Chapter 4
Sparse Grid Quadrature

This chapter is concerned with sparse grid (SG) quadrature methods. These methods are constructed using certain combinations of tensor products of one-dimensional quadrature rules. They can exploit the smoothness of f, overcome the curse of dimension to a certain extent and profit from low effective dimensions, see, e.g., [16, 44, 45, 57, 116, 146].

The remainder of this chapter is structured as follows: Section 4.1 is concerned with the construction of several variants (classical, delayed, generalised, dimension-adaptive) of the general sparse grid approach. Then, in Section 4.2 generalised sparse grid constructions are derived which satisfy certain optimality criteria for integrands from weighted Sobolev spaces. Finally, in Section 4.3, we relate sparse grids to the anchored-ANOVA decomposition and show that sparse grids methods can be regarded as special cases of the general class (3.18) of dimension-wise quadrature methods.

4.1 Sparse Grid Methods

The sparse grid approach is based on a decomposition of the d-dimensional integral If into a infinite telescope sum and on a truncation of this sum which balances work and accuracy. Different ways to truncate the sum thereby correspond to different sparse grid constructions such as the *classical construction* [146], the *delayed construction* [130], the *generalised construction* [45, 66, 132, 164] and the *dimension-adaptive construction* [46] which we will summarise in the next sections.

M. Holtz, *Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance*, Lecture Notes in Computational Science and Engineering 77,
DOI 10.1007/978-3-642-16004-2_4,

4.1.1 Classical Construction

Sparse grids (SG) can be defined for general tensor product domains $\Omega^d \subseteq \mathbf{R}^d$. We here consider the case $\Omega = [0,1]$, but most results can be generalised in a straightforward way to other domains. We will give remarks on the case $\Omega = \mathbf{R}$.

For a univariate function $f : [0,1] \to \mathbf{R}$ and a sequence of non-decreasing integers m_k, $k \in \mathbb{N}$, let

$$U_{m_k} f := \sum_{i=1}^{m_k} w_{i,k} f(x_{i,k}) \tag{4.1}$$

denote a sequence of univariate quadrature rules with m_k points $x_{i,k}$ and weights $w_{i,k}$, which converges pointwise to If for $k \to \infty$. We assume $m_1 = 1$ and $U_{m_1} f = f(1/2)$ and define the difference quadrature formulae

$$\Delta_k := U_{m_k} - U_{m_{k-1}} \quad \text{with } U_{m_0} := 0 \tag{4.2}$$

for $k \geq 1$.

Now let $f : [0,1]^d \to \mathbf{R}$ be a multivariate function. Then, the d-dimensional integral If can be represented by the infinite telescoping sum

$$If = \sum_{\mathbf{k} \in \mathbb{N}^d} \Delta_{\mathbf{k}} f \tag{4.3}$$

which collects the products of each possible combination of the univariate difference formulae. Here, $\mathbf{k} \in \mathbb{N}^d$ denotes a multi-index with $k_j > 0$ and

$$\Delta_{\mathbf{k}} f := \left(\Delta_{k_1} \otimes \ldots \otimes \Delta_{k_d}\right) f. \tag{4.4}$$

For a given level $\ell \in \mathbb{N}$, the *classical sparse grid method*, often also denoted as Smolyak method, see, e.g., [146, 116, 45], is then defined by

$$\mathrm{SG}_\ell f := \sum_{|\mathbf{k}|_1 \leq \ell + d - 1} \Delta_{\mathbf{k}} f \tag{4.5}$$

where $|\mathbf{k}|_1 := \sum_{j=1}^d k_j$.

From the set of all possible indices $\mathbf{k} \in \mathbb{N}^d$ thus only those are considered whose $|\cdot|_1$-norm is smaller than a constant. Note that the product approach (see Section 3.1.3) is recovered if instead of the $|\cdot|_1$-norm in (4.5) the norm $|\cdot|_\infty := \max\{k_1, \ldots, k_d\}$ is used for the selection of indices.

Example 4.1. For $d = 2$ and the case that U_{m_k} is the open trapezoidal rule with $m_k = 2^k - 1$ nodes, we show in Figure 4.1 the sparse grid $\mathrm{SG}_\ell f$ on level $\ell = 3$ and the product grids $\Delta_{(k_1,k_2)} f$ with $k_1 + k_2 \leq 4$ which are required for its construction according to the representation (4.5).

