$$\boldsymbol{u} = \left([Z]^T[Z] + [\Gamma]^T[\Gamma]\right)^{-1} [Z]^T\boldsymbol{y}.$$

  Typical choice for the regularization matrix $[\Gamma]$ is

$$[\Gamma] \propto \mathrm{Diag}\left(\langle \Psi_\alpha, \Psi_\alpha \rangle\right),$$

  giving solution with lower 2nd moment.
- Suitable regularization matrix can be defined *a priori*, for instance if one has information regarding the decay rate of the spectrum of $U(\boldsymbol{\xi})$.
- Alternatively, $[\Gamma]$ can be optimized (over a prescribed family) using a cross-validation sample set.

Compressive Sensing - ℓ_1 minimization

- If $M < |\mathcal{A}|$, the LS problem is clearly underdetermined (there multiple solutions).
- However, in many situations, $U(\boldsymbol{\xi})$ has in fact a sparse representation in the basis of $\mathcal{S}_{\mathcal{A}}$[‡‡], meaning that many of the coefficients u_l in the expansion are negligible or zero.
- If the expansion of $U(\boldsymbol{\xi})$ in $\mathcal{S}_{\mathcal{A}}$ is K-sparse, that is $\|\boldsymbol{u}\|_{\ell_0} = K$, then the solution can be computed even for $K < M < |\mathcal{A}|$, provided the matrix $[Z]$ satisfy some technical properties. It suggests to determined the vector of expansion coefficients as the minimizer of the **constrained** optimization problem

$$\boldsymbol{u} = \arg\min_{\boldsymbol{v}} \|\boldsymbol{v}\|_{\ell_0} \quad s.t. \quad \|[Z]\boldsymbol{v} - \boldsymbol{y}\|^2 = 0.$$

- Further, it can be shown that the above problem is equivalent for some $\gamma > 0$ to the ℓ_1 minimization problem

$$u = \arg\min_{\boldsymbol{v}} \left\{ \|[Z]\boldsymbol{v} - \boldsymbol{y}\|^2 + \gamma \|\boldsymbol{u}\|_{\ell_1} \right\}.$$

- Several algorithms are available for the ℓ_1-minimization problems (LASSO, LARS see `http://www-stat.stanford.edu/~tibs/lasso.html`, Basis-Pursuit, ...)

[‡‡] Think of the expansion of an additive model in the basis for $\mathcal{A}_\infty^{\mathrm{No}}$.

Table of content

Orthogonal projection

The approximation

$$U^{\mathcal{A}}(\boldsymbol{\xi}) = \sum_{\alpha \in \mathcal{A}} u_\alpha \Psi_\alpha(\boldsymbol{\xi}),$$

minimizing the L_2-error,

$$\epsilon^2 = \mathbb{E}\left[\left\| U - U^{\mathcal{A}} \right\|_{\mathcal{V}}^2\right]$$

corresponds to the orthogonal projection of $U \in \mathcal{V} \otimes L_2(\Xi, p_{\boldsymbol{\xi}})$ onto $\mathcal{V} \otimes \mathcal{S}_{\mathcal{A}}$:

$$\mathbb{E}\left[\left(U - U^{\mathcal{A}}, V\right)_{\mathcal{V}}\right] = 0 \quad \forall V \in \mathcal{V} \otimes \mathcal{S}_{\mathcal{A}}.$$

Because $V \in \mathcal{V} \otimes \mathcal{S}_{\mathcal{A}}$ has for generic expansion $V = \sum_\alpha v_\alpha \Psi_\alpha$, and $\{\Psi_\alpha, \alpha \in \mathbb{N}^{\mathrm{N}}\}$ is an **orthogonal basis**, we immediately have the relations:
$\forall \alpha \in \mathcal{A}$, $v \in \mathcal{V}$:

$$\mathbb{E}\left[\sum_{\beta \in \mathcal{A}} (u_\beta \Psi_\beta, v\Psi_\alpha)_{\mathcal{V}}\right] = \sum_{\beta \in \mathcal{A}} (u_\beta, v)_{\mathcal{V}} \mathbb{E}\left[\Psi_\alpha \Psi_\beta\right] = (u_\beta, v)\left\langle \Psi_\beta, \Psi_\beta \right\rangle.$$

such that

$$u_\beta \left\langle \Psi_\beta, \Psi_\beta \right\rangle = \left\langle U, \Psi_\beta \right\rangle = \int_\Xi U(\boldsymbol{y}) \Psi_\beta(\boldsymbol{y}) p_{\boldsymbol{\xi}}(\boldsymbol{y}) d\boldsymbol{y}.$$

Non-Intrusive Spectral Projection

The NISP method uses the relations

$$u_\alpha = \frac{1}{\langle \Psi_\alpha, \Psi_\alpha \rangle} \langle U, \Psi_\alpha \rangle = \frac{1}{\langle \Psi_\alpha, \Psi_\alpha \rangle} \int_\Xi U(\boldsymbol{y}) \Psi_\beta(\boldsymbol{y}) p_{\boldsymbol{\xi}}(\boldsymbol{y}) d\boldsymbol{y},$$

to estimate the expansion coefficients of U.

- the constant $\langle \Psi_\alpha, \Psi_\alpha \rangle$ (norm of the polynomials) are known exactly.
- the coefficients u_α are independently computed.
- its amounts to the computation of N-dimensional integrals in a product space:

$$\int_\Xi U(\boldsymbol{y}) \Psi_\beta(\boldsymbol{y}) p_{\boldsymbol{\xi}}(\boldsymbol{y}) d\boldsymbol{y} = \int \dots \int U(y_1, \dots, y_{\mathrm{N}}) \Psi_\alpha(y_1, \dots, y_{\mathrm{N}}) p_1(y_1) \dots p_{\mathrm{N}}(y_{\mathrm{N}}) dy_1 \dots dy_{\mathrm{N}}.$$

Classically, the integrals are estimated by means of **numerical quadrature** or **sampling approaches**.

Monte-Carlo integration:

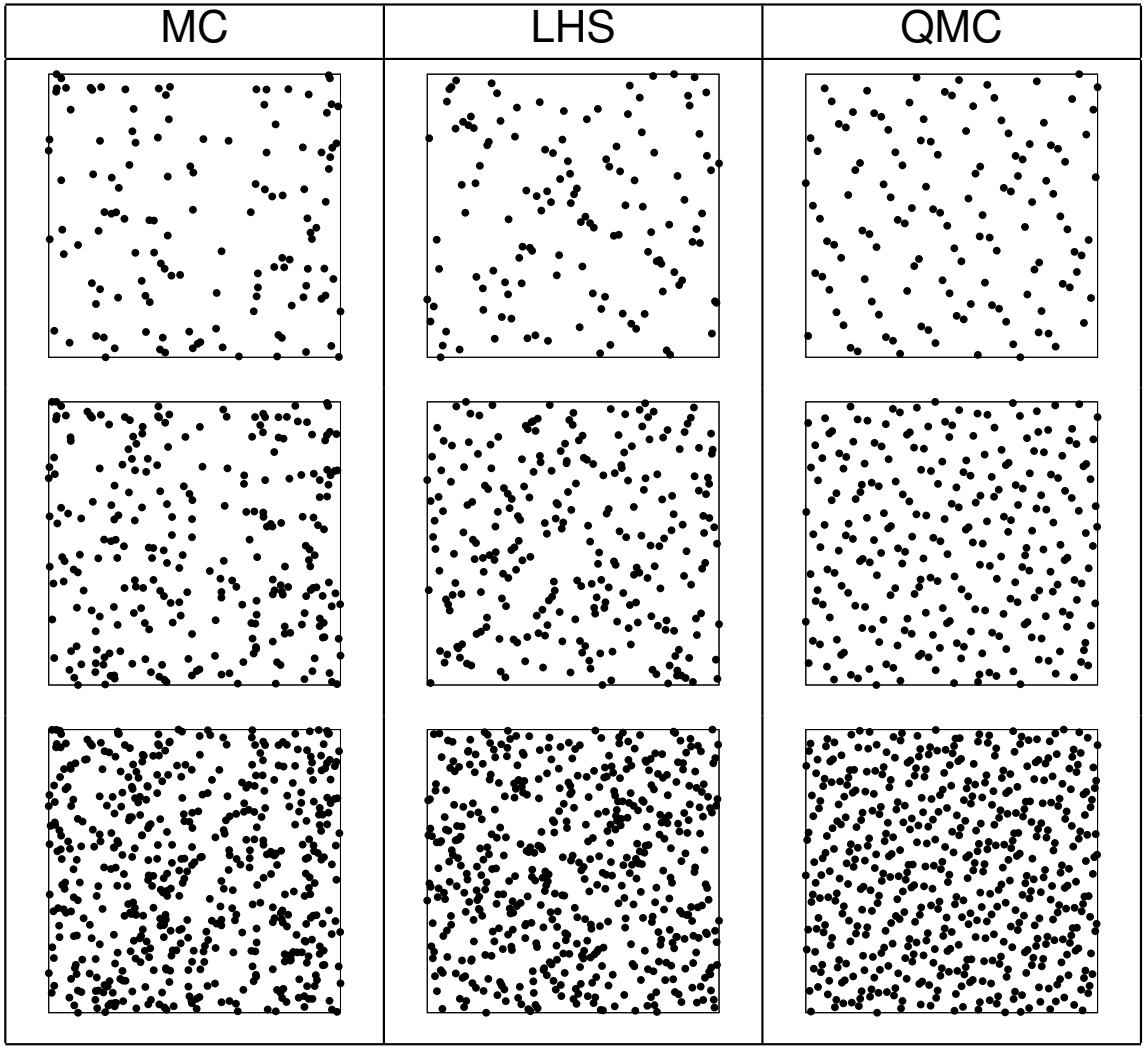
Estimate integrals from a random sample sets (MC and variants):

$$I_\alpha = \int_\Xi U(\xi)\Psi_\alpha(\boldsymbol{\xi})p_{\boldsymbol{\xi}}(\boldsymbol{\xi})d\boldsymbol{\xi} \approx I_\alpha^m = \frac{1}{m}\sum_{i=1}^{m} U(\boldsymbol{\xi}^{(i)})\Psi_\alpha(\boldsymbol{\xi}^{(i)}),$$

where $\{\boldsymbol{\xi}^{(i)}, i = 1, \ldots, m\}$ is a sample set drawn randomly (or pseudo-randomly) in Ξ according to the density $p_{\boldsymbol{\xi}}$.

