

Quadrature de Smolyak

Michaël Baudin

13 juin 2022

Table des matières

Introduction	Sparse multidimensional quadrature
Références bibliographiques	Multi-indices
Bibliographie des implémentations	Tables
Quadrature unidimensionnelle	Combinaison et superposition des noeuds
Tensorisation	Error of sparse quadrature
Exemples de quadrature par tensorisation	Exemple
Difference 1D quadrature	Nombre de noeuds
Tensor product of difference univariate quadratures	Convergence
	Références

Introduction

- ▶ La méthode de quadrature de Smolyak [[Smolyak, 1963](#)] utilise des grilles creuses tensorisées.
- ▶ Elle se fonde sur des règles de quadrature unidimensionnelles imbriquées telles la seconde règle de Fejér et la règle de Clenshaw-Curtis ou non imbriquées comme la quadrature de Gauss.
- ▶ La règle de Smolyak permet de créer une règle de quadrature multidimensionnelle associée à un nombre de noeuds très réduit comparé à une règle de quadrature multidimensionnelle par tensorisation.
- ▶ Sa précision est souvent comparable en pratique à celle de la quadrature de Gauss tensorisée.

Quelques références bibliographiques commentées

- ▶ Le point de départ est l'article de Smolyak [[Smolyak, 1963](#)], en russe.
- ▶ L'article de Gerstner et Griebel [[Gerstner and Griebel, 1998](#)] est un point d'entrée intéressant sur la méthode. L'article est une synthèse de travaux antérieurs, en particulier [[Griebel et al., 1992](#)] qui introduit la méthode par combinaison.
- ▶ Les articles de Novak et Ritter [[Novak and Ritter, 1996](#)] et [[Novak and Ritter, 1999](#)] présentent des bornes d'erreur pour la quadrature de Smolyak, le degré d'exactitude polynomial et des exemples. Ils s'appuient en particulier sur le travail théorique de [[Wasilkowski and Wozniakowski, 1995](#)].
- ▶ Les articles de Knut Petras [[Petras, 2001](#)] et [[Petras, 2003](#)] présentent comment réduire le nombre de noeuds de la quadrature, en utilisant en particulier des formules de quadrature univariées imbriquées. L'auteur présente en particulier la séquence de Kronrod-Patterson avec retard. Plusieurs exemples d'applications sur des cas de différentes dimensions sont présentés.
- ▶ L'article de Garcke [[Garcke, 2006](#)] présente les grilles creuses et les bases de fonction hiérarchiques que l'on peut utiliser pour l'interpolation. L'annexe présente un script Python implémentant la méthode en base 2.

Quelques références bibliographiques commentées

- ▶ L'utilisation de la quadrature de Smolyak pour l'estimation des coefficients des polynômes du chaos est présentée dans [Le Maître and Knio, 2010] page 56.
- ▶ Dans [Sullivan, 2015] page 176 on peut lire une présentation synthétique pour le calcul d'intégrales.
- ▶ Les diapos [Winokur and Red-Horse, 2016] de Winokur et Red-Horse sont une synthèse intéressante et introduisent le thème plus général des grilles creuses.
- ▶ Dans [Tempone and Wolfers, 2018], la section 3.1 présente le problème du calcul de l'ensemble optimal de multi-indices permettant de tenter de minimiser l'erreur sous la contrainte d'un travail inférieur à un seuil. Ce problème est similaire au problème du sac à dos.

Implémentations de la quadrature de Smolyak

Analysons différentes implémentations de la quadrature de Smolyak dans plusieurs bibliothèques.

1. Smolpack, Knut Petras [[Petras, 2003](#)], Langage : C. Environ 500 lignes d'algorithme et 1000 lignes de paramètres numériques. Dimension inférieure ou égale à 40. Quadrature de Clenshaw-Curtis et quadrature de Kronrod-Patterson delayed (selon l'article). [Smolpack chez Burkardt](#) [Dépôt github](#) Le code de K.Petras permet d'intégrer une fonction, mais ne présente pas d'interface pour calculer les noeuds et les poids. Les codes dans [smolyak.cpp](#) et [smolyak.h](#) ont été modifiés par Chong Gu pour les besoins du paquet R [gss](#) pour récupérer les noeuds et les poids, sans nécessairement intégrer une fonction. Une analyse de la bibliothèque Smolpack est disponible.
2. smolyak, Soeren Wolfers [[Tempone and Wolfers, 2018](#)] : [Sur github](#). Paquet pip. [R](#). Tempone, S. Wolfers (2017) "Smolyak's algorithm: a powerful black box" Limitation : pas de quadrature ? Question posée à l'auteur. Réponse : non.

Liste des librairies

3. chaospy, Jonathan Feinberg. Fonction `generate_quadrature()` avec option `sparse = True`. Option `growth` pour l'imbrication. [Page d'aide](#). [Feinberg and Tennøe., 2015a, Feinberg and Tennøe., 2015b]
4. Jochen Garcke [Garcke, 2006]. Classe `SparseGrid`. Langage : Python. Génère des points en base 2. Ne génère pas les points de la quadrature. Une analyse de `SparseGrid` par Garcke est disponible. Le code est disponible dans l'article. Le [code est disponible sur le site de l'auteur](#) mais j'ai créé une [variante améliorée avec des démos](#). Ne génère pas de quadrature (pas de poids).
5. EconForge¹. Langage Python (et Julia). Sur [github](#). Génère les points pour l'interpolation, mais pas les poids de la quadrature.

1. Judd, K. L., Maliar, L., Maliar, S., & Valero, R. (2014). Smolyak method for solving dynamic economic models : Lagrange interpolation, anisotropic grid and adaptive domain. *Journal of Economic Dynamics and Control*, 44, 92-123.