In Figure 4.2 we show the corresponding sparse grid $\mathrm{SG}_5 f$ and product grid $U_{m_k} f \otimes U_{m_k} f$ with $m_k = 2^\ell - 1 = 31$. One can see that the sparse grid contains

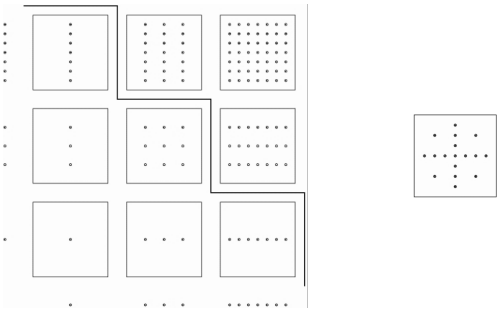


Fig. 4.1 Product grids $\Delta_{(k_1,k_2)}f$ for $1 \le k_j \le 3$ for $j = 1,2$ based on the univariate trapezoidal rules with $1,3$ and 7 points (left) and corresponding sparse grid $\mathrm{SG}_3 f$ (right). The figure is taken from [44].

much fewer nodes than the corresponding product grid. This effect is even more pronounced in higher dimensions.

Based on the representation

$$\mathrm{SG}_\ell f = \sum_{l=d}^{\ell+d-1} \sum_{|\mathbf{k}|_1=l} \Delta_{\mathbf{k}} f$$

the SG method (4.5) can be implemented as shown in Algorithm 4.1. Note that $\Delta_{\mathbf{k}} f$ is a d-dimensional product quadrature rule with $n_{\mathbf{k}} := \prod_{j=1}^{d} m_{k_j}$ points and can hence be computed by Algorithm 3.2. To find the indices $\mathbf{k} \in \mathbb{N}^d$ with $|\mathbf{k}|_1 = l$ we use Algorithm 4.2. This drop algorithm is taken from [44] but stated in an incremental form which increases the flexibility and readability of the code in our opinion.

Example 4.2. For $d = 2$ and $l = 2,3,4$ the Algorithm 4.2 returns the indices $\mathbf{k} = (k_1,k_2)$ in the order

$$\begin{aligned} &(1,1) \quad \text{for } l = 2 \\ &(2,1), (1,2) \quad \text{for } l = 3 \\ &(3,1), (2,2), (1,3) \quad \text{for } l = 4. \end{aligned}$$

These indices enumerate the product grids required for the computation of $\mathrm{SG}_3 f$ as illustrated in Figure 4.1.

The SG method (4.5) can exploit the smoothness of f and in this way also obtain convergence rates larger than one. In addition, it can overcome the curse of dimension to a certain extent. The function classes for which this is possible are the spaces

$$H^r([0,1]^d) := \left\{ f : [0,1]^d \to \mathbb{R} : \max_{|\mathbf{k}|_\infty \le r} \left\| \frac{\partial^{|\mathbf{k}|_1} f}{\partial x_1{}^{k_1} \dots \partial x_d{}^{k_d}} \right\|_\infty < \infty \right\}$$

Algorithm 4.1: Implementation of the sparse grid method (4.5).

$If = 0$ and $m = \ell + d - 1$
for $l = d, \ldots, m$ **do**
  $p = 1$, $\mathbf{k} = (m, 1, \ldots, 1)$ and $\hat{\mathbf{k}} = (m, \ldots, m)$
  repeat
    compute the the product formula $\Delta_{\mathbf{k}} f$ by Alg. 3.2
    $If = If + \Delta_{\mathbf{k}} f$
    det. the next index $\mathbf{k}$ with $|\mathbf{k}|_1 = l$ by Alg. 4.2 `/* input: p, k and k̂ */`
  until $\mathbf{k} = \mathbf{0}$;
return $\mathrm{SG}_\ell f = If$

Algorithm 4.2: Drop algorithm for the iterative enumeration of all indices $\mathbf{k} \in \mathbb{N}^d$, $k_j > 0$ with $|\mathbf{k}|_1 = l$.

Input: $p \in \mathbb{N}$ and $\mathbf{k}, \hat{\mathbf{k}} \in \mathbb{N}^d$
repeat
  $k_p = k_p + 1$
  if $k_p > \hat{k}_p$ **then**
    if $p = d$ **then return** $\mathbf{0}$ `/* all indices already enumerated */`
    $k_p = 1$
    $p = p + 1$
  else
    for $j = 1, 2, \ldots, p-1$ **do**
      $\hat{k}_j = \hat{k}_p - k_p + 1$
    $k_1 = \hat{k}_1$
    $p = 1$
    return $\mathbf{k}$ `/* the next index to use */`
until $\mathbf{k} = \mathbf{0}$;

of functions which have bounded mixed partial derivatives of order r. Let n denote the number of points in the sparse grid. Then the error of the classical SG quadrature formula is bounded by