- Error estimation:
$$\lim_{m\to\infty} |I_\alpha - I_\alpha^m| = \frac{\mathbb{V}\left[U\Psi_\alpha\right]}{\sqrt{m}}.$$
- Convergence rate independent of the regularity of the functional
- Convergence rate independent of the of the dimensionality
- **Slow convergence rate**
- Improved sampling strategy

Improved Monte-Carlo integration:



Deterministic Quadratures

The integrals can also be computed by means of **deterministic quadratures** involving a set of Nq quadrature points $\xi^i di$ and weights $w^{(i)}$:

$$\int_{\Xi} U(\xi)\Psi_\alpha(\xi) \approx \sum_{i=1}^{N_Q} w^{(i)} U(\xi^{(i)})\Psi_{\boldsymbol{\alpha}}\left(\xi^{(i)}\right).$$

One dimensional quadratures rules

$$\int f(x)dx \approx \sum_{i=1}^{n_q} f(x_i)w_i,$$

such as **mid-point rule**, Simpson rules, Gauss' quadratures, ..., can be tensorized. For instance, in the case of the same measure along the N-dimension:

$$\int \dots \int f(x_1,\dots,x_{\mathrm{N}})dx_1 \dots dx_{\mathrm{N}} \approx \sum_{i_1=1}^{n_q} \cdots \sum_{i_{\mathrm{N}}=1}^{n_q} f(x_{i_1},\dots,x_{i_{\mathrm{N}}})w_{i_1} \times \cdots \times w_{i_N},$$

requiring a total of n_q^{N} function evaluations.
Tensorization can use a different number of quadrature points along the different dimensions.

Multi-dimensional quadrature

Approximate integrals by N-**dimensional quadratures**:
Owing to the product structure of Ξ the quadrature points $\boldsymbol{\xi}^{(i)}$ and weights $w^{(i)}$ can be obtained by

- **full tensorization** of n points 1-D quadrature (*i.e.* Gauss):

$$N_Q = n^N$$

- **partial tensorization** of nested 1-D quadrature formula (Féjer, Clenshaw-Curtis) using Smolyak formula: [Smolyak, 63]

$$N_Q << n^N$$

The partial tensorization results in so-called Sparse-Grid cubature formula, that can be constructed adaptively to the integrants (anisotropic formulas) in order to account for variable behaviors along the stochastic directions. [Gerstner and Griebel, 2003]

- Important development of **sparse-grid methods**
- Anisotropy and adaptivity
- Also (sparse grid) collocation methods (N-dimensional interpolation) [Mathelin and Hussaini, 2003], [Nobile et al, 2008]

Recap.

We want to construct a functional approximation of a model-output U, where the model involves random parameters $\boldsymbol{\xi}$

$$\boldsymbol{\xi} \in \Xi \subseteq \mathbb{R}^N, \quad \text{with probability density function } p_{\boldsymbol{\xi}}.$$

The approximation is sought as

$$U(\boldsymbol{\xi}) \approx \hat{U}(\boldsymbol{\xi}) = \sum_{\boldsymbol{\beta} \in \mathcal{B}} u_{\boldsymbol{\beta}} \Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi}),$$

with $\mathcal{B}$ a multi-index set and $\{ \Psi_{\boldsymbol{\beta}} \}$ CONS.

You have seen different types of non-intrusive methods:

- **Non-Intrusive Spectral Projection**: compute $\{s_{\boldsymbol{\beta}}, \boldsymbol{\beta} \in \mathcal{B}\}$ by **numerical quadrature**, exploiting orthogonality of the PC $\Psi_{\boldsymbol{\beta}}$ (orthogonal projection on $\text{span}\{\Psi_{\boldsymbol{\beta}}\}$)
- **Least-Squares type**: compute $\{\boldsymbol{\beta}, \boldsymbol{\beta} \in \mathcal{B}\}$ by solving an **optimization problem** based on a set of observation points and possibly regularization techniques
- **Collocation**: use a set of model-output observations to construct an **interpolation**

Observe that the two first differ from the latter by the *a priori* / implicit selection of the approximation space.

Comments

Non-intrusive methods are very attractive by the fact they **reuse code and rely on deterministic computations**. However their **computational complexity** -measured as the number of deterministic simulations needed- quickly grows with the dimension N of the parameter space (and with $\mathrm{Card}(\mathcal{B})$).

This is particularly critical when one relies on **straightforward tensorization of one-dimensional objects** (quadrature or interpolation rules) to construct N-dimensional ones: complexity is then in $\mathcal{O}(C^{\mathrm{N}})$.

Sparse grid methods aim at reducing the complexity by relying on **smarter tensorization strategies**.

Hint: total degree truncation of the PC basis, instead of partial degree truncation, for the PC basis $\mathcal{S}^{\mathrm{P}} = \mathrm{span}\{\Psi_{\boldsymbol{\beta}}, \boldsymbol{\beta} \in \mathcal{B}\}$:

$$\mathrm{Card}\left\{\boldsymbol{\beta} \in \mathbb{N}^{\mathrm{N}}, \sum_{i=1}^{\mathrm{N}} \beta_i \leq \mathrm{No}\right\} \ll \mathrm{Card}\left\{\boldsymbol{\beta} \in \mathbb{N}^{\mathrm{N}}, \beta_{1 \leq i \leq \mathrm{N}} \leq \mathrm{No}\right\}.$$

Question: how to reuse the idea of sparse tensorization for quadrature or interpolation rules?
(Answer: Smolyak formula.)

Smolyak Formula

Sparse quadrature

We consider here **the cubature problem for NISP** where we need to approximate N-dimensional integrals of type

$$I_{\mathrm{N}}(f) = \int_0^1 \dots \int_0^1 f(x_1, \dots, x_{\mathrm{N}}) dx_1 \dots dx_{\mathrm{N}}.$$

This situation corresponds to $\Xi = [0,1]^{\mathrm{N}}$ and

$$p_{\boldsymbol{\xi}}(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{x} \in \Xi \\ 0, & \text{otherwise} \end{cases}.$$

Ideas and concepts of sparse grid immediately extend to collocation and integration, and to more general situations having **product structures**,

$$\mathbb{R}^{\mathrm{N}} \supseteq \Xi := \Xi_1 \times \dots \times \Xi_{\mathrm{N}}, \quad p_{\boldsymbol{\xi}}(x_1, \dots, x_{\mathrm{N}}) := p_{\xi_1}(x_1) \times \dots \times p_{\xi_{\mathrm{N}}}(x_{\mathrm{N}}),$$

using *ad-hoc* one-dimensional quadrature/interpolation rules along each direction $1 \leq j \leq \mathrm{N}$.

Sparse quadrature

Consider the sequence of **nested one-dimensional** quadrature formulas for $l = 1, 2, \ldots$

$$I_1(f) = \int_0^1 f(x)dx \approx I^{(l)}(f) = \sum_{q=1}^{Q(l)} w_q^{(l)} f(x_q^{(l)}),$$

such that $\{x_q^{(i)}, q = 1, \ldots Q(i)\} \subset \{x_q^{(j)}, q = 1, \ldots Q(j)\}$ for $1 \leq i < j$.