Liste des librairies

6. Ronaldo Carpio, [py_tsg](#). `py_tsg` is a Python wrapper around the Tasmanian Sparse Grid Library (TSG) ([lien](#)) for high-dimensional interpolation and integration using sparse grids (e.g. Smolyak grids). TSG is written in C++ and supports a variety of polynomial basis functions, anisotropic weights, and adaptive local refinement. Calcul des noeuds et des poids : OK pour quadrature.
7. Richard Dwight and Andrea Resmini [Lien](#). Sparse grid and Sobol index code (Python).
8. Morgan Fouesneau, [sparsegrid](#) Smolyak Sparse Grid Interpolation. Langage : Python. Crée la grille creuse et évalue une interpolation de Chebyshev. Pas de quadrature.
9. [SG++](#) Dirk Pflüger² Code source sur [github](#). Langage : C++. Pas de quadrature ? TODO : interroger le développeur.
10. [UQLab](#). Pas d'accès au code source.

2. D. Pflüger, Spatially Adaptive Sparse Grids for High-Dimensional Problems, Verlag Dr. Hut, 2010.

Synthèse

Code	Sparse Grid	Quadrature
Smolpack, K.Petras	Non	✓
py_tsg, R. Carpio	✓	✓
Chaospy, J. Feinberg	?	✓
(UQLab)	Non	✓
SG++, D. Pflüger	✓	?
Smolyak, S. Wolfers	✓	?
Sparse, R. Dwight	?	?
Jochen Garcke	✓	Non
EconForge, Judd	✓	Non
Sparsegrid, M. Fouesneau	✓	Non

Quadrature unidimensionnelle

Soit $\mathcal{X} \subset \mathbb{R}$ un domaine d'intégration *unidimensionnel* et soit $f : \mathcal{X} \rightarrow \mathbb{R}$ une fonction intégrable. Dans ce cas, on s'intéresse à l'intégrale :

$$\int_{\mathcal{X}} f(x) dx.$$

On considère ici les notations des méthodes numériques dans lesquelles f est l'intégrande, et non pas la densité de probabilité. Il n'y a pas de densité de probabilité dans ce document.

Quadrature unidimensionnelle

Definition 1

(*Univariate quadrature rule*) For any $\ell \geq 0$, let Q_ℓ be a one-dimensional quadrature formula with $n_\ell^{(1)}$ nodes which integrates the function $f : \mathcal{X} \subset \mathbb{R}$. The integral is :

$$\int_{\mathcal{X}} f(x) dx \approx Q_\ell(f)$$

where ([[Gerstner and Griebel, 1998](#)] page 214)

$$Q_\ell(f) = \sum_{k=1}^{n_\ell^{(1)}} w_{\ell,k} f(x_{\ell,k})$$

where $x_{\ell,1}, \dots, x_{\ell,n_\ell}$ are the nodes and $w_{\ell,1}, \dots, w_{\ell,n_\ell}$ are the weights.

Dans la définition précédente, l'exposant (1) indique que la quadrature est unidimensionnelle, l'entier ℓ est le niveau de la quadrature, $n_\ell^{(1)}$ est le nombre de noeud la quadrature, $w_{\ell,1}, \dots, w_{\ell,n_\ell^{(1)}}$ sont les poids et $x_{\ell,1}, \dots, x_{\ell,n_\ell^{(1)}}$ sont les noeuds.

Tensorisation

Soit $\mathcal{X} \subset \mathbb{R}^p$ un domaine d'intégration *multidimensionnel* et soit $f : \mathcal{X} \rightarrow \mathbb{R}$ une fonction intégrable. Dans ce cas, on s'intéresse à l'intégrale :

$$\int_{\mathcal{X}} f(\mathbf{x}) d\mathbf{x}.$$

Let $\mathbf{k} \in (\mathbb{N}^*)^p$ be the multi-index $\mathbf{k} = (k_1, \dots, k_p)$ where

$$\mathbb{N}^* = \{1, 2, \dots, \}$$

is the set of natural numbers without zero. Consider the 1 and the infinity norms ([[Le Maître and Knio, 2010](#)] page 57, eq. 3.28) :

$$\|\mathbf{k}\|_1 = \sum_{i=1}^p k_i, \quad \|\mathbf{k}\|_\infty = \max_{i=1, \dots, p} k_i,$$

for any $\mathbf{k} \in (\mathbb{N}^*)^p$.

Definition 2

(*Tensor product quadrature*) For any given $\mathbf{k} = (k_1, \dots, k_p)$, the tensor product quadrature is ([Gerstner and Griebel, 1998] page 214) :

$$Q_{\ell}^{(p)} = Q_{\ell_1}^{(1)} \otimes \dots \otimes Q_{\ell_p}^{(1)}$$

where

$$Q_{\ell_1}^{(1)} \otimes \dots \otimes Q_{\ell_p}^{(1)}(f) = \sum_{k_1=1}^{n_{\ell_1}^{(1)}} \dots \sum_{k_p=1}^{n_{\ell_p}^{(1)}} w_{\ell_1, k_1} \dots w_{\ell_p, k_p} f(x_{\ell_1, k_1}, \dots, x_{\ell_p, k_p}).$$

Dans la définition précédente, pour $i = 1, \dots, p$, l'exposant (1) indique que la quadrature est unidimensionnelle, l'entier ℓ_i est le niveau de la i-ème quadrature, $n_{\ell_i}^{(1)}$ est le nombre de noeud la i-ème quadrature, $w_{\ell_i, 1}, \dots, w_{\ell_i, n_{\ell_i}^{(1)}}$ sont les poids et

$x_{\ell_i, 1}, \dots, x_{\ell_i, n_{\ell_i}^{(1)}}$ sont les noeuds.

En d'autres termes on considère toutes les *combinaisons* des noeuds et les *produits* correspondants des poids.