$$\varepsilon(n) = O(n^{-r} (\log n)^{(d-1)(r-1)}) \tag{4.6}$$

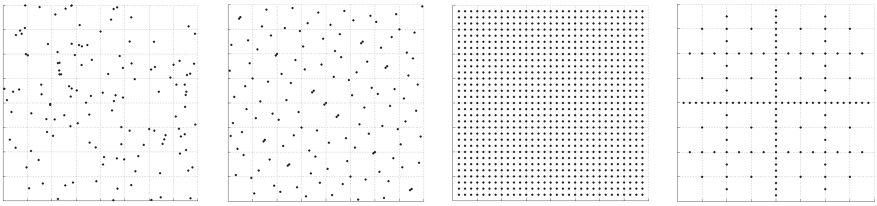
for all $f \in H^r([0,1]^d)$.[1] The convergence rate is thus independent of the dimension up to a logarithmic factor. For analytic functions even spectral convergence is observed. In addition, the SG method (4.5) achieves (up to logarithmic factors) optimal rates of convergence for the classical spaces $C^r([0,1]^d)$ which have been defined in Section 3.1.3. These rates depend exponentially on the dimension, however, see [116].

[1] We will give a proof of this result for the case $r = 1$ in Section 4.2.

SG quadrature formulas come in various types depending on the one-dimensional basis integration routine. SG methods with *Clenshaw-Curtis* rules are used in [116]. *Gauss-Patterson* and *Gauss-Legendre* formulas are investigated in [45]. For Gaussian integration, SG methods with *Gauss-Hermite* and *Genz-Keister* rules are used in [113, 118] and [68], respectively.

In many cases, the performance of the classical sparse grid method can be enhanced by the use of *delayed sequences* of one-dimensional quadrature rules (see Section 4.1.2), by *spatial adaptivity* [9, 14] or by a *dimension-adaptive refinement* (see Section 4.1.4).

Remark 4.1. Sparse grid methods can directly be applied to the numerical computation of integrals on $\mathbf{R}^d$ with Gaussian weight. To this end, only the sequence of univariate quadrature rules U_{m_k} must be replaced by quadrature formulas for functions $f : \mathbb{R} \to \mathbb{R}$ on unbounded domains, such as Gauss-Hermite or Genz-Keister rules, see, e.g., [68, 113, 118].



(a) Monte Carlo (b) Quasi-MC (c) Product grid (d) Sparse grid

Fig. 4.2 Sample grids of four different classes of methods for multivariate numerical integration. The QMC points come from the Sobol low discrepancy sequence. The product grid and the sparse grid are based on the trapezoidal rule with 31 nodes.

4.1.2 Delayed Basis Sequences

This section is concerned with an optimization of the classical sparse grid construction with respect to its polynomial degree of exactness which has been proposed in [130].

To this end, we denote by $\mathscr{P}_\ell^d$ the space of all multivariate polynomials in d dimensions which have the maximum degree ℓ. We assume that the univariate formulas U_{m_ℓ} are exact for the space $\mathscr{P}_\ell^1$. Then the classical sparse grid quadrature rule SG_ℓ is exact for the non-classical space of polynomials

$$\tilde{\mathscr{P}}_\ell^d := \left\{ \mathscr{P}_{k_1}^1 \otimes \ldots \otimes \mathscr{P}_{k_d}^1 : |\mathbf{k}|_1 = \ell + d - 1 \right\},$$

see [116]. The method also preserves the classical polynomial exactness (based on the space $\mathscr{P}_\ell^d$) of the univariate quadrature rules to a certain extend as the following lemma shows. The lemma is taken from [117].

Lemma 4.1. *Let m_i, $i \in \mathbb{N}$, be a non-decreasing sequence of integers. Let further U_{m_i} denote a univariate quadrature rule with m_i points and*

$$\deg(U_{m_i}) \geq 2i-1. \tag{4.7}$$

Then,

$$\deg(SG_\ell) \geq 2\ell - 1, \tag{4.8}$$

where SG_ℓ is the sparse grid method (4.5) based on the sequence U_{m_i}.

Proof. The proof proceeds via induction over d using dimension recursive formulations [164] of the method SG_ℓ. We refer to [117] and [130] for details.

The classical SG quadrature rule thus integrates all polynomials up to a certain degree exactly. The number of points thereby only increases polynomially with the dimension. Nevertheless, the number of points increases still rather fast. In [130], modifications of the classical SG construction are presented which have the same degree of polynomial exactness with a fewer number of points. To construct these methods, *delayed basis sequences* are used which result from a repetition of quadrature formulas in the univariate basis sequences.