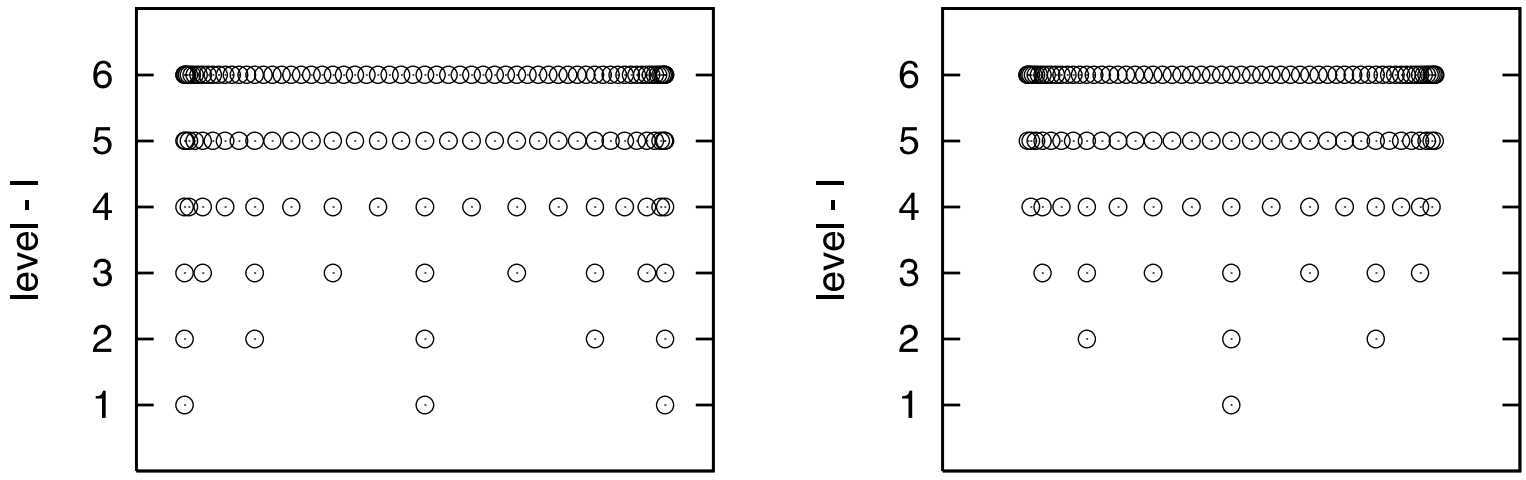


Fig. 3.3 Nodes of the Clenshaw-Curtis (*left*) and Fejèr (*right*) rules for levels $1 \leq l \leq 6$

Owing to the **nested nature**, increasing l to $l + 1$ introduces additional points and changes the weights associated to the older ones.

Sparse quadrature

Consider the sequence of **nested one-dimensional** quadrature formulas for $l = 1, 2, \ldots$

$$I_1(f) = \int_0^1 f(x)dx \approx I^{(l)}(f) = \sum_{q=1}^{Q(l)} w_q^{(l)} f(x_q^{(l)}),$$

such that $\{x_q^{(i)}, q = 1, \ldots Q(i)\} \subset \{x_q^{(j)}, q = 1, \ldots Q(j)\}$ for $1 \leq i < j$.

The fully tensorized N-dimensional quadrature formula at level l has for expression

$$\begin{aligned} I_{\mathrm{N}}(f) \approx I_{\mathrm{N}}^{F(l)}(f) &:= \left(I^{(l)} \otimes \cdots \otimes I^{(l)}\right)(f) \\ &= \sum_{q_1=1}^{Q(l)} \cdots \sum_{q_{\mathrm{N}}=1}^{Q(l)} \left(w_{q_1}^{(l)} \times \cdots \times w_{q_{\mathrm{N}}}^{(l)}\right) f(x_{q_1}^{(l)}, \ldots, x_{q_{\mathrm{N}}}^{(l)}). \end{aligned}$$

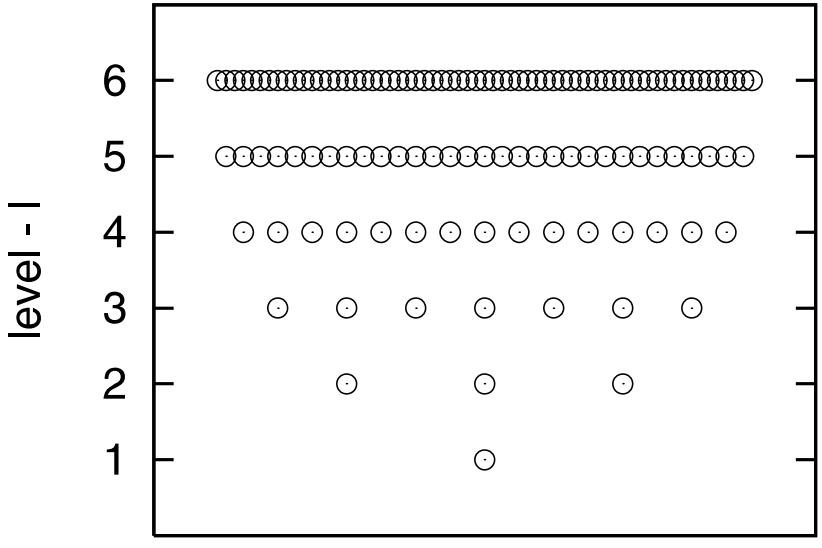
This formula has $Q_F^{\mathrm{N}}(l) = Q(l)^{\mathrm{N}}$ points with **positive** weights $w_{q_1}^{(l)} \times \cdots \times w_{q_{\mathrm{N}}}^{(l)}$ provided the one-dimensional sequence has positive weights $w_q^{(l)} > 0$!

Sparse quadrature

Back to the nested one-dimensional formulas

Denote $\Delta^{(l)} f$ **the difference formula** between levels $l-1$ and l,

$$\Delta^{(l)} f := \left(I^{(l)} - I^{(l-1)} \right) (f) = \sum_{q=1}^{Q(l)} \Delta w_q^{(l)} f(x_q^{(l)}), \quad \Delta^{(0)} f := 0.$$



Sparse quadrature

Back to the nested one-dimensional formulas

Denote $\Delta^{(l)} f$ **the difference formula** between levels $l-1$ and l,

$$\Delta^{(l)} f := \left(I^{(l)} - I^{(l-1)} \right)(f) = \sum_{q=1}^{Q(l)} \Delta w_q^{(l)} f(x_q^{(l)}), \quad \Delta^{(0)} f := 0.$$

Clearly, the one-dimensional quadrature formula at level l is expressed as

$$I^{(l)}(f) = \sum_{i=1}^{l} \Delta^{(i)}(f).$$

Observe: for nested formulas that exactly integrate constants,

$$\sum_{q=1}^{Q(l)} \Delta w_q^{(l)} = \begin{cases} 1, & l = 1 \\ 0, & l > 1 \end{cases}$$

$\Rightarrow$ **positive weights $w_q^{(l)}$ do not imply $\Delta w_q^{(l)} \geq 0$ for $l > 1$.**

Differences Formula

For the construction of sparse N-dimensional cubatures, we introduce the **multi-index** $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N) \in (\mathbb{N}^*)^N$, and use the norms

$$|\boldsymbol{\alpha}|_{\ell_1} = \sum_{i=1}^{N} |\alpha_i|, \quad |\boldsymbol{\alpha}|_{\ell_\infty} = \max_{1 \leq i \leq N} |\alpha_i|.$$

The **fully tensorized formula** can be recast as

$$I_N^{F(l)}(f) := \left(I^{(l)} \otimes \cdots \otimes I^{(l)}\right)(f) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}_\infty(l)} \left(\Delta^{(\alpha_1)} \otimes \cdots \otimes \Delta^{(\alpha_N)}\right)(f),$$

where the summation is over the multi-index set

$$\mathcal{A}_\infty(l) := \{\boldsymbol{\alpha} \in (\mathbb{N}^*)^N, |\boldsymbol{\alpha}|_{\ell_\infty} \leq l\},$$

or explicitly,

$$\left(\Delta^{(\alpha_1)} \otimes \cdots \otimes \Delta^{(\alpha_N)}\right)(f) = \sum_{q_1=1}^{Q(\alpha_1)} \cdots \sum_{q_N=1}^{Q(\alpha_N)} \left(\Delta w_{q_1}^{\alpha_1} \times \cdots \times \Delta w_{q_N}^{\alpha_N}\right) f(x_{q_1}^{\alpha_1}, \ldots, x_{q_N}^{\alpha_N}).$$

