Numerical Integration using Sparse Grids

Thomas Gerstner and Michael Griebel
Abteilung für Wissenschaftliches Rechnen und Numerische Simulation
Institut für Angewandte Mathematik
Universität Bonn
Wegelerstr. 6, D–53115 Bonn

Abstract

We present and review algorithms for the numerical integration of multivariate functions defined over d–dimensional cubes using several variants of the sparse grid method first introduced by Smolyak [51]. In this approach, multivariate quadrature formulas are constructed using combinations of tensor products of suited one–dimensional formulas. The computing cost is almost independent of the dimension of the problem if the function under consideration has bounded mixed derivatives. We suggest the usage of extended Gauss (Patterson) quadrature formulas as the one–dimensional basis of the construction and show their superiority in comparison to previously used sparse grid approaches based on the trapezoidal, Clenshaw–Curtis and Gauss rules in several numerical experiments and applications.

Classification: 65C20, 65D30, 65D32, 65M99, 65R20, 65U05, 65Y20, 90A09.
Keywords: Multivariate numerical quadrature, sparse grids, Smolyak's construction, complexity, curse of dimension.

1 Introduction

Multivariate integrals arise in many application fields, such as statistical mechanics, the valuation of financial derivatives, the discretization of partial differential and integral equations or the numerical computation of path integrals. Conventional algorithms for the numerical computation of such integrals are often limited by the "curse of dimension" meaning that the computing cost grows exponentially with the dimension of the problem. Moreover, theoretical complexity investigations reveal that also lower bounds for the computing cost grow exponentially with the dimension for many integration problems [54].

However, for special function classes, such as spaces of functions which have bounded mixed derivatives, Smolyak's construction [51] can overcome this curse of dimension to a certain extent. In this approach, multivariate quadrature formulas are constructed using combinations of tensor products of suited one–dimensional formulas. In this way, the number of function evaluations and the numerical accuracy get independent of the dimension of the problem up to logarithmic factors.

Smolyak's construction is known under various names, such as (discrete) blending method [22], Boolean method [11], or sparse grid method [58]. It has been applied to numerical integration by several authors, using the midpoint rule [2], the rectangle rule [41], the trapezoidal rule [3], the Clenshaw–Curtis rule [8, 37, 38] and Gauss rules [39] as the one–dimensional basis integration procedure. Further studies have been made concerning extrapolation methods [3], discrepancy measures [15] and complexity questions [57].

There is a large variety of other methods for the numerical integration of multivariate functions such as Monte Carlo and Quasi–Monte Carlo methods [35], lattice rules [50], adaptive subdivision methods [18, 55] and approximation methods based on neural networks [1, 32]. Each of these methods is particularly suitable for functions from a certain function class and has a complexity which is then also independent or nearly independent of the dimension of the problem.

Further applications of Smolyak's construction are the Fourier transformation [28], wavelet analysis [52], the solution of elliptic and hyperbolic partial differential equations [5, 23, 27, 58], integral equations [26, 46], eigenvalue problems [16], interpolation and approximation [11, 53], global optimization [36], data compression [20] and image reconstruction [56].

The scope of this paper is to review several methods based on Smolyak's approach and to introduce additional constructions based on univariate extended Gauss (Patterson) formulas which achieve the highest possible polynomial exactness among all nested quadrature formulas which use the same number of function evaluations. We also indicate some extensions and modifications of the method and show a numerically stable implementation. The performance of several variants of sparse grid quadrature formulas is compared in a variety of applications from computational physics and financial mathematics.

2 Problem Formulation

In the following, boldface letters indicate vectors or multi–indices. We consider the numerical integration of functions $f(\mathbf{x})$ from a function class $\mathcal{F}$ over the d–dimensional unit hypercube $\Omega := [-1,1]^d$,

$$I^d f := \int_\Omega f(\mathbf{x})\, d\mathbf{x},$$

by a sequence of n_l^d–point quadrature formulas with level $l \in \mathbb{N}$ and $n_l^d < n_{l+1}^d$,

$$Q_l^d f := \sum_{i=1}^{n_l^d} w_{li} \cdot f(\mathbf{x}_{li}).$$

using the weights w_{li} and abscissas $\mathbf{x}_{li}$. Furthermore, we define the underlying grid of a quadrature formula by

$$\Gamma_l^d := \{\mathbf{x}_{li} : 1 \leq i \leq n_l^d\} \subset [-1,1]^d.$$

Quadrature formulas are nested (imbedded) if the corresponding grids are nested, that is

$$\Gamma_l^d \subset \Gamma_{l+1}^d.$$

In order to compare the performance of different quadrature formulas we look at the quadrature error given by

$$E_l^d f := |I^d f - Q_l^d f|.$$

Bounds for the quadrature error can be obtained by assuming certain smoothness conditions for the function class $\mathcal{F}$ such as bounds on derivatives of functions $f \in \mathcal{F}$.