Delayed Gauss-Patterson sequences

In the classical SG construction the sequence $m_i = 2^i - 1$ is used. If U_{m_i} is the Gauss-Patterson rule then $\deg(U_{m_i}) = 3/2m_i + 1/2$ for $i > 1$. The degree of exactness thus increases considerably faster than $2i-1$, see Table 4.1(a). To avoid this overhead, it is proposed in [130] to base the SG construction on a different sequence $\tilde{m}_i$ which increases slower with i than the sequence m_i. The sequence $\tilde{m}_i$ is thereby determined as

$$\tilde{m}_i = 1,3,3,7,7,7,15,\ldots \quad \text{for } i = 1,2,\ldots \tag{4.9}$$

in such a way that $\deg(U_{\tilde{m}_i})$ increases as slow as possible but is always larger than $2i-1$. This way, the number of points of SG_ℓ increases much slower with the level ℓ while (4.8) holds as in the classical SG construction. The degrees of polynomial exactness of U_{m_i} and $U_{\tilde{m}_i}$ are shown in Table 4.1(a) for $i = 1,\ldots,7$.

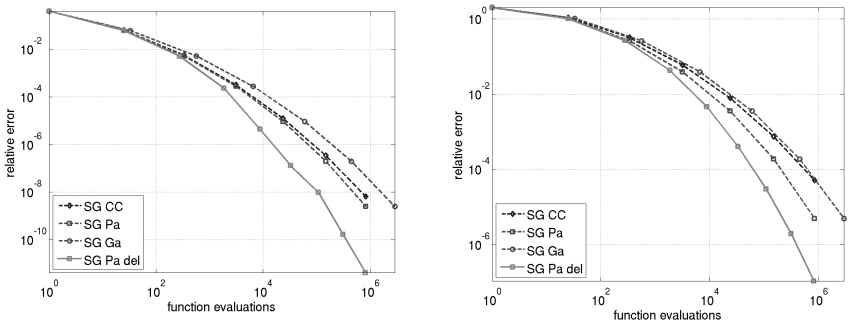
Remark 4.2. It is shown in [130] that the SG method with the sequence $U_{\tilde{m}_i}$ is almost asymptotically optimal and that there are only minor improvements possible. Also the numerical experiments presented in [130] indicate that the delayed constructions are more efficient than their non-delayed counterparts provided smooth functions are integrated, since such functions are well approximated by polynomials.

Example 4.3. We apply for illustration several SG variants for the integration of the quadrature test functions

$$f_1(\mathbf{x}) := \exp\left(-\sum_{j=1}^{d} x_i^2\right) \quad \text{and} \quad f_2(\mathbf{x}) := \prod_{j=1}^{d}\left(1+\left(\frac{3}{4}\right)^j\left(\frac{\pi}{4}\sin(\pi x_j)-\frac{1}{2}\right)\right)$$

with $d = 8$. The function f_1 is from the testing package of Genz [39]. The function f_2 is a slight modification[2] of a test function from [160].

The results are shown in Figure 4.3. There we observe the same effect as described in Remark 4.2.



(a) $Function f_1$ (b) $Function f_2$

Fig. 4.3 Errors and required number of function evaluations of the SG method with Clenshaw-Curtis (SG CC), Gauss-Patterson (SG Pa), Gauss-Legendre (SG Ga) and delayed Gauss-Patterson (SG Pa del).

Slowly increasing Gauss-Legendre sequences

The approach from [130] is not restricted to SG methods based on Gauss-Patterson formulas. It can be used to determine improved sequences $\tilde{m}_i$ for SG methods based on arbitrary univariate quadrature rules. To this end, we determine the slowest increasing sequence $\tilde{m}_i$, $i = 1,2,\ldots$, which fulfils (4.7). The sparse grid method corresponding to this sequence then satisfies the condition (4.8) as in the classical case but requires fewer points.

For the Gauss-Legendre rule, which has a higher polynomial degree than the Gauss-Patterson rule, but is not nested, we obtain

$$\tilde{m}_i = i \quad \text{for } i = 1,2,\ldots.$$

In this way, the degree of exactness $\deg(U_{\tilde{m}_i}) = 2i-1$ exactly equals the minimum requirement, see Table 4.1(b).

[2] In [160] specific polynomials are used to study the performance of QMC methods for integrands from weighted spaces. Since SG methods can integrate these polynomials exactly we replaced all monomials x_j by $\frac{\pi}{4}\sin(\pi x_j)$.