Summation of tensored differences to FT formula

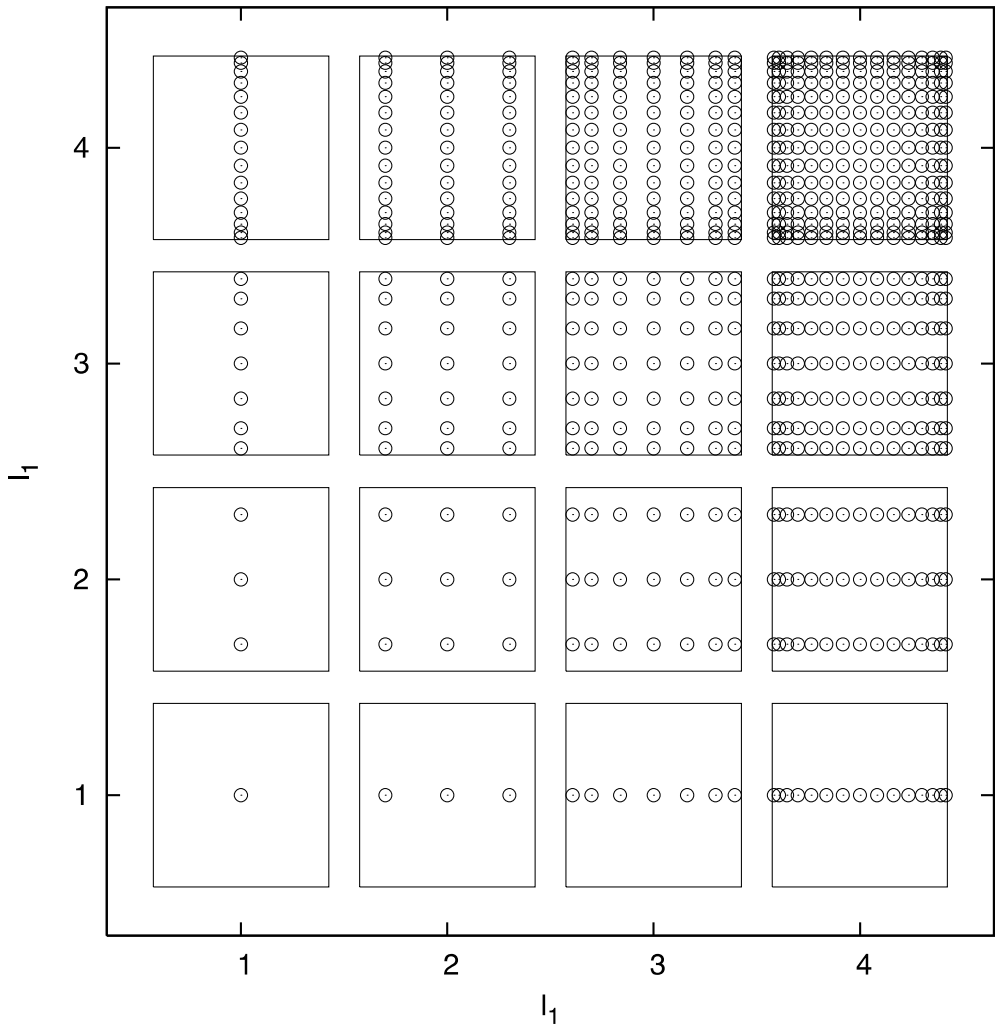


Fig. 3.4 Illustration of cubature rules constructed by products of nested Fejèr quadratures: plotted are the 2D grids of integration nodes from (3.25) for different values of the levels l_1 and l_2 along the integration dimensions. Grids on the diagonal plots correspond to the definition (3.23) of the cubature

The Smolyak formula (1963) is constructed by defining a new set of multi-indices for the summation of tensored difference formulas; specifically the **Smolyak formula at level** l is given by

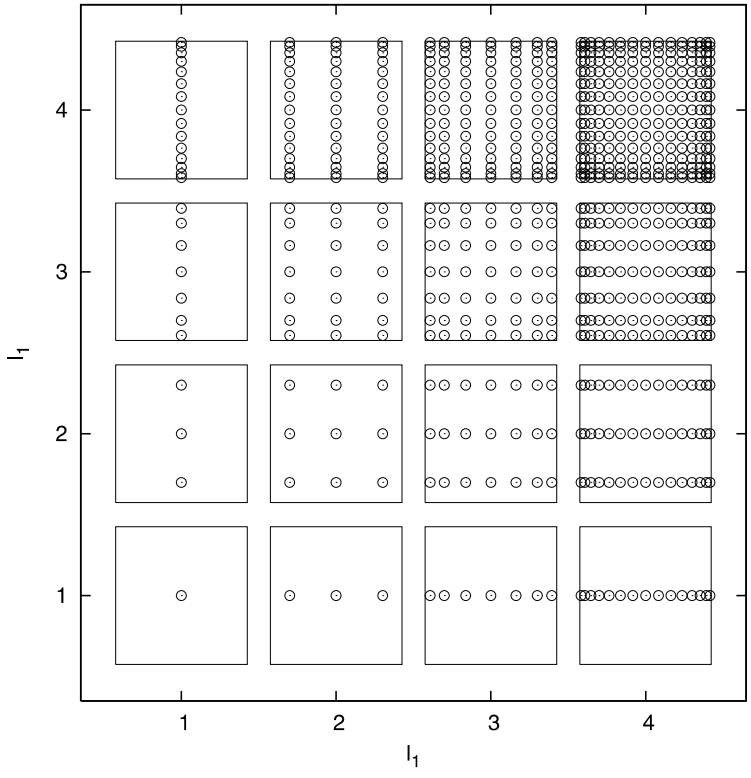
$$I_{\mathrm{N}}^{S(l)}(f) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}_S(l)} \left(\Delta^{(\alpha_1)} \otimes \cdots \otimes \Delta^{(\alpha_{\mathrm{N}})} \right)(f),$$

where the summation is now over the multi-index set

$$\mathcal{A}_S(l) := \left\{ \boldsymbol{\alpha} \in (\mathbb{N}^*)^{\mathrm{N}}, |\boldsymbol{\alpha}|_{\ell_1} \leq l + N - 1 \right\} \subset \mathcal{A}_\infty(l).$$

This is essentially the idea of the total order truncation -as opposed to the partial order truncation- for polynomial bases, but here applied to the **partial tensorization of one-dimensional quadrature formulas**.

Comparison of FT and Smolyak cubature rules



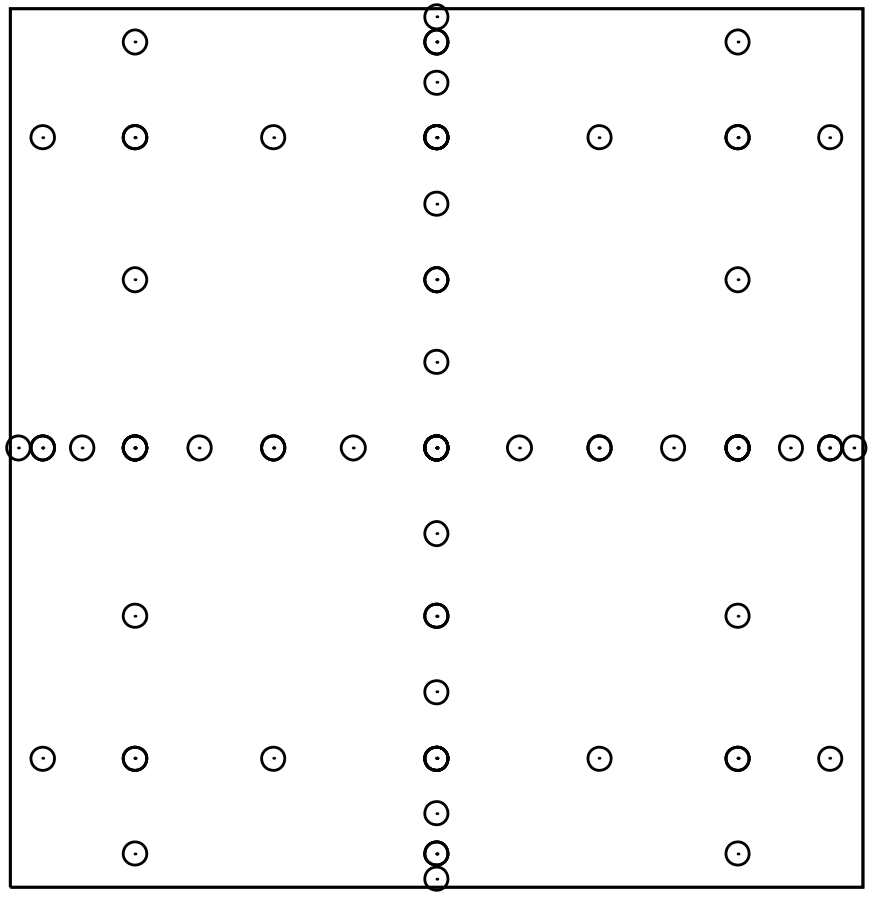


Fig. 3.4 Illustration of cubature rules constructed by products of nested Fejèr quadratures: plotted are the 2D grids of integration nodes from (3.25) for different values of the levels l_1 and l_2 along the integration dimensions. Grids on the diagonal plots correspond to the definition (3.23) of the cubature