En utilisant les multi-indices pour la quadrature, on obtient :

$$Q_{\ell_1}^{(1)} \otimes \dots \otimes Q_{\ell_p}^{(1)}(f) = \sum_{\mathbf{k} \leq \left(n_{\ell_1}^{(1)}, \dots, n_{\ell_p}^{(1)}\right)} w_{\mathbf{k}} f(\mathbf{x}_{\mathbf{k}}).$$

où les poids sont :

$$w_{\mathbf{k}} = \prod_{i=1}^p w_{\ell_i, k_i}$$

et les noeuds sont :

$$\mathbf{x}_{\mathbf{k}} = (x_{\ell_1, k_1}, \dots, x_{\ell_p, k_p}).$$

On peut utiliser une règle d'énumération de tous les multi-indices satisfaisant l'équation :

$$\mathbf{k} \leq \left(n_{\ell_1}^{(1)}, \dots, n_{\ell_p}^{(1)}\right).$$

Chaque poids de la règle multidimensionnelle est égal au produit des poids des règles unidimensionnelles et chaque noeud de la règle multidimensionnelle a pour composantes les noeuds des règles unidimensionnelles. Cette règle est anisotrope. Elle ne correspond ni à la norme 1, ni à la norme infinie du multiindice \mathbf{k} .

Exemple de quadrature par tensorisation

Exemple 3

(*Intégration d'une fonction*). On considère l'intégrale [Morokoff and Caflisch, 1995] :

$$\int_{[0,1]^p} (1 + 1/p)^p \prod_{i=1}^p x_i^{(1/p)} d\mathbf{x} = 1.$$

Ce problème-test est implémenté dans ³ et dans ⁴ j'ai testé différentes méthodes quadrature pour calculer l'intégrale.

-
3. [QuadratureProblem.py](#)
 4. [tensor_product_quadrature.py](#)

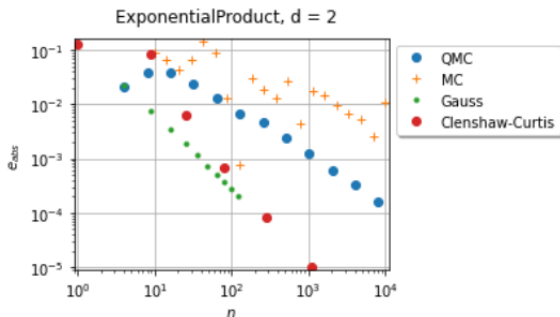


Figure 1 – Erreur de différentes méthodes de quadrature pour le problème `ExponentialProduct` en dimension 2.

On observe que la meilleure vitesse est atteinte par la quadrature de Gauss-Legendre, suivie de près par la quadrature tensorisée de Clenshaw-Curtis. La méthode QMC est intermédiaire. Les résultats les moins précis sont obtenus par Monte-Carlo simple.

Exemple 4

(*Règles isotropes*). Dans⁵, j’ai montré comment créer une règle de quadrature par tensorisation de règles unidimensionnelles. J’ai utilisé la classe `GaussProductExperiment` d’OpenTURNS avec 20 points dans chaque direction, ce qui crée une règle isotrope.

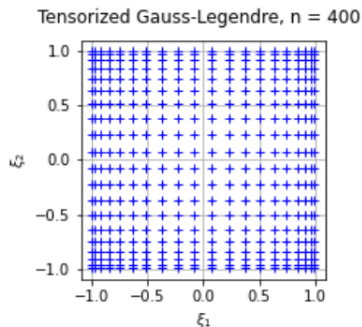


Figure 2 – Quadrature de Gauss en dimension 2 par tensorisation de la règle de Gauss-Legendre dans les deux dimensions.

5. [tensorization_gauss.py](#)

Difference 1D quadrature

For any $\ell \geq 0$, let Q_ℓ be a one-dimensional quadrature formula with $n_\ell^{(1)}$ nodes which integrates the function $f : \mathcal{X} \subset \mathbb{R}$. The integral is :

$$\int_{\mathcal{X}} f(x) dx \approx Q_\ell(h)$$

where

$$Q_\ell(f) = \sum_{j=1}^{n_\ell^{(1)}} w_\ell^{(j)} h \left(\xi_\ell^{(j)} \right)$$

We can express the one dimensional quadrature as the difference :

$$Q_\ell = \sum_{k=1}^{\ell} (Q_k - Q_{k-1})$$

where $Q_0 = \emptyset$ is the empty quadrature. This leads to the following definition of a one dimensional difference quadrature.

1D quadrature

Definition (*Difference quadrature*) Let Q_k be a univariate quadrature of level k , for any $k \geq 1$. Let Δ_k be the difference quadrature defined by :

$$\Delta_k = Q_k - Q_{k-1},$$

for any $k \geq 1$.

Therefore, the quadrature formula Q_ℓ can be expressed depending on difference quadratures :

$$Q_\ell = \sum_{k=1}^{\ell} \Delta_k.$$

for any $\ell \geq 1$.

1D quadrature

Example. (*Difference quadrature of levels 1 and 2*) If $k = 1$, we get

$$\Delta_1 = Q_1 - Q_0 = Q_1$$

since $Q_0 = \emptyset$. If $k = 2$, we get

$$\Delta_2 = Q_2 - Q_1.$$

Example. (*Difference quadrature of level 3*) If $\ell = 3$, therefore :

$$\begin{aligned} Q_3 &= \sum_{k=1}^3 \Delta_k \\ &= \Delta_1 + \Delta_2 + \Delta_3 \\ &= (Q_3 - Q_2) + (Q_2 - Q_1) + Q_1 \\ &= Q_3. \end{aligned}$$

The calculation involves Q_1 , Q_2 and Q_3 , but uses difference quadrature which may prevent extra computation if the quadrature is nested.

Tensor product quadrature of difference univariate quadratures

Let $\mathbf{k} \in \mathbb{N}_1^p$ be the multi-index $\mathbf{k} = (k_1, \dots, k_p)$ where

$$\mathbb{N}_1 = \{1, 2, \dots\}$$

is the set of natural numbers without zero. Consider the 1 and the infinity norms ([Le Maître and Knio, 2010] page 57, eq. 3.28) :

$$\|\mathbf{k}\|_1 = \sum_{i=1}^p k_i \quad \|\mathbf{k}\|_\infty = \max_{i=1} k_i.$$

for any $\mathbf{k} \in \mathbb{N}_1^p$.

Definition (*Tensor product quadrature*) The tensor product quadrature is :

$$Q_\ell^{(p)} = Q_\ell^{(1)} \otimes \dots \otimes Q_\ell^{(1)}.$$

The following theorem provides an equivalent equation for the tensor product quadrature ([Le Maître and Knio, 2010] page 57, eq. 3.30).