3 Nested Univariate Quadrature Formulas

In the following, we give a short review of nested univariate quadrature formulas for functions $f \in \mathcal{C}^r$ with

$$\mathcal{C}^r := \left\{ f : \Omega \longrightarrow \mathbb{R}, \left\| \frac{\partial^s f}{\partial x^s} \right\|_\infty < \infty, \; s \leq r \right\},$$

which are used in conjunction with Smolyak's construction. As we will see later, it is of great importance that $n_1^1 = 1$ and $n_l^1 = O(2^l)$. Therefore, in the following we always set

$$Q_1^1 = 2 \cdot f(0).$$

3.1 Trapezoidal Rule

The Newton–Cotes formulas [9] use equidistant points and determine the corresponding weights by integration of the Lagrange polynomials through these points. The closed versions include the endpoints of the interval, whereas the open ones omit one or both of them. The formulas get numerically instable for large numbers of points, i.e. some of the weights will become negative.

Therefore, iterated versions of low degree formulas are most commonly used [2, 3, 41]. A well known example is the iterated trapezoidal rule. Here we use

$$n_l^1 = 2^{l-1} + 1, \; l \geq 2,$$

and therefore have

$$Q_l^1 = \sum_{i=1}^{n_l^1}{}'' \; 2^{2-l} \cdot f((i-1) \cdot 2^{2-l} - 1).$$

The $\sum''$ indicates that the first and the last term of the sum are halved. The error bounds are well known and for functions $f \in \mathcal{C}^2$ of the form

$$|E_l^1 f| = O(2^{-2l}).$$

For $[-1,1]$–periodic functions $f \in \mathcal{C}^r$, this bound improves to

$$|E_l^1 f| = O(2^{-lr}).$$

Similar error bounds can be obtained for Simpson's rule and higher degree formulas.

3.2 Clenshaw–Curtis Formulas

The Clenshaw–Curtis formulas [7] are numerically more stable and use the non–equidistant abscissas given as the zeros or the extreme points of the Chebyshev polynomials. The quadrature formulas are nested in case the extreme points are used. We set for this case

$$n_l^1 = 2^{l-1} + 1, \ l \geq 2.$$

The abscissas are given by

$$x_{li} = -\cos \frac{\pi \cdot (i-1)}{n_l^1 - 1}$$

and the weights by

$$w_{l1} = w_{ln_l^1} = \frac{1}{n_l^1(n_l^1 - 2)} \quad \text{and}$$

$$w_{li} = \frac{2}{n_l^1 - 1} \left(1 + 2 \cdot {\sum_{j=1}^{(n_l^1-1)/2}}' \frac{1}{1-4j^2} \cdot \cos \frac{2\pi(i-1)j}{n_l^1 - 1} \right) \quad \text{for} \ \ 2 \leq i \leq n_l^1 - 1.$$

The $\sum'$ indicates that the last term of the sum is halved. The amount of work for the computation of the weights can be reduced to order $O(n_l^1 \log n_l^1)$ using a variant of the FFT algorithm [17]. The polynomial degree of exactness is $n_l^1 - 1$ and the error bounds for $f \in \mathcal{C}^r$ are therefore [9]

$$|E_l^1 f| = O(2^{-lr}).$$

A variant of the Clenshaw–Curtis formulas are the Filippi formulas in which the abscissas at the boundary of the interval are omitted. The number of points is chosen as in the Gauss–Patterson case which is described below. The degree of exactness is $n_l^1 - 1$ similar to the Clenshaw–Curtis formulas.

3.3 Gauss and Gauss–Patterson Formulas

Gauss formulas have the maximum possible polynomial degree of exactness of $2n-1$. For the case of the unit weight function the abscissas are the zeroes of the Legendre polynomials and the weights are computed by integrating the associated Lagrange polynomials. However, these Gauss–Legendre formulas are in general not nested.