Comparison of FT and Smolyak cubature rules

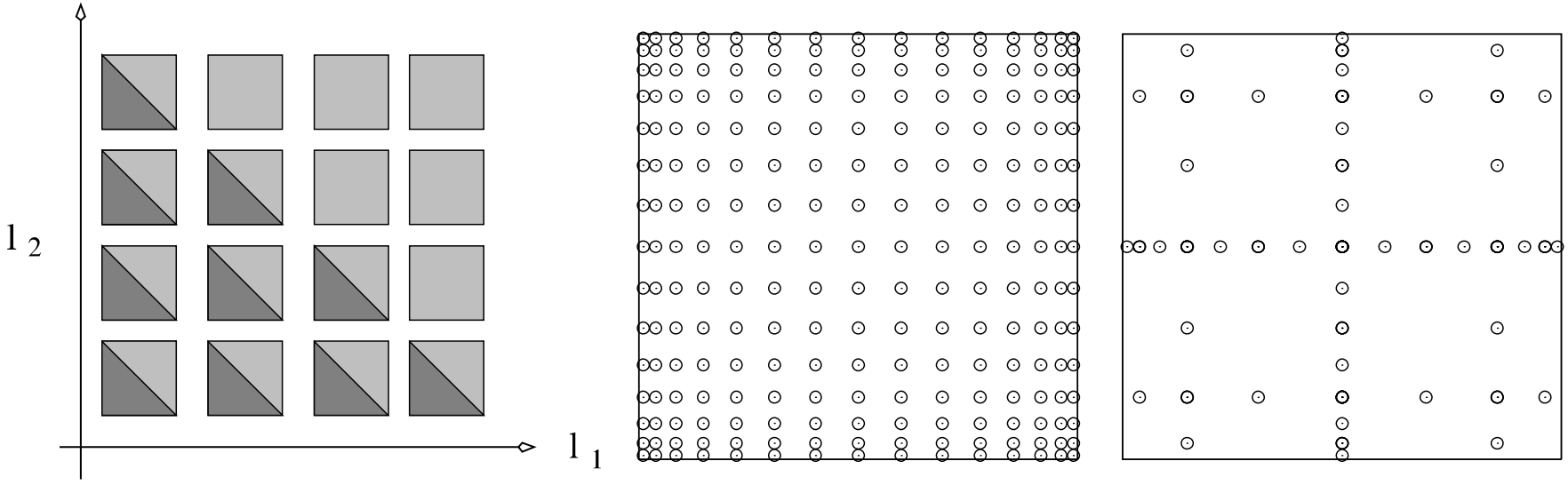


Fig. 3.5 Comparison of product and sparse tensorizations in the construction of cubature formulas of level $l = 4$, for the numerical integration in $N = 2$ dimensions. The *left plot* shows the indexes of the summation of difference formulas $\Delta_k^{(N)}$ for the product form in (3.30) (*squares*) and Smolyak's algorithm in (3.29) (*triangle*). The resulting grids for the Fejèr nested quadrature rule are shown in the *middle* (product form) and *right* (sparse grid) *plots*

Sparse grids in 2 and 3-D

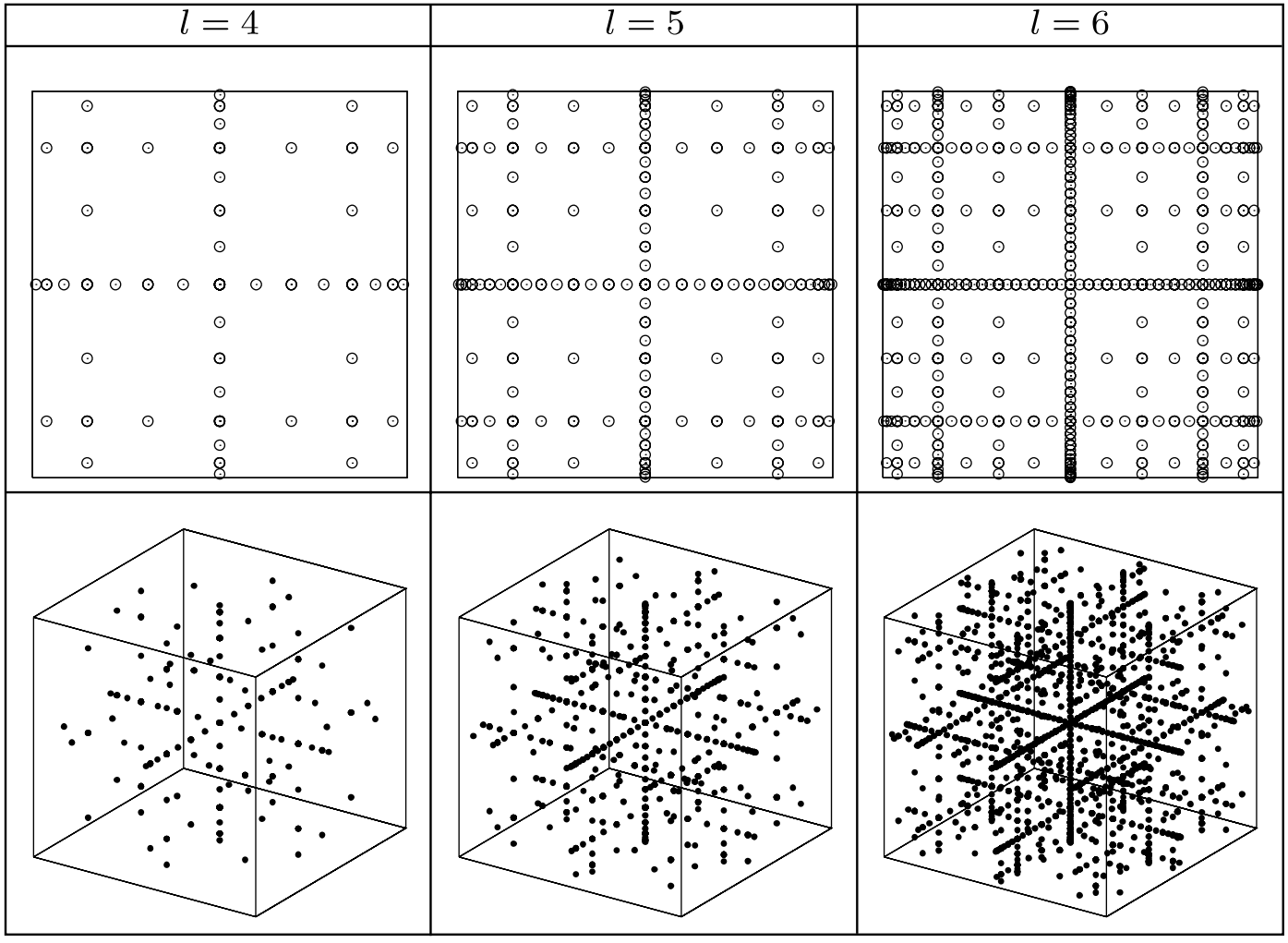
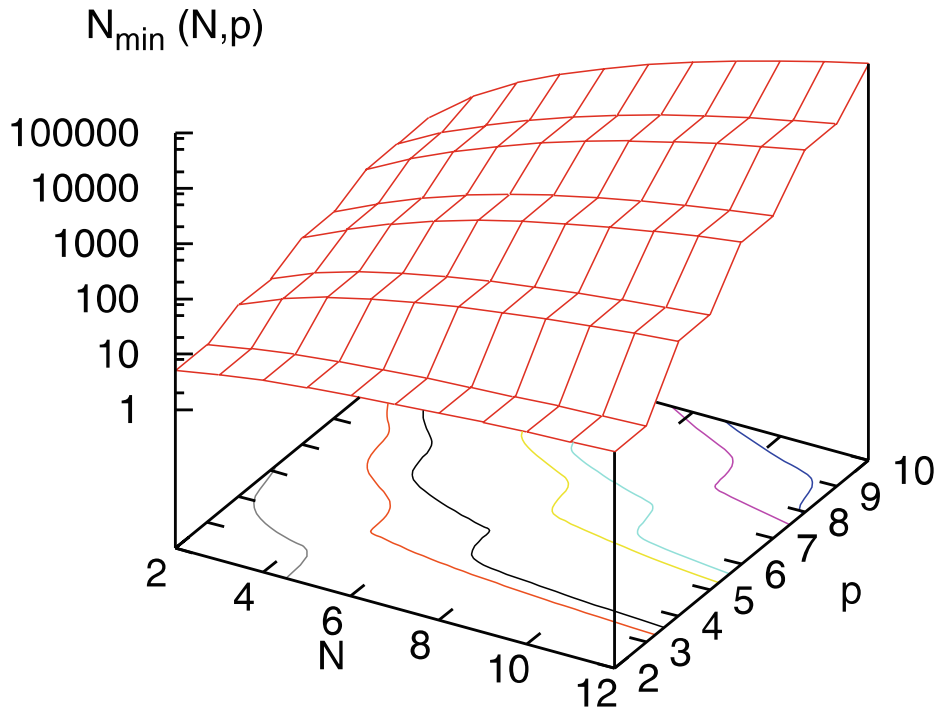


Fig. 3.6 Illustration of the sparse grid cubature nodes in N = 2 and N = 3 dimensions for the Smolyak's method and nested Fejèr quadrature formulas. Different levels l are considered as indicated

Number of points in Sparse grids (Nested Clenshaw-Curtis quadrature rule)

Fig. 3.7 Minimum number of nodes $N_{\min}$ for the Smolyak's sparse cubature for exact integration of polynomial integrands with degree $\leq p$ over hypercubes with uniform weight (nested Clenshaw-Curtis rules and Smolyak's sparse tensorization)



Comments & Remarks

Observe: the support points of the tensored difference formula associated to the multi-index $\boldsymbol{\alpha}$ are a subset of those associated to $\boldsymbol{\beta} \geq \boldsymbol{\alpha}$ (that is $\alpha_i \leq \beta_i$ for $i = 1, \ldots, \mathrm{N}$), owing to the nested nature of the one-dimensional formulas.