Slowly increasing Gauss-Hermite sequences

Next, we consider possible improvements of the classical construction of SG methods for integrals on $\mathbf{R}^d$ with Gaussian weight with respect to its polynomial degree.

SG methods with Gauss-Hermite rules and the sequence $m_i = 2^i - 1$ have been used in [113, 118]. The degree of polynomial exactness of the m_i-point Gauss-Hermite rule U_{m_i} is $2m_i - 1$. Hence, its degree increases much faster than $2i - 1$ in the classical construction where $m_i = 2^i - 1$. For the Gauss-Hermite rule, the slowest increasing sequence subject to (4.7) is given by

$$\tilde{m}_i = i \quad \text{for } i = 1, 2, \ldots.$$

It exactly satisfies the minimum requirement $\deg(U_{\tilde{m}_i}) = 2i - 1$, see Table 4.1(c). SG methods with such slowly increasing Gauss-Hermite sequences maintain the condition (4.8) from the classical case but require fewer points on level ℓ.

Delayed Genz-Keister sequences

Finally, we consider Genz-Keister rules. They have a lower degree as Gauss-Hermite rules but have the advantage that they are nested. SG methods with these rules have been used in [68] with the sequence $m_i = 1, 3, 9, 19, 35$.

In order to reduce the number of points on level ℓ while keeping at the same time the property (4.8), we again determine the slowest increasing sequence subject to (4.7). One obtains the delayed version

$$\tilde{m}_i = 1, 3, 3, 9, 9, 9, 9, 19, \ldots \quad \text{for } i = 1, 2, \ldots. \tag{4.10}$$

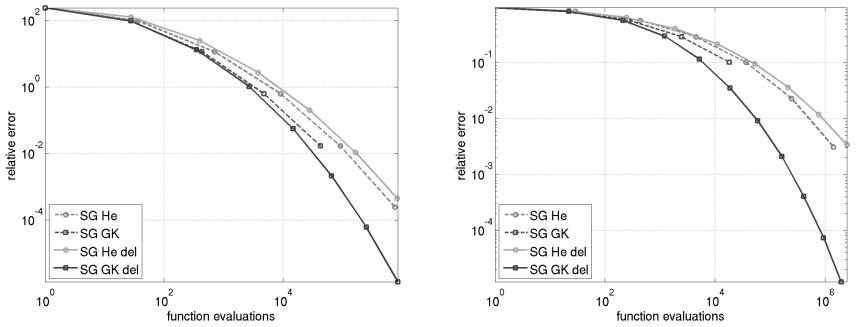
The degrees of exactness of U_{m_i} and $U_{\tilde{m}_i}$ are shown in Table 4.1(d).

Example 4.4. To provide numerical examples which illustrate the improvement that results from the delay of basis sequences, we consider the quadrature test functions

$$f_3(\mathbf{x}) := \cos\left(\sum_{i=1}^{d} x_i^2\right) \varphi_d(\mathbf{x}) \quad \text{and} \quad f_4(\mathbf{x}) := \exp\left(\sum_{i=1}^{d} x_i\right) \varphi_d(\mathbf{x})$$

with $d = 9$ and $d = 7$, respectively. Here, φ_d denotes the standard Gaussian density (2.16) in d dimensions. The function f_3 is also used in [124, 127] to show that QMC methods can be superior to MC methods also for isotropic problems. The function f_4 corresponds to a simplified version of the problem to price Asian options, see [160].

One can see in Figure 4.4 that the SG method with delayed Genz-Keister sequences is in both examples more efficient than its classical counterpart. While SG methods with non-delayed Genz-Keister rules can only be constructed until level five [42], higher levels and also higher accuracies can be achieved in the delayed case. In the two tests, the slowly increasing Gauss-Hermite sequences turn out to be



(a) $Function f_3$ (b) $Function f_4$

Fig. 4.4 Errors and required number of function evaluations of the SG method with Gauss-Hermite (SG He), Genz-Keister (SG GK), slowly increasing Gauss-Hermite (SG He del) and delayed Genz-Keister (SG GK del).

slightly less effective than the classical Gauss-Hermite sequences. Nevertheless, we will see in Section 4.1.4 that dimension-adaptive SG methods can benefit from such sequences as then the grid refinement can be performed more fine granular.

Table 4.1 Degrees of polynomial exactness of different univariate quadrature rules with the classical sequence m_i and with the delayed sequence $\tilde{m}_i$. The lower bound is $2i-1$.