Theorem (*Tensor product quadrature*) The tensor product quadrature is :

$$Q_\ell^{(p)} = \sum_{\|\mathbf{k}\|_\infty \leq \ell} \Delta_{k_1}^{(1)} \otimes \dots \otimes \Delta_{k_p}^{(1)}.$$

The significant part of the previous equation is the set of multi-indices $\|\mathbf{k}\|_\infty \leq \ell$, which may be very large depending on the dimension of the problem.

The following proof is given in [Winokur and Red-Horse, 2016] page 14. La preuve s'appuie sur les propriétés de la quadrature tensorisée, en particulier la distributivité de la tensorisation.

Proof. We notice that, for any $i = 1, \dots, p$, we have :

$$Q_\ell^{(1)} = \sum_{k_i=1}^{\ell} \Delta_{k_i}^{(1)}.$$

Hence,

$$Q_\ell^{(p)} = \left(\sum_{k_1=1}^{\ell} \Delta_{k_1}^{(1)} \right) \otimes \dots \otimes \left(\sum_{k_p=1}^{\ell} \Delta_{k_p}^{(1)} \right).$$

The quadrature operator is linear which implies :

$$Q_\ell^{(p)} = \sum_{k_1=1}^{\ell} \dots \sum_{k_p=1}^{\ell} \Delta_{k_1}^{(1)} \otimes \dots \otimes \Delta_{k_p}^{(1)}.$$

The set of multi-indices which correspond to the previous sum is equivalent to $\|\mathbf{k}\|_\infty \leq \ell$, which concludes the proof. □

Example. (*Tensor product quadrature in 2 dimensions*) Consider the dimension $p = 2$. Therefore,

$$Q_\ell^{(2)} = \sum_{\|\mathbf{k}\|_\infty \leq \ell} \Delta_{k_1}^{(1)} \otimes \Delta_{k_2}^{(1)}$$

If $\ell = 1$, therefore,

$$\begin{aligned} Q_1^{(2)} &= \sum_{\|\mathbf{k}\|_\infty \leq 1} \Delta_{k_1}^{(1)} \otimes \Delta_{k_2}^{(1)} \\ &= \Delta_1^{(1)} \otimes \Delta_1^{(1)} \\ &= Q_1^{(1)} \otimes Q_1^{(1)} \end{aligned}$$

since $\Delta_1^{(1)} = Q_1^{(1)}$. The right hand side of the previous equation is, indeed, the tensor product of dimension 1 quadratures.

If $\ell = 2$, therefore,

$$\begin{aligned}
 Q_1^{(2)} &= \sum_{\|k\|_\infty \leq 2} \Delta_{k_1}^{(1)} \otimes \Delta_{k_2}^{(1)} \\
 &= \Delta_1^{(1)} \otimes \Delta_1^{(1)} + \Delta_1^{(1)} \otimes \Delta_2^{(1)} \\
 &\quad + \Delta_2^{(1)} \otimes \Delta_1^{(1)} + \Delta_2^{(1)} \otimes \Delta_2^{(1)} \\
 &= Q_1^{(1)} \otimes Q_1^{(1)} + Q_1^{(1)} \otimes (Q_2^{(1)} - Q_1^{(1)}) \\
 &\quad + (Q_2^{(1)} - Q_1^{(1)}) \otimes Q_1^{(1)} \\
 &\quad + (Q_2^{(1)} - Q_1^{(1)}) \otimes (Q_2^{(1)} - Q_1^{(1)}).
 \end{aligned}$$

We expand the previous products and get :

$$\begin{aligned}
 Q_1^{(2)} &= Q_1^{(1)} \otimes Q_1^{(1)} + Q_1^{(1)} \otimes Q_2^{(1)} - Q_1^{(1)} \otimes Q_1^{(1)} \\
 &\quad + Q_2^{(1)} \otimes Q_1^{(1)} - Q_1^{(1)} \otimes Q_1^{(1)} \\
 &\quad + Q_2^{(1)} \otimes Q_2^{(1)} - Q_2^{(1)} \otimes Q_1^{(1)} \\
 &\quad - Q_1^{(1)} \otimes Q_2^{(1)} + Q_1^{(1)} \otimes Q_1^{(1)} \\
 &= Q_2^{(1)} \otimes Q_2^{(1)},
 \end{aligned}$$

which is the tensor product quadrature in 2 dimensions.

Sparse multidimensional quadrature

One of the ways to reduce the size of this set is to consider the smaller set of mult-indices such that $\|\mathbf{k}\|_1 \leq \ell + p - 1$. The sparse quadrature ([Le Maître and Knio, 2010] page 57, eq. 3.29, [Gerstner and Griebel, 1998] page 214) is introduced in the following definition.

Definition (*Sparse Smolyak quadrature by differences*) The sparse quadrature formula at level ℓ is :

$$Q_{\ell}^{(p)} = \sum_{\|\mathbf{k}\|_1 \leq \ell + p - 1} \Delta_{k_1}^{(1)} \otimes \cdots \otimes \Delta_{k_p}^{(1)}$$

for any $\ell \geq 1$.

Sparse multidimensional quadrature

Example (*Infinite and one norms*) Consider for example the dimension $p = 2$. Hence, for any $\mathbf{k} \in (\mathbb{N}_1)^2$, we are interested by the set corresponding to the tensor product :

$$\|\mathbf{k}\|_\infty = \max(k_1, k_2) \leq \ell$$

and the sparse quadrature :

$$\|\mathbf{k}\|_1 = k_1 + k_2 \leq \ell + p - 1 = \ell + 1.$$

Consider for example $\ell = 5$. The following figure presents the multi-indices in dimension $p = 2$ with level $\ell = 5$ corresponding to the tensor product quadrature. We count $5 \times 5 = 25$ multi-indices.

Sparse multidimensional quadrature

La figure suivante⁶ montre le résultat.