In practice, given $l > 1$ the Smolyak formula is recast as a weighted-sum,

$$I_{\mathrm{N}}^{S(l)}(f) = \sum_{q=1}^{Q_S^{\mathrm{N}}(l)} w_q^{S(l),\mathrm{N}} f(\boldsymbol{x}_q^{S(l),\mathrm{N}}).$$

- increasing level l introduces additional points and change the weights
- Fast algorithm for the computation of points and weights are mandatory when $\mathrm{N} > 10$
- $Q_S^{\mathrm{N}}(l) \ll Q_F^{\mathrm{N}}(l)$ as $\mathrm{N} \uparrow$

Comments & Remarks

- The sparse tensorization **tempers the curse of dimensionality**
- **The Smolyak cubature is less accurate** than the fully-tensored formula (for the same level)
- **Question:** what's the polynomial space Π^N for which the Smolyak formula is exact?
- The Smolyak cubature **does not define a discrete inner product**. Why ?
- **What about collocation methods?**

Regarding **Non-Instrusive Spectral Projection**

- Use the same sparse rule for all integrands $(U(\boldsymbol{\xi})\Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi}))$, $\boldsymbol{\beta} \in \mathcal{B}$: Direct NISP
- Determine $\mathcal{B}$ such that $\forall \boldsymbol{\beta}, \boldsymbol{\beta}' \in \mathcal{B}$ the cubature **exactly** integrates $(\Psi_{\boldsymbol{\beta}}\Psi_{\boldsymbol{\beta}'})$: internal-aliasing free NISP
- Alternatively, consider differences formulas for the Fully-Tensored NISP projection at different level l, resulting in the use of **F-T quadratures depending on $\boldsymbol{\beta}$**. Internal aliasing-free while allowing for larger set $\mathcal{B}$. Pseudo-Spectral NISP
  [Marzouk, 2013], [Constantine, 2013]
- **External aliasing remains!**

Adaptive Sparse Grid

Generalization of Smolyak formula

One can consider general classes of cubature formulas through the generic expression

$$I_N^{\mathcal{A}}(f) = \sum_{\boldsymbol{\alpha} \in \mathcal{A}} \left(\Delta^{(\alpha_1)} \otimes \cdots \otimes \Delta^{(\alpha_N)} \right) (f).$$

This calls for a definition of the multi-index set $\mathcal{A}$.

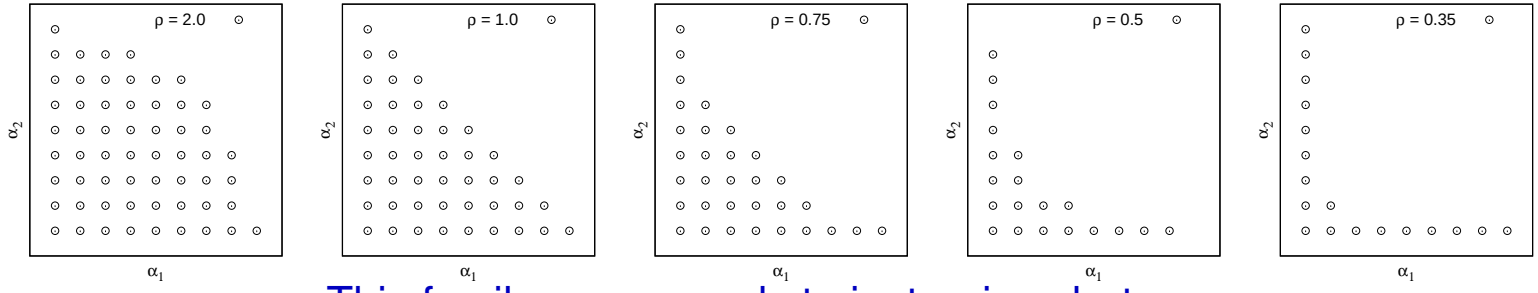
Ideas?

Generalization of Smolyak formula

$$I_N^{\mathcal{A}}(f) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \left(\Delta^{(\alpha_1)} \otimes \cdots \otimes \Delta^{(\alpha_N)}\right)(f)$$

ℓ_ρ**-(quasi)norm**: hyperbolic cross product

$$|\boldsymbol{\alpha}|_{\ell_\rho} := \left(\sum_{i=1}^{N} |\alpha_i - 1|^\rho\right)^{1/\rho}, \quad \mathcal{A}(\rho) := \left\{\boldsymbol{\alpha} \in (\mathbb{N}^*)^N, |\boldsymbol{\alpha}|_{\ell_\rho} < l\right\}, \quad \rho > 0.$$



This family corresponds to isotropic cubatures.

Generalization of Smolyak formula

$$I_N^{\mathcal{A}}(f) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \left(\Delta^{(\alpha_1)} \otimes \cdots \otimes \Delta^{(\alpha_N)}\right)(f)$$

Weighted ℓ_1-norms: dimension adaptivity / anisotropic rules

Let $W_{1\leq i\leq N} > 0$ be directional weights:

$$|\boldsymbol{\alpha}|_{\ell_1(\boldsymbol{w})} := \sum_{i=1}^{N} W_i|\alpha_i - 1|, \quad \mathcal{A} = \mathcal{A}(\boldsymbol{w}, C) := \{\boldsymbol{\alpha} \in (\mathbb{N}^*)^N, |\boldsymbol{\alpha}|_{\ell_1(\boldsymbol{w})} < C\}.$$

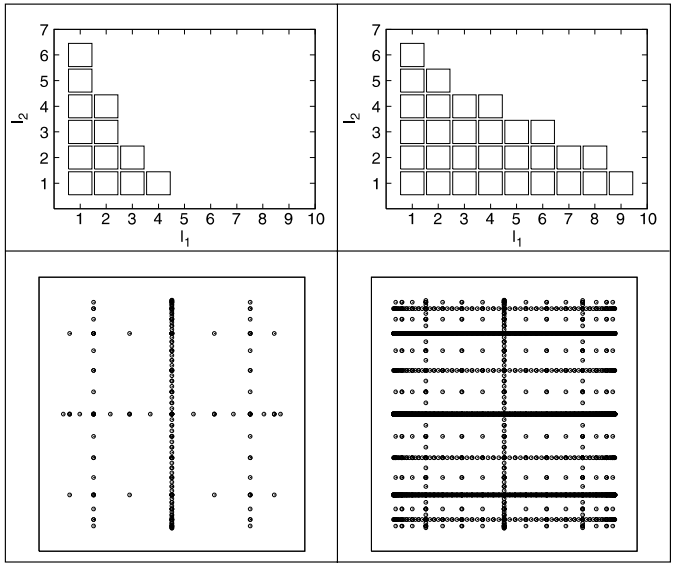


Fig. 3.8 Example of two-dimensional cubatures constructed with the dimension-adaptive strategy using $a_1 = 1.5$ (*left*) and $a_1 = 0.6$ (*right*) and $a_2 = 1 + (1 - a_1)/l$, $l = 6$. Plotted are the respective multi-index sets (*top row*, see (3.34)) and the corresponding sparse grids (*bottom row*, Fejèr nested nodes)

Admissible multi-index sets

What is the constraint on the structure of the multi-index set $\mathcal{A}$? Why?

Admissible multi-index sets

What is the constraint on the structure of the multi-index set $\mathcal{A}$? Why?

Admissible sets: $\mathcal{A}$ is said admissible if **all $\alpha \in \mathcal{A}$ has predecessors** in all the N directions:

$$\forall \boldsymbol{\alpha} \in \mathcal{A},\ \alpha_j > 1 \Rightarrow \boldsymbol{\alpha} - \boldsymbol{e}_j \in \mathcal{A}, \quad j = 1, \ldots, \mathrm{N},$$

where $\boldsymbol{e}_j$ is the unit vector in direction j,

$$(\boldsymbol{e}_j)_i = \begin{cases} 1, & i = j \\ 0, & i \neq j \end{cases}.$$

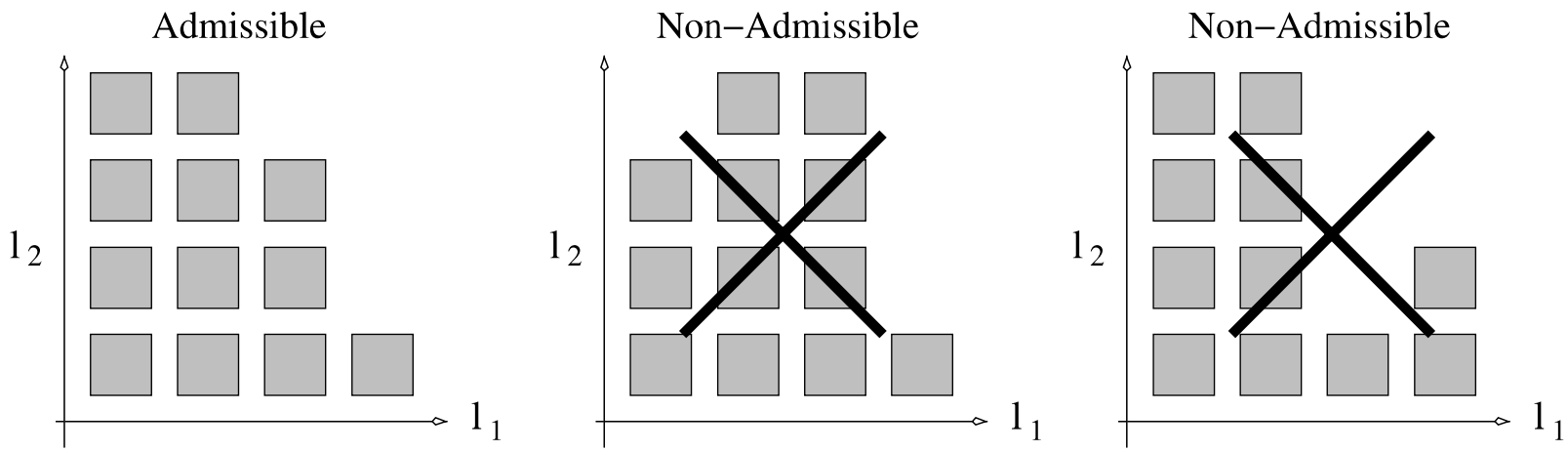


Fig. 3.9 Examples of admissible and non-admissible multi-index sets in two dimensions