(a) Gauss-Patterson

i	1	2	3	4	5	6	7
$\deg(U_{m_i})$	1	5	11	23	47	95	191
$\deg(U_{\tilde{m}_i})$	1	5	5	11	11	11	13
$2i-1$	1	3	5	7	9	11	13

(b) Gauss-Legendre

i	1	2	3	4	5	6	7
$\deg(U_{m_i})$	1	5	13	29	61	125	252
$\deg(U_{\tilde{m}_i})$	1	3	5	7	9	11	13
$2i-1$	1	3	5	7	9	11	13

(c) Gauss-Hermite

i	1	2	3	4	5	6	7
$\deg(U_{m_i})$	1	5	13	29	61	125	252
$\deg(U_{\tilde{m}_i})$	1	3	5	7	9	11	13
$2i-1$	1	3	5	7	9	11	13

(d) Genz-Keister

i	1	2	3	4	5	6	7
$\deg(U_{m_i})$	1	5	15	29	51	-	-
$\deg(U_{\tilde{m}_i})$	1	5	5	15	15	15	15
$2i-1$	1	3	5	7	9	11	13

4.1.3 Generalised Sparse Grids

The sparse grid construction can be tailored to certain classes of integrands if a priori information about the importance of the dimensions or the importance of the interactions between the dimensions is available. This is achieved by choosing appropriate index sets $\mathscr{I} \subset \mathbb{N}^d$ in the representation (4.3).

To ensure the validity of the telescoping sum expansion, the index set $\mathscr{I}$ has to satisfy the admissibility condition

$$\mathbf{k} \in \mathscr{I} \text{ and } \mathbf{l} \leq \mathbf{k} \quad \Longrightarrow \quad \mathbf{l} \in \mathscr{I}, \tag{4.11}$$

where $\mathbf{l} \leq \mathbf{k}$ is defined by $l_j \leq k_j$ for $j = 1, \ldots, d$.

In this way, the *generalised sparse grid method*

$$\mathrm{SG}_{\mathscr{I}} f := \sum_{\mathbf{k} \in \mathscr{I}} \Delta_{\mathbf{k}} f \tag{4.12}$$

is defined, see, e.g., [45, 66, 132, 164].

Different ways to truncate the sum (4.3) then correspond to different quadrature methods. Examples are the product approach (3.8), the classical SG construction (4.5) and SG methods with delayed basis sequences. Moreover, SG methods based on weighted norms $|\mathbf{k}|_{1,\mathbf{a}} := \sum_{j=1}^{d} a_j k_j$, where $\mathbf{a} \in \mathbb{R}_+^d$ is a weight vector for the different dimensions (see, e.g., [45]), or SG methods with finite-order weights, where the norm $|\mathbf{k}|_0 := \sum_{j=1}^{d} 1_{k_j > 1}$ is used for the selection of indices, are special cases of this approach.

Example 4.5. For $d = 2$ and the case that U_{m_k} is the open trapezoidal rule with $m_k = 2^k - 1$ nodes, we show in Figure 4.5 the sparse grid $\mathrm{SG}_{\mathscr{I}} f$ for a particular set $\mathscr{I}$. Moreover, the product grids $\Delta_{(k_1,k_2)} f$ are displayed which are required for its construction. One can see that the sparse grid is more refined into the y-direction than into the x-direction (compare also Figure 4.1), which can be advantageous if the second dimension of the integrand is more important than the first one to compute the integral value sufficiently accurate.

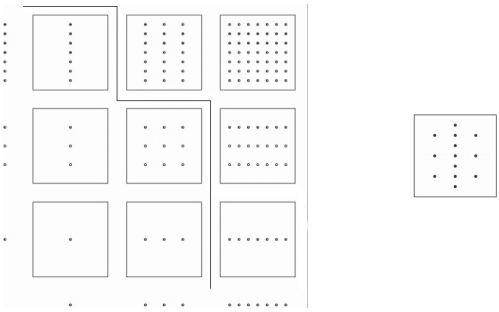
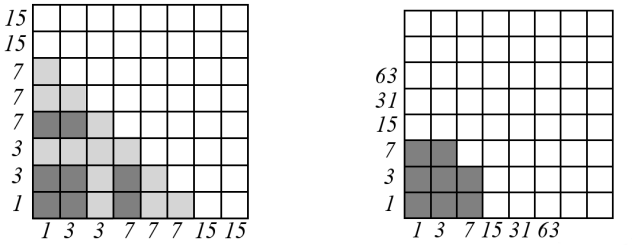


Fig. 4.5 Product grids $\Delta_{(k_1,k_2)} f$ for $1 \leq k_j \leq 3$ for $j = 1,2$ based on the univariate trapezoidal rules with $1,3$ and 7 points (left) and sparse grid $\mathrm{SG}_{\mathscr{I}} f$ with $\mathscr{I} = \{(1,1),(2,1),(1,2),(2,2),(1,3)\}$ (right). The figure is taken from [44].