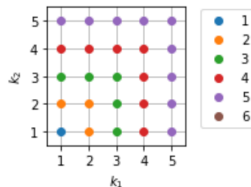


Figure 3 – Multi-indices in dimension $p = 2$ with level $\ell = 5$ corresponding to the tensor product quadrature.

6. [scripts/sparse_indices.py](#)

Sparse multidimensional quadrature

The following figure presents the multi-indices in dimension $p = 2$ with level $\ell = 5$ corresponding to the sparse quadrature. We count 15 multi-indices.

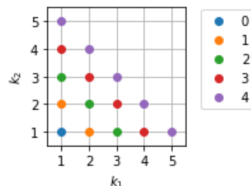


Figure 4 – Multi-indices in dimension $p = 2$ with level $\ell = 5$ corresponding to the sparse quadrature.

Sparse multidimensional quadrature

The following theorem rewrites the previous sparse cubature in terms of lower dimensionality quadratures ([Sullivan, 2015] page 177, eq. 9.9). This is the *recursive* expression of Smolyak's quadrature ([Gerstner and Griebel, 1998] page 215).

Theorem (*Smolyak quadrature by recursion*) The sparse quadrature formula at level ℓ is :

$$Q_{\ell}^{(p)} = \sum_{i_{\ell}=1}^{\ell} \Delta_{i_{\ell}}^{(1)} \otimes Q_{\ell-i_{\ell}+1}^{(p-1)}$$

for any $\ell \geq 1$.

In the previous equation, the sparse quadrature $Q_{\ell}^{(p)}$ is written depending on the dimension one difference quadrature $\Delta_{i_{\ell}}^{(1)}$ and the $(p-1)$ quadrature $Q_{\ell-i_{\ell}+1}^{(p-1)}$.

We can expand the sum and get :

$$Q_{\ell}^{(p)} = \Delta_1^{(1)} \otimes Q_{\ell}^{(p-1)} + \Delta_2^{(1)} \otimes Q_{\ell-1}^{(p-1)} + \dots + \Delta_{\ell}^{(1)} \otimes Q_1^{(p-1)}$$

for any $\ell \geq 1$.

Smolyak quadrature involve weights which are negative ([Sullivan, 2015] page 177).

Sparse multidimensional quadrature

Example. Consider the dimension $p = 2$:

$$Q_\ell^{(2)} = \sum_{i_\ell=1}^{\ell} \Delta_{i_\ell}^{(1)} \otimes Q_{\ell-i_\ell+1}^{(1)}$$

for any $\ell \geq 1$. If $\ell = 1$, then

$$Q_1^{(2)} = \Delta_1^{(1)} \otimes Q_1^{(1)}.$$

But $\Delta_1^{(1)} = Q_1^{(1)} - Q_0^{(1)} = Q_1^{(1)}$. Therefore,

$$Q_1^{(2)} = Q_1^{(1)} \otimes Q_1^{(1)}$$

which is the tensor product rule at level $\ell = 1$. If $\ell = 2$, then

$$\begin{aligned} Q_\ell^{(2)} &= \Delta_1^{(1)} \otimes Q_2^{(1)} + \Delta_2^{(1)} \otimes Q_1^{(1)} \\ &= \left(Q_1^{(1)} - Q_0^{(1)} \right) \otimes Q_2^{(1)} + \left(Q_2^{(1)} - Q_1^{(1)} \right) \otimes Q_1^{(1)} \\ &= Q_1^{(1)} \otimes Q_2^{(1)} + Q_2^{(1)} \otimes Q_1^{(1)} - Q_1^{(1)} \otimes Q_1^{(1)} \end{aligned}$$

Sparse multidimensional quadrature

Another expression of Smolyak's formula is given in [Gerstner and Griebel, 1998] page 215 and is presented in the following theorem. It expresses the multivariate quadrature in terms of combinations univariate quadratures, known as the *combination technique*.

Theorem (*Smolyak quadrature by combination technique*) The sparse quadrature formula at level ℓ is :

$$Q_{\ell}^{(p)} = \sum_{\ell \leq \|\mathbf{k}\|_1 \leq \ell+p-1} (-1)^{\ell+p-\|\mathbf{k}\|_1-1} \binom{p-1}{\|\mathbf{k}\|_1-\ell} Q_{k_1}^{(1)} \otimes \cdots \otimes Q_{k_p}^{(1)}$$

for any $\ell \geq 1$ where the binomial coefficient is :

$$\binom{n}{m} = \frac{n!}{m! (n-m)!}$$

for any integers $n \geq 0$ and $0 \leq m \leq n$.

Proof TODO

Multi-indices

Exemple (*Multi-indices pour la technique par combinaison*) La figure 3 montre l'ensemble des multi-indices en dimension $p = 2$ pour le niveau $\ell = 3$. Puisque $\ell + p - 1 = 4$, les multi-indices sont tels que $3 \leq \|\mathbf{k}\|_1 \leq 4$ et $k_i \geq 1$ pour $i = 1, 2$.

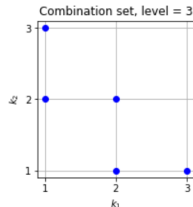


Figure 5 – Multi-indices en dimension $p = 2$ pour le niveau $\ell = 3$.

Multi-indices

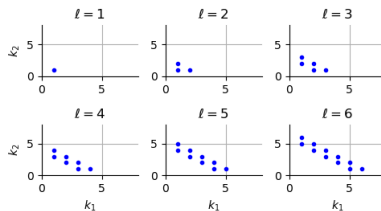


Figure 6 – Multi-indices en dimension $p = 2$ pour les niveaux $\ell = 1$ à 6.

Multi-indices

Les multi-indices en dimension $p = 2$ pour le niveau $\ell = 3$ sont présentés dans la table suivante.

Multi-indices	$\ \mathbf{k}\ _1$
(2,1)	3
(1,2)	3
(3,1)	4
(2,2)	4
(1,3)	4

Table 1 : Multi-indices en dimension $p = 2$ pour le niveau $\ell = 3$.