Forward neighborhood and candidate set

Given a multi-index $\boldsymbol{\alpha}$, we define its **forward neighborhood** as the multi-index set

$$\mathcal{F}(\boldsymbol{\alpha}) = \{\boldsymbol{\alpha} + \boldsymbol{e}_j, j = 1, \ldots, \mathrm{N}\}.$$

Given an **admissible** multi-index set $\mathcal{A}$, we define its admissible forward multi-index set $\mathcal{C}$ as

$$\mathcal{C}(\mathcal{A}) := \{\boldsymbol{\alpha} \in (\mathbb{N}^*)^{\mathrm{N}}, \boldsymbol{\alpha} \notin \mathcal{A} \text{ and } \mathcal{A} \cup \{\boldsymbol{\alpha}\} \text{ admissible}\}.$$

Clearly,

$$\forall \boldsymbol{\alpha} \in \mathcal{C}(\mathcal{A}), \exists \boldsymbol{\beta} \in \mathcal{A} \text{ such that } \boldsymbol{\alpha} \in \mathcal{F}(\boldsymbol{\beta}).$$

Adaptive strategy:
The multi-index set of the adapted cubature for the approximation of $I_N(f)$ is constructed by building a sequence of admissible sets $\mathcal{A}^{(0)} \to \mathcal{A}^{(1)} \to \mathcal{A}^{(2)} \ldots$, such that

$$\mathcal{A}^{(k+1)} = \mathcal{A}^{(k)} \cup \boldsymbol{\alpha}_k, \quad \boldsymbol{\alpha}_k \in \mathcal{C}(\mathcal{A}^{(k)}).$$

In words, given $\mathcal{A}^{(k)}$, a new tensorization is added (one at a time) that leaves the resulting set admissible.

How to pick $\boldsymbol{\alpha}_k \in \mathcal{C}(\mathcal{A}^{(k)})$?

Error indicator

For a multi-index $\boldsymbol{\alpha} \in (\mathbb{N}^*)^N$, we define the associated **excess** as

$$e(\boldsymbol{\alpha}) = \left|\left(\Delta^{\alpha_1} \otimes \cdots \otimes \Delta^{\alpha_N}\right)(f)\right|.$$

To enrich $\mathcal{A}^{(k)}$, we should choose $\boldsymbol{\alpha}_k \in \mathcal{C}(\mathcal{A}^{(k)})$ corresponding to the largest excess $e_{\boldsymbol{\alpha}}$,

$$\boldsymbol{\alpha}_k = \arg \max_{\boldsymbol{\alpha} \in \mathcal{C}(\mathcal{A}^{(k)})} \left|\left(\Delta^{\alpha_1} \otimes \cdots \otimes \Delta^{\alpha_N}\right)(f)\right|.$$

However, the objective of the adaptation is to reduce the error for the least possible number of function evaluations, so we also want to **balance the excess with computational complexity of the new tensorization**. Gerstner and Griebel proposed to pick $\boldsymbol{\alpha}_k$ from

$$\boldsymbol{\alpha}_k = \arg \max_{\boldsymbol{\alpha} \in \mathcal{C}(\mathcal{A}^{(k)})} \max\left((1-\rho)e(\boldsymbol{\alpha}), C/N_Q(\boldsymbol{\alpha})\right), \quad C \in [0,1],$$

where $N_Q(\boldsymbol{\alpha}) = \prod_i N_Q^{\alpha_i}$ is the number of points in the tensored difference formula.

- for $C = 0$, the adaptation only considers the excess,
- for $C = 1$, the adaptation only considers the complexity.

A stopping criteria ?

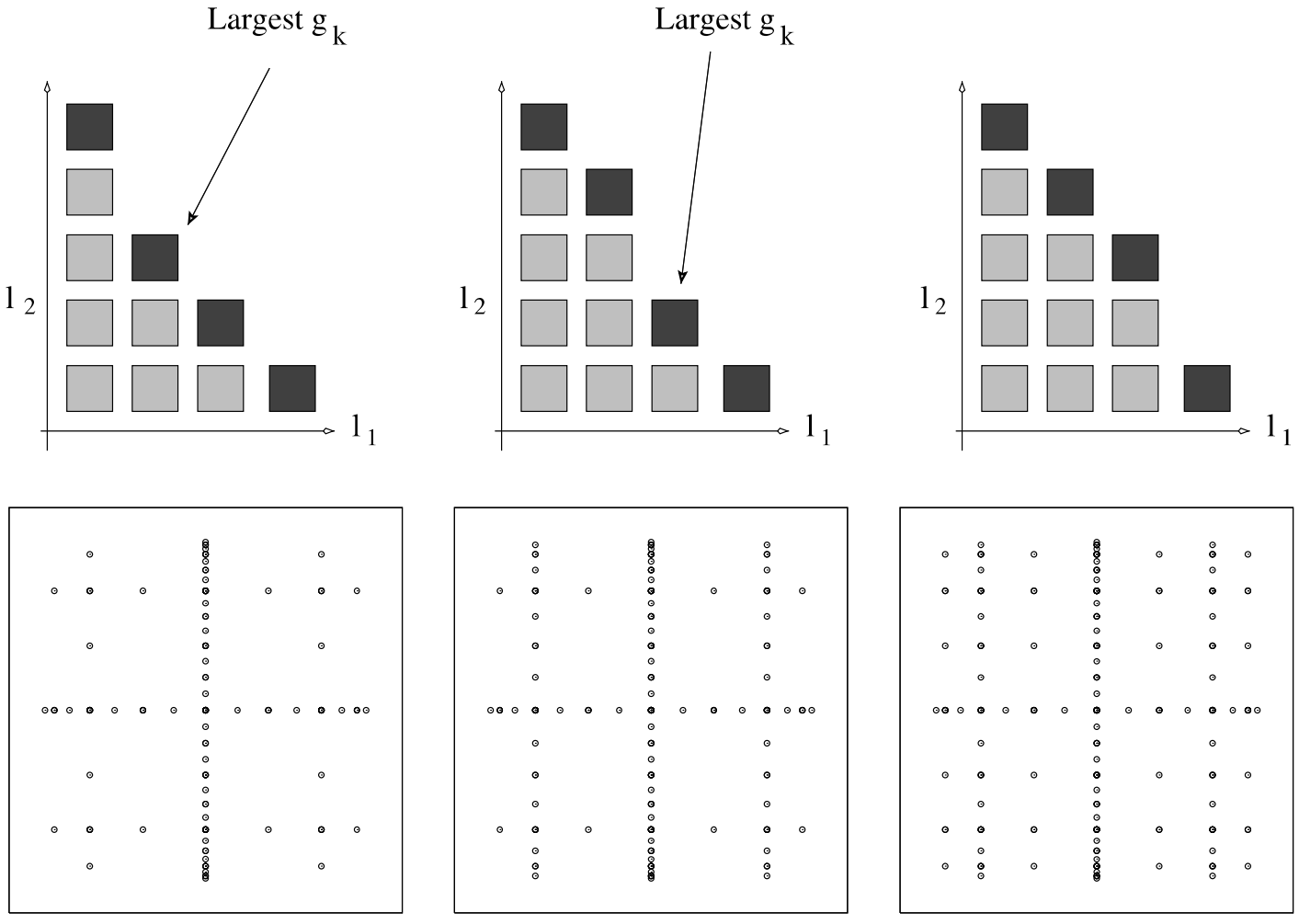


Fig. 3.10 Illustration of the adaptive sparse grid procedure for N = 2. The plots on the *top row* show the evolution of the multi-index set $\mathcal{I}$, distinguishing the sets of old multi-indexes $\mathcal{O}$ (*light gray squares*) and active multi-indexes $\mathcal{A}$ (*dark gray squares*). The corresponding sparse grids are plotted in the *bottom row*

For NISP, one uses a cubature formula

$$I_N^{\mathcal{A}}(f) = \sum_{\boldsymbol{\alpha}\in\mathcal{A}} \left(\Delta^{(\alpha_1)} \otimes \cdots \otimes \Delta^{(\alpha_N)}\right)(f),$$

to compute a **series of integrals** with $f = f_{\boldsymbol{\beta}}(\boldsymbol{\xi}) = S(\boldsymbol{\xi})\Psi_{\boldsymbol{\beta}}(\boldsymbol{\xi})$ and $\boldsymbol{\beta} \in \mathcal{B}_{PC}$.