Kronrod [31] extended an n–point Gauss quadrature formula by $n+1$ points such that the polynomial degree of exactness of the resulting $2n+1$ formula is maximal. This way, quadrature formulas with degree $2n + \bar{n} + 2$ with

$$\bar{n} := \begin{cases} n, & \text{if } n \text{ odd} \\ n-1, & \text{else} \end{cases} .$$

are obtained. For the Gauss–Legendre formula, the new abscissas are real, symmetric, inside the integration interval and interlace with the original points.

Furthermore, all weights are positive. It turned out [33], that the new abscissas are the zeros of the Stieltjes polynomial F_{n+1} satisfying

$$\int_{-1}^{1} P_n(x) F_{n+1}(x) x^j \, dx = 0, \quad \text{for} \quad j = 0, 1, \ldots, n,$$

where $P_n(x)$ is the n–th Legendre polynomial. Therefore, F_{n+1} can be seen as the orthogonal polynomial with respect to the weight function $P_n(x)$ which is of varying sign. The polynomial F_{n+1} can be computed by expanding it in terms of Legendre [43] or Chebyshev [47] polynomials and solving the resulting linear system. The zeroes of F_{n+1} can then be calculated by a modified Newton method. Alternatively, the computation of the abscissas can be achieved by the solution of a partial inverse eigenvalue problem [21].

Patterson [43] iterated Kronrod's scheme recursively and obtained a sequence of nested quadrature formulas with maximal degree of exactness. He constructed a sequence of polynomials $G_k(x)$ of degree $2^{k-1}(n+1), \; k \geq 1$, satisfying

$$\int_{-1}^{1} P_n(x) (\prod_{i=1}^{k-1} G_i(x)) G_k(x) x^j \, dx = 0 \quad \text{for} \quad j = 0, 1, \ldots, 2^{k-1}(n+1) - 1.$$

This way, $G_1(x) = F_{n+1}(x)$ and the G_j are orthogonal to all polynomials of degree less than $2^{k-1}(n+1)$ with respect to the variable signed weight function $P_n(x)(\prod_{i=1}^{j-1} G_i(x))$. The $2^k(n+1) - 1$ abscissas of the resulting quadrature formulas are the zeroes of P_n and all $G_j, 1 \leq j < k$. The abscissas and weights can be computed similar to the Kronrod case. This way, formulas of degree $(3 \cdot 2^{k-1} - 1)(n+1) + \bar{n}$ can be obtained – at least in theory.

However, Patterson extensions do not exist for all Gauss–Legendre formulas. For example, in the case of the 2–point Gauss–Legendre formula, only four extensions are possible [45]. But, starting with the 3–point formula, extensions exist for practicable k and all properties of Kronrod's scheme are preserved.

We set Q_2^1 equal to the 3–point Gauss–Legendre formula, and $Q_l^1, \; l \geq 3$, equal to its $(l-2)$nd Patterson extension. This way, $n_l^1 = 2^l - 1$ and the polynomial degree of exactness is $3 \cdot 2^{l-1} - 1$ for $l \geq 2$. The error is therefore for $f \in \mathcal{C}^r$ again

$$|E_l^1 f| = O(2^{-lr}).$$

3.4 Comparison

Of the considered nested quadrature formulas (with the restriction to periodic functions in the case of the trapezoidal rule) all achieve the optimal order of accuracy $O(2^{-lr})$. Among these, the Gauss–Patterson formulas achieve the highest possible polynomial exactness of nearly $(3/2)n_l^1$ compared to $n_1^l - 1$ for the Clenshaw–Curtis and Filippi formulas and 1 for the trapezoidal rule. From the results in [4] also follows that the Peano constants are smaller in comparison to the other formulas considered.

However, the existence of Patterson extensions is at the time not clear for large k, i.e. $k > 5$. Still, for Smolyak's construction the existing Patterson formulas are sufficient for moderate and high dimensional problems.

Note, that although the order of n_l^1 is the same in all cases, the actual number of points in the trapezoidal and Clenshaw–Curtis formulas compared to the Filippi, Gauss–Legendre and Patterson formulas can differ by almost a factor of 2 for the same level l.