Table de noeuds de la quadrature de Smolyak

Les tables suivantes, produites avec **chaospy**, présentent la quadrature de Smolyak. Les tables 1 à 3 sont fondées sur une quadrature marginale de Gauss-Legendre. On l'obtient avec l'instruction **rule=["gaussian", "gaussian"]**.

Indice	Noeuds	Poids
0	[-0.5774, 0.0000]	0.5000
1	[0.0000, -0.5774]	0.5000
2	[0.0000, 0.0000]	-1.0000
3	[0.0000, 0.5774]	0.5000
4	[0.5774, 0.0000]	0.5000

Table 1 – Noeuds et poids de la quadrature de Smolyak de niveau $\ell = 1$ fondée sur une quadrature de Gauss-Legendre avec **chaospy** en dimension 2.

Indice	Noeuds	Poids
0	$[-0.7746, 0.0000]$	0.2778
1	$[-0.5774, -0.5774]$	0.2500
2	$[-0.5774, 0.0000]$	-0.5000
3	$[-0.5774, 0.5774]$	0.2500
4	$[-0.0000, 0.0000]$	0.4444
5	$[0.0000, -0.7746]$	0.2778
6	$[0.0000, -0.5774]$	-0.5000
7	$[0.0000, -0.0000]$	0.4444
8	$[0.0000, 0.5774]$	-0.5000
9	$[0.0000, 0.7746]$	0.2778
10	$[0.5774, -0.5774]$	0.2500
11	$[0.5774, 0.0000]$	-0.5000
12	$[0.5774, 0.5774]$	0.2500
13	$[0.7746, 0.0000]$	0.2778

Table 2 – Noeuds et poids de la quadrature de Smolyak de niveau $\ell = 2$ fondée sur une quadrature de Gauss-Legendre avec **chaospy** en dimension 2.

Combinaison et superposition des noeuds

La figure 7 montre comment les règles de quadrature unidimensionnelles sont combinées pour former la règle de quadrature en dimension $p = 2$ par la technique de combinaison de Smolyak. On constate que, pour la règle de quadrature notée $\Pi_{i,j}$, le nombre de noeuds dans chaque direction est égal à $(1 + i, 1 + j)$. Par exemple, la règle de quadrature $\Pi_{1,2}$ possède 2 noeuds dans la direction x_1 et 3 noeuds dans la direction x_2 .

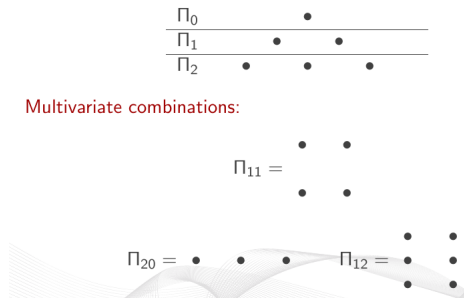


Figure 7 – En haut, trois règles de quadrature unidimensionnelles. En bas, les règles de quadrature en dimension $p = 2$ créées par la quadrature de Smolyak ([Feinberg and Tennøe., 2015b] page 8).

Combinaison et superposition des noeuds

La figure 8 montre comment les noeuds sont combinés par la quadrature de Smolyak. Certains noeuds sont ajoutés à un niveau élevé (en bleu), mais peuvent être retranchés par des quadratures de niveau inférieurs (en rouge).

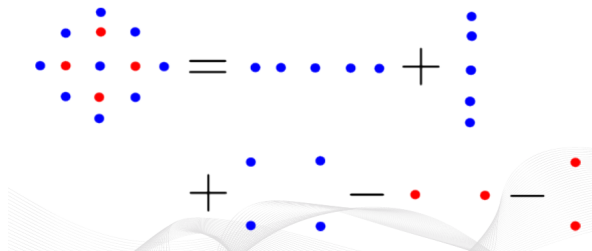


Figure 8 – Combinaison des noeuds de les règles de quadrature en dimension $p = 2$ créés par la quadrature de Smolyak ([Feinberg and Tennøe., 2015b] page 13).

Combinaison et superposition des noeuds

Pour limiter le nombre de noeuds dans la quadrature de Smolyak, on peut utiliser des quadratures imbriquées comme par exemple la quadrature de Clenshaw-Curtis. On peut aussi utiliser une autre règle, comme par exemple la règle de Fejér de type 2 ou la règle du trapèze composite.

Clenshaw-Curtis:



Nested Clenshaw-Curtis:



Figure 9 – Quadrature de Clenshaw-Curtis et de Clenshaw-Curtis imbriquée ([Feinberg and Tennøe., 2015b] page 18).

Combinaison et superposition des noeuds

Lorsqu'on combine les noeuds par la quadrature de Smolyak, certains noeuds sont superposés, comme le montre la figure 10. Le calcul des noeuds uniques et des poids correspondants peut être calculé par la méthode présentée dans l'algorithme pour la quadrature de Smolyak.

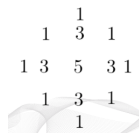


Figure 10 – Nombre de noeuds superposés par une quadrature de Clenshaw-Curtis imbriquée en dimension $p = 2$ créées par la quadrature de Smolyak ([Feinberg and Tennøe., 2015b] page 19).

Error of sparse quadrature

The following theorem presents the absolute error of a sparse quadrature ([Sullivan, 2015] page 177, eq. 9.10).

Theorem (*Error of sparse quadrature*) Assume that $h \in \mathcal{C}^r([0, 1]^p)$ and that we use a sparse quadrature with n_ℓ nodes at level ℓ . Therefore,

$$\left| \int_{[0,1]^p} h(\boldsymbol{\xi}) d\boldsymbol{\xi} - Q_\ell(h) \right| = O \left(n_\ell^{-r} (\log(n_\ell))^{(p-1)(r+1)} \right).$$

Exemple

Dans `pycrash/smolyak_chaospy_uniform.py`, nous avons utilisé la librairie `chaospy` pour dessiner les noeuds et poids de la quadrature de Smolyak. La figure 7 présente la quadrature de Smolyak-Legendre. Les noeuds associés à un poids positif sont en bleu tandis que les noeuds associés à un poids négatif sont en gris. De plus, les noeuds associés à des poids plus grands sont de diamètre supérieur. On observe quelques différences avec la figure 6, obtenue avec `OpenTURNS`. Par exemple, la quadrature de niveau 3 présente 13 noeuds avec `OpenTURNS`, mais 14 noeuds avec `chaospy`. Visuellement, on compte $1 + 4 + 2 \times 4 = 13$ noeuds seulement, ce qui montre que `chaospy` pourrait avoir une anomalie de ce point de vue dans l'algorithme de fusion des noeuds.