- A minimal compatibility relation between $\mathcal{A}$ and $\mathcal{B}$ is that they satisfy the **discrete orthogonality** relation for the PC, that is

$$I_N^{\mathcal{A}}(\Psi_{\boldsymbol{\beta}}\Psi'_{\boldsymbol{\beta}}) = \delta_{\boldsymbol{\beta}\boldsymbol{\beta}'}\left\langle \Psi_{\boldsymbol{\beta}}, \Psi_{\boldsymbol{\beta}}\right\rangle, \quad \forall \boldsymbol{\beta}, \boldsymbol{\beta}' \in \mathcal{B}.$$

- Given $\mathcal{B}$ one can determine the **minimal admissible set** $\mathcal{A}$ from the exactness properties of the one-dimensional quadratures.
- Alternatively, given $\mathcal{A}$, one can determine $\mathcal{B}$ as **the largest admissible set of PC tensorizations** that satisfies the discrete orthogonality.
- **Recall:** the model-output S is usually not polynomial. Be conservative!
- What about the case of multiple model-outputs ?

Preconditioning

Non-Intrusive projections are appealing for complex & non-linear problems

BUT non-intrusive **does not mean that $U(\xi)$ is easily approximated**.

Difficulties remains for

- non-smooth mapping $\xi \mapsto U(\xi)$
- but also for enforcement of **positivity constraints, presence of plateau, saturation behavior, highly stretched dependences**.

Preconditioning can help in these situations, introducing an **inversible** transformation Φ:

$$Y(\xi) = \Phi(U(\xi)) \rightarrow U(\xi) \approx \Phi^{-1}(\sum_{\alpha \in \mathcal{A}} y_\alpha \Phi_\alpha(\xi)).$$

Transformation must be chosen so $Y(\xi)$ has a tight spectrum, allowing the use of a low degree PC expansion.

Preconditioning

Preconditioning can help in these situations, introducing an **inversible** transformation Φ:

$$Y(\xi) = \Phi(U(\xi)) \rightarrow U(\xi) \approx \Phi^{-1}(\sum_{\alpha \in \mathcal{A}} y_\alpha \Phi_\alpha(\xi)).$$

Transformation must be chosen so $Y(\xi)$ has a tight spectrum, allowing the use of a low degree PC expansion.

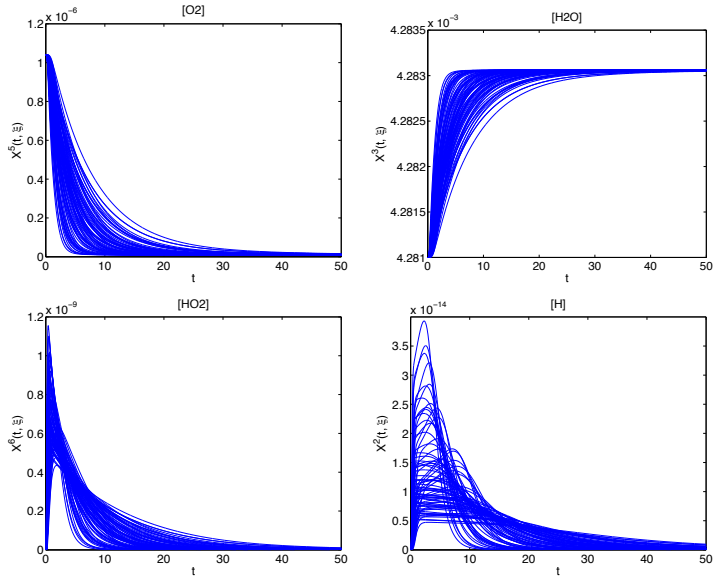


Figure 1: Concentration versus time selected realizations in $\mathcal{S}$. Shown are curves for O_2, H_2O the original realizations shown in Figure 1. HO_2, and H, as indicated.

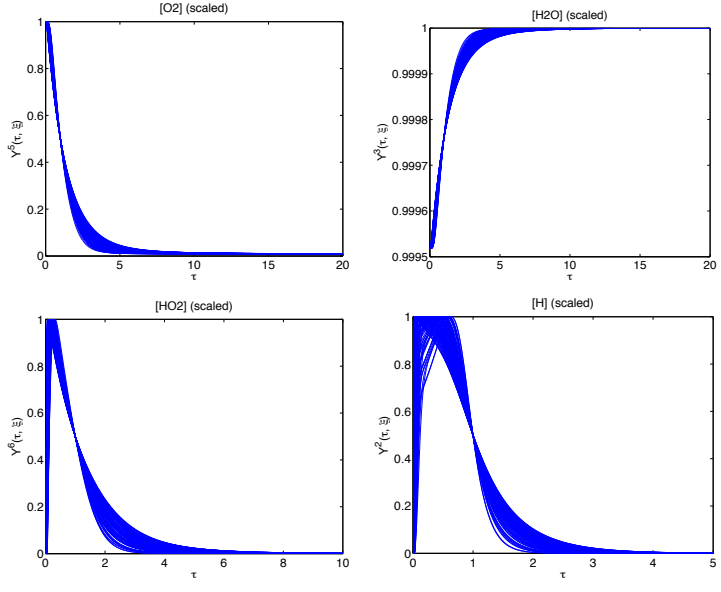


Figure 2: Scaled variables versus stretched time. Individual curves are obtained by transforming

Preconditioning

Preconditioning can help in these situations, introducing an **inversible** transformation Φ:

$$Y(\xi) = \Phi(U(\xi)) \rightarrow U(\xi) \approx \Phi^{-1}(\sum_{\alpha \in \mathcal{A}} y_\alpha \Phi_\alpha(\xi)).$$

Transformation must be chosen so $Y(\xi)$ has a tight spectrum, allowing the use of a low degree PC expansion.

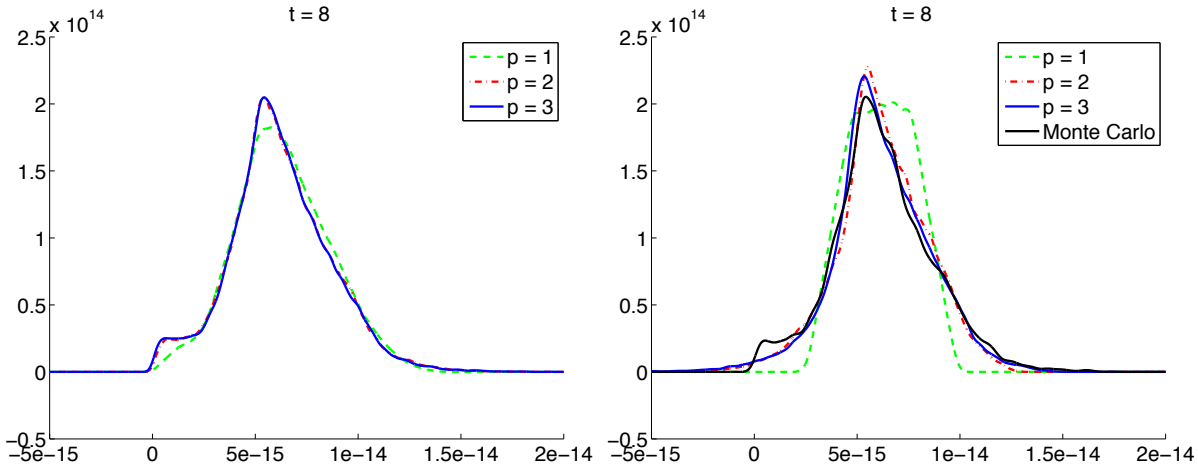


Figure 14: PDFs of $[H]$ at time $t = 8$. Left: preconditioned NISP at different PC orders as indicated. Right: direct NISP method with the same orders. Also shown on the right is the PDF of $[H]$ generated with Monte-Carlo sampling.

Questions?

Further readings:

- S. Smolyak, Quadrature and interpolation formulas for tensor products of certain class of functions. *Dokl. Akad. Nauk. SSSR*, **4**, 240-243, (1963).
- E. Novak and K. Ritter, High dimensional integration of smooth functions over cubes. *Numer. Math.*, **75**, pp. 79-97, (1996).
- T. Gerstner and M. Griebel, Numerical integration using sparse grids. *Numerical Algorithms*, **18**, 209-232, (1998).
- K. Petras, On the Smolyak cubature error for analytic functions. *Adv. Comput. Math.*, **12**, 71-93, (2000).
- K. Petras, Fast calculation in the Smolyak algorithm. *Numerical Algorithms*, **26**, 93-109, (2001).
- F. Nobile, R. Tempone and C. Webster, An anisotropic sparse grid stochastic collocation method for partial differential equations with random input data. *SIAM J. Numerical Analysis*, **46**:5, 2411-2442, (2008).
- A. Keese and H. Matthies, Numerical methods and Smolyak quadrature for nonlinear stochastic partial differential equations. *Technical Report, Institute of Scientific Computing TU Braunschweig*, (2003).
- X. Ma and N. Zabaras, Adaptative hierarchical sparse grid collocation algorithm for the solution of stochastic differential equations. *J. Computational Physics*, (2009).