Example 4.6. Using the example of the Gauss-Patterson sequence, we illustrate in Figure 4.6 the fact that SG methods with delayed basis sequences are special cases of the generalised SG method. Due to the delay of the basis sequence it holds the

equality $U_{m_{k-1}} = U_{m_k}$ and thus $\Delta_k = U_{m_k} - U_{m_{k-1}} = 0$ for several $k \in \mathbb{N}$. Therefore, also several product difference formulas $\Delta_{\mathbf{k}} = 0$ and do not contribute to the integral value. In Figure 4.6(a) these indices $\mathbf{k}$ are colored light gray. If these zero quadrature rules are omitted, one obtains a generalised index set as indicated in Figure 4.6(b).



(a) Sample index set of the SG method with the delayed Gauss-Patterson sequence

(b) Index set of the corresponding generalised SG method

Fig. 4.6 Illustration of the relation between SG methods based on delayed sequences and generalized SG methods.

4.1.4 Dimension-adaptive Sparse Grids

In practise, usually no a priori information about the dimension structure of the integrand is available. In these cases algorithms are required which can construct appropriate index sets $\mathscr{I}$ automatically during the actual computation. Such algorithms were presented in [46, 66] where the index sets are found in a dimension-adaptive way by the use of suitable error indicators.

The adaptive methods start with the smallest index set $\mathscr{I} = \{(1,\ldots,1)\}$. Then, step-by-step from the set of all admissible indices, the index $\mathbf{k}$ is added to $\mathscr{I}$ which has the largest $|\Delta_{\mathbf{k}} f|$ value and is therefore expected to provide an as large as possible error reduction, see [46, 44, 113] for details.

The resulting dimension-adaptive construction of the index set[3] is shown in Algorithm 4.3. There, $\mathscr{A}$ denotes the subset of all indices $\mathbf{k} \in \mathbb{N}^d \setminus \mathscr{I}$ which satisfy the admissibility condition (4.11) with respect to the set $\mathscr{I}$.

Example 4.7. For a two-dimensional example we show four steps of Algorithm 4.3 in Figure 4.7. There the index sets are (from left to right) given by

[3] Note that in [46] more sophisticated stop criterions are used than $|\Delta_{\mathbf{k}} f| \leq \varepsilon$.

$$\mathscr{I} = \{(1,1),(2,1),(1,2),(3,1),(1,3)\}$$
$$\mathscr{I} = \{(1,1),(2,1),(1,2),(3,1),(1,3),(2,2)\}$$
$$\mathscr{I} = \{(1,1),(2,1),(1,2),(3,1),(1,3),(2,2),(1,4)\}$$
$$\mathscr{I} = \{(1,1),(2,1),(1,2),(3,1),(1,3),(2,2),(1,4),(2,3)\}$$

Altogether, the algorithm allows for an adaptive detection of the important dimensions and heuristically constructs optimal index sets $\mathscr{I}$ in the sense of [15, 59]. Note that this is closely related to best N-term approximation [27].

Remember that we already presented a similar algorithm, Algorithm 3.4, in Section 3.2.4. We next compare these two algorithms. Recall that Algorithm 3.4 can be based on two *separate* types of adaptivity. The important anchored-ANOVA terms $If_{\mathbf{u}}$ are found in a dimension-adaptive fashion with help of the weights $\gamma_{\mathbf{u}}$ and are approximated by $q_{\mathbf{u}}$ using possibly locally adaptive methods. In Algorithm 4.3 the calculation of the contributions $\Delta_{\mathbf{k}}$ is more restrictive since the telescoping sum expansion has to hold. The algorithm is already completely determined by the choice of the univariate quadrature rule U_{m_k}. While Algorithm 3.4 has the advantage that low regularity of low order anchored-ANOVA terms can be resolved by local adaptivity, Algorithm 4.3 has the advantage that modeling and discretization errors are *simultaneously* taken into account and can thus be balanced in an more optimal way.

Remark 4.3. Note finally that dimension-adaptive SG methods do not profit from the *delay of basis sequences* since dimension-adaptive SG methods operate on the class of generalised SG methods. This class includes all SG methods based on delayed sequences as special cases as we explained in the previous section. Therefore, dimension-adaptive SG methods based, e.g., on the Gauss-Patterson sequence and dimension-adaptive SG methods based on the delayed Gauss-Patterson sequence provide the same results. Their grids equal to the grid of the classical SG method based on the delayed Gauss-Patterson sequence if this grid is optimal with respect to the error indicator used for the dimension-adaptive grid refinement.