Exemple

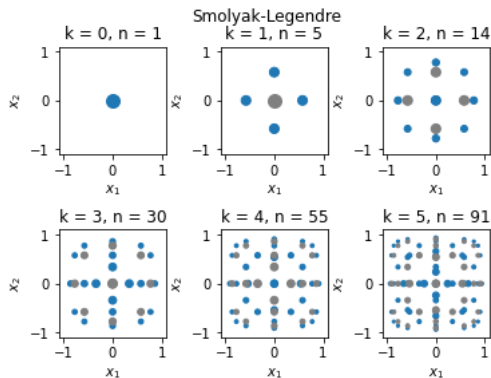


Figure 11 – Noeuds de la quadrature de Smolyak en dimension $p = 2$ de niveau $k = 0$ à 5 sur une construction de Gauss-Legendre avec `chaospy`.

Nombre de noeuds

Soit Γ_ℓ^1 l'ensemble des noeuds de la quadrature en dimension 1. On suppose que $\Gamma_0^1 = \emptyset$. On peut démontrer que si la règle unidimensionnelle est imbriquée, alors la règle multidimensionnelle de Smolyak est également imbriquée.

Supposons que $n_\ell^1 = O(2^\ell)$. Dans une grille par tensorisation complète, le nombre de noeuds est ([Gerstner and Griebel, 1998] page 216) :

$$n_{\text{tensorisation}} = O(2^{\ell d}).$$

Nous allons voir que la quadrature de Smolyak réduit considérablement ce nombre. Soit m_ℓ le nombre de noeuds de la règle de quadrature unidimensionnelle de niveau ℓ . Le nombre de points dans la grille creuse est :

$$n_\ell^d = \sum_{\|\mathbf{k}\|_1 \leq \ell + d - 1} m_{k_1} \cdots m_{k_d}.$$

Si $n_\ell^1 = O(2^\ell)$ alors le nombre de noeuds de la construction de Smolyak est :

$$n_{\text{Smolyak}} = O(2^\ell \ell^{d-1}).$$

Nombre de noeuds

La figure 12 présente le nombre de noeuds avec Smolpack⁷.

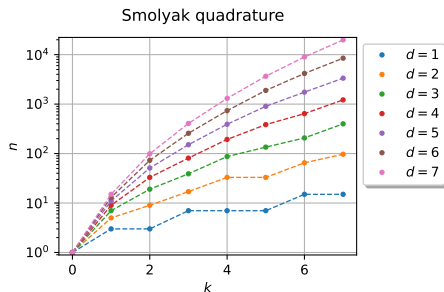


Figure 12 – Nombre de noeuds de la quadrature de Smolyak en fonction du niveau pour différentes dimensions avec Smolpack.

7. [smolyak_quadrature.py](#)

Nombre de noeuds

La figure 13 présente le nombre de noeuds avec OpenTURNS⁸. On utilise ici une quadrature marginale de Gauss-Legendre.

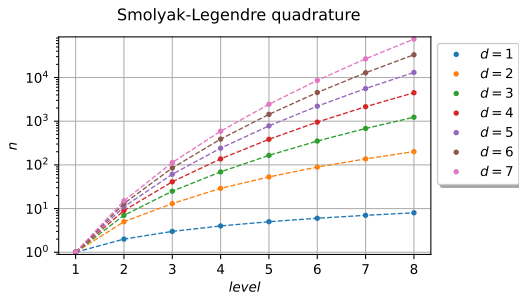


Figure 13 – Nombre de noeuds de la quadrature de Smolyak en fonction du niveau pour différentes dimensions avec OpenTURNS.

8. [plot_smolyak_experiment.py](#)

Nombre de noeuds

La figure 13 présente le nombre de noeuds de la quadrature de Smolyak et de la quadrature tensorisée de Gauss avec OpenTURNS⁹. On fait l'hypothèse que $n_{\text{tensorisation}} = 2^{\ell d}$ et $n_{\text{Smolyak}} = 2^{\ell} \ell^{d-1}$ ([Gerstner and Griebel, 1998] page 216).

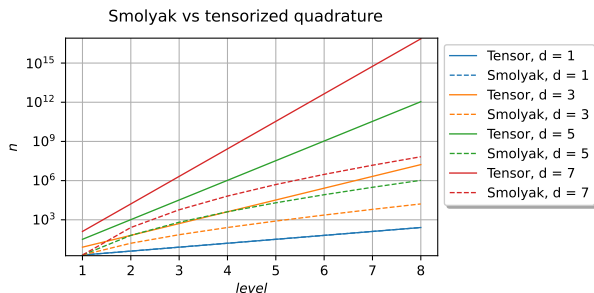


Figure 14 – Nombre de noeuds de la quadrature de Smolyak et de la quadrature tensorisée de Gauss en fonction du niveau pour différentes dimensions avec OpenTURNS.

9. [plot_smolyak_experiment.py](#)

Nombre de noeuds

La quadrature de Smolyak possède un nombre de noeuds croissant en fonction du niveau de la quadrature. Toutefois, la figure 15 montre que cette croissance est plus modérée que la quadrature tensorisée complète.

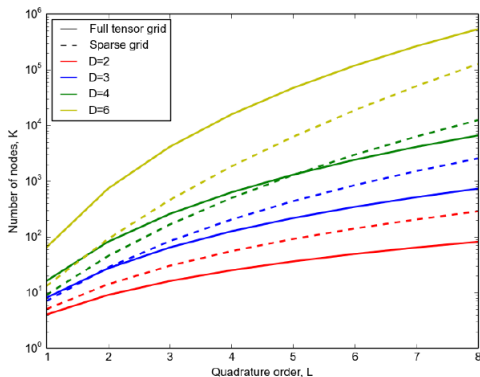


Figure 15 – Nombre de noeuds de la quadrature de Smolyak et de la quadrature tensorisée ([Feinberg and Tennøe., 2015b] page 15).