Dimension-adaptive SG methods can however take advantage of *slowly increasing sequences* like Gauss-Hermite rules (see Section 4.1.2) as here the grid refinement can be performed more fine granular. In this way, the constant of the approximation can be reduced in particular for anisotropic integrands in high dimensions.

Algorithm 4.3: Dimension-adaptive sparse grid method.

Initialise: Let $\mathscr{I} = \{(1,\ldots,1)\}$ and $s = \Delta_{(1,\ldots,1)} f$.

repeat

1) Compute the values $\Delta_{\mathbf{k}} f$ from (4.4) for all $\mathbf{k} \in \mathscr{A}$ for which $\Delta_{\mathbf{k}} f$ has not yet been computed and set $s = s + \Delta_{\mathbf{k}} f$.

2) Move the index $\mathbf{k} \in \mathscr{A}$ with the largest weight $|\Delta_{\mathbf{k}} f|$ from $\mathscr{A}$ to $\mathscr{I}$.

until $|\Delta_{\mathbf{k}} f| \leq \varepsilon$;

Set $SG_{\mathscr{I}} f = s$.

We will present some numerical results in Chapter 6 which illustrate the efficiency of this particular SG method.

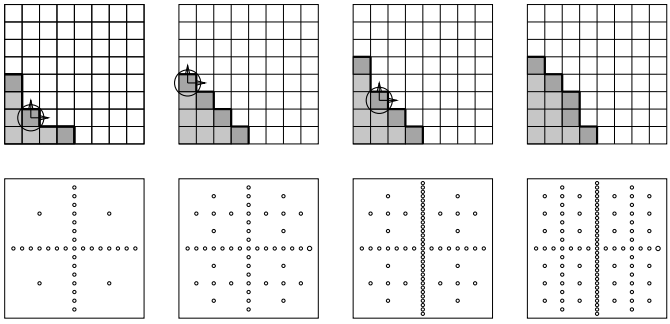


Fig. 4.7 Two-dimensional example for the construction of the index set $\mathscr{I}$ (top) and of the corresponding sparse grids $SG_{\mathscr{I}} f$ (bottom). The set $\mathscr{A}$ of admissible indices is dark-shaded. The index in $\mathscr{A}$ with the largest $|\Delta_{\mathbf{k}} f|$ value is encircled. This index is added to $\mathscr{I}$ in the next step of the algorithm. The figure is taken from [46].

4.2 Optimal Sparse Grids in Weighted Spaces

In this section we consider the generalised SG method (4.12) in more detail following mainly [58]. We determine the index set $\mathscr{I}$ of this method, which balances error and cost in an optimal way for integrands from weighted tensor product Sobolev spaces [144, 164]. To this end, we partly proceeded as in [16, 164].

For reasons of simplicity, we restrict ourselves to the case that the univariate quadrature rules U_{m_k} in (4.1) are given by the trapezoidal rule with $m_1 = 1$, $U_1 f = f(0)$ and $m_i = 1 + 2^{i-2}$ points for $i \geq 2$. The analysis is based on the univariate function space

$$H^1_\gamma([0,1]) := \{f : [0,1] \to \mathbb{R} : \|f\|_{1,\gamma} < \infty\}$$

with the norm

$$\|f\|^2_{1,\gamma} := f(0)^2 + \gamma^{-1} \|f'\|^2_{L_2}, \tag{4.13}$$

where $\gamma \in (0,1]$ denotes a weight. In the multivariate case we consider a given sequence of weights

$$1 = \gamma_1 \geq \gamma_2 \geq \ldots \geq \gamma_d \geq 0$$

and assign to each set $\mathbf{u} \subseteq \mathscr{D}$ the product weight $\gamma_{\mathbf{u}}$ from (3.22). We then define the tensor product space[4]

[4] Note that $H^{1,\mathrm{mix}}_\gamma([0,1]^d)$ is the reproducing kernel Hilbert space to the product kernel $K(\mathbf{x},\mathbf{y}) = \prod_{j=1}^d k(x_j, y_j)$, where $k(x,y) := 1 + \gamma \min\{x,y\}$ is the reproducing kernel of the space $H^1_\gamma([0,1])$. We provide more detail on reproducing kernel Hilbert spaces in Appendix A.1.

http://www.springer.com/978-3-642-16003-5

Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance

Holtz, M.

2011, VIII, 192 p. 32 illus., Hardcover
ISBN: 978-3-642-16003-5