Erreurs d'arrondi

Une des difficultés de la quadrature de Smolyak est la présence de poids négatifs. Cette difficulté intervient à deux niveaux.

- ▶ Dans la création du plan d'expériences réduit par l'algorithme de fusion, on doit sommer les poids associés à des noeuds superposés.
- ▶ Dans l'évaluation de la règle de quadrature, on doit sommer les valeurs de la fonction pondérée par les poids.

Dans ces deux cas, l'implémentation naïve utilise la somme de deux nombres flottants, ce qui peut mener à une perte de précision. Une amélioration possible consisterait à utiliser par exemple un algorithme de somme compensée “à la Kahan”.

Convergence

We now create an exponential product problem from¹⁰, table 1, page 221. This example is also used in¹¹ page 221 to demonstrate the properties of Smolyak's quadrature. It is defined by the equation :

$$g(\mathbf{x}) = (1 + 1/d)^d \prod_{i=1}^d x_i^{1/d}$$

for any $\mathbf{x} \in [0, 1]^d$ where d is the dimension of the problem. We are interested in the integral :

$$I = \int_{[0,1]^d} g(\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$$

where $f(\mathbf{x}) = 1$ is the uniform density probability function in the $[0, 1]^d$ interval. When the dimension increases, the variance goes to zero, but the variation in the sense of Hardy and Krause goes to infinity.

10. Morokoff, W. J., & Caflisch, R. E. (1995). *Quasi-monte carlo integration*. Journal of computational physics, 122(2), 218-230.

11. Gerstner, T., & Griebel, M. (1998). Numerical integration using sparse grids. *Numerical algorithms*, 18(3), 209-232.

Convergence

Pour cela, nous utilisons les méthodes suivantes :

- la construction de Smolyak associée à des quadratures marginales unidimensionnelles par quadrature de Gauss-Legendre,
- la quadrature de Gauss-Legendre tensorisée.

La figure 8 présente la convergence de la construction de Smolyak et de la quadrature de Gauss tensorisée en fonction du nombre de noeuds. On observe que la quadrature de Gauss-Legendre est un peu moins précise, mais que la vitesse de convergence est semblable.

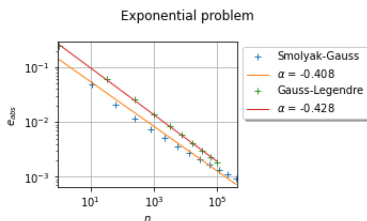


Figure 16 – Problème exponentiel avec OpenTURNS : convergence de la construction de Smolyak et de la quadrature de Gauss tensorisée en fonction du nombre de noeuds.

Convergence

La figure 9 présente la quadrature de Smolyak avec différentes règles de quadrature marginales. On observe que la quadrature de Smolyak est plus précise dans ce cas particulier lorsqu'on utilise la règle de quadrature marginale de Gauss-Legendre.

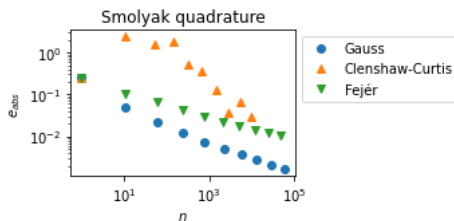


Figure 17 – Problème exponentiel avec chaospy : convergence de la construction de Smolyak avec différentes quadratures marginales en fonction du nombre de noeuds.

Références I



Feinberg, J. and Tennøe., S. (2015a).
Polynomial chaos expansions part 1 : Method introduction.



Feinberg, J. and Tennøe., S. (2015b).
Polynomial chaos expansions part 2 : Practical implementation.



Garcke, J. (2006).
Sparse grid tutorial.
Mathematical Sciences Institute, Australian National University, Canberra Australia, page 7.



Gerstner, T. and Griebel, M. (1998).
Numerical integration using sparse grids.
Numerical algorithms, 18(3) :209–232.



Griebel, M., Schneider, M., and Zenger, C. (1992).
A combination technique for the solution of sparse grid problems, iterative methods in linear algebra, p. de groen and r. beauwens.

Références II



Le Maître, O. and Knio, O. M. (2010).

Spectral methods for uncertainty quantification : with applications to computational fluid dynamics.

Springer Science & Business Media.



Morokoff, W. J. and Caflisch, R. E. (1995).

Quasi-monte carlo integration.

Journal of computational physics, 122(2) :218–230.



Novak, E. and Ritter, K. (1996).

High dimensional integration of smooth functions over cubes.

Numerische Mathematik, 75(1) :79–97.



Novak, E. and Ritter, K. (1999).

Simple cubature formulas with high polynomial exactness.

Constructive approximation, 15(4) :499–522.



Petras, K. (2001).

Asymptotically minimal smolyak cubature.

Références III



Petras, K. (2003).

Smolyak cubature of given polynomial degree with few nodes for increasing dimension.

Numerische Mathematik, 93(4) :729–753.



Smolyak, S. A. (1963).

Quadrature and interpolation formulas for tensor products of certain classes of functions.

148(5) :1042–1045.



Sullivan, T. J. (2015).

Introduction to uncertainty quantification, volume 63.
Springer.



Tempone, R. and Wolfers, S. (2018).

Smolyak's algorithm : A powerful black box for the acceleration of scientific computations.

In *Sparse Grids and Applications-Miami 2016*, pages 201–228. Springer.

Références IV



Wasilkowski, G. W. and Wozniakowski, H. (1995).

Explicit cost bounds of algorithms for multivariate tensor product problems.
Journal of Complexity, 11(1) :1–56.



Winokur, J. and Red-Horse, J. R. (2016).

Smolyak multi-dimensional methods.

Technical Report No. SAND2016-0946PE, Sandia National Lab.(SNL-NM),
Albuquerque, NM (United States).