4 Smolyak's Construction

Let us now consider the multivariate case. Smolyak [51] proposed a construction of multivariate quadrature formulas on the basis of one–dimensional quadrature formulas in a tensor product setting. The function classes he considered include functions with bounded mixed derivatives of order r, that is,

$$\mathcal{W}_d^r := \left\{ f : \Omega \longrightarrow \mathbb{R}, \ \left\| \frac{\partial^{|\mathbf{s}|_1} f}{\partial x_1^{s_1} \dots \partial x_d^{s_d}} \right\|_\infty < \infty, \ s_i \leq r \right\},$$

with $|\mathbf{s}|_1 := s_1 + \dots + s_d$. These spaces correspond to partially separable function spaces [38], and, in the case f is $[-1,1]^d$–periodic, to Korobov spaces [53]

$$\mathcal{E}_d^r := \{ f : \Omega \longrightarrow \mathbb{R}, \ a(m_1, \dots, m_d) = O(|\bar{m}_1 \cdot \dots \cdot \bar{m}_d|^{-r}) \},$$

with $r > 1$, $\bar{m}_j := \max\{1, m_j\}$ and $a(m_1, \dots, m_d)$ being the Fourier coefficients of the series

$$f(\mathbf{x}) = \sum_{m_1, \dots, m_d = -\infty}^{\infty} a(m_1, \dots, m_d) \cdot e^{-2\pi i (m_1 x_1 + \dots + m_d x_d)}.$$

4.1 Algorithm

First, consider a sequence of (not necessarily nested) one–dimensional quadrature formulas for a univariate function f,

$$Q_l^1 f := \sum_{i=1}^{n_l^1} w_{li} \cdot f(x_{li}).$$

Now, define the difference quadrature formula by

$$\begin{aligned} \Delta_k^1 f &:= (Q_k^1 - Q_{k-1}^1) f \quad \text{with} \\ Q_0^1 f &:= 0. \end{aligned}$$

In general, the difference formulas are therefore quadrature formulas on the union of the grids $\Gamma_k^1 \cup \Gamma_{k-1}^1$ (which is Γ_k^1 in the nested case). Smolyak's construction for d–dimensional functions f is then for $l \in \mathbb{N}$ and $\mathbf{k} \in \mathbb{N}^d$

$$Q_l^d f := \sum_{|\mathbf{k}|_1 \leq l+d-1} (\Delta_{k_1}^1 \otimes \dots \otimes \Delta_{k_d}^1) f.$$

The tensor product of d quadrature formulas $(Q_{l_1}^1 \otimes \dots \otimes Q_{l_d}^1)$ is hereby defined as the sum over all possible combinations

$$(Q_{l_1}^1 \otimes \dots \otimes Q_{l_d}^1) f := \sum_{i_1=1}^{n_{l_1}^1} \dots \sum_{i_d=1}^{n_{l_d}^1} w_{l_1 i_1} \cdot \dots \cdot w_{l_d i_d} \cdot f(x_{l_1 i_1}, \dots, x_{l_d i_d}).$$

Note, that a simple product formula is characterized by

$$(Q_l^1 \otimes \ldots \otimes Q_l^1) f = \sum_{j=1}^{d} \sum_{1 \leq k_j \leq l} (\Delta_{k_1}^1 \otimes \ldots \otimes \Delta_{k_d}^1) f,$$

and corresponds to a summation over the cube $|\mathbf{k}|_\infty \leq l$ with $|\mathbf{k}|_\infty := \max\{k_j\}$ instead of the simplex $|\mathbf{k}|_1 \leq l + d - 1$.

Alternatively, Smolyak's formula can be written in terms of $Q_{k_j}^1$ instead of $\Delta_{k_j}^1$ [10],

$$Q_l^d f = \sum_{l \leq |\mathbf{k}|_1 \leq l+d-1} (-1)^{l+d-|\mathbf{k}|_1-1} \cdot \begin{pmatrix} d-1 \\ |\mathbf{k}|_1 - l \end{pmatrix} \cdot (Q_{k_1}^1 \otimes \ldots \otimes Q_{k_d}^1) f,$$

which is also known as combination technique [25]. Of further importance are the dimension recursive versions [41, 57]

$$Q_l^d f = \sum_{k=1}^{l-1} (\Delta_k^1 \otimes Q_{l-k}^{d-1}) f,$$

and

$$Q_{l+1}^{d+1} f = \sum_{|\mathbf{k}|_1 \leq l+d-1} (\Delta_{k_1}^1 \otimes \ldots \otimes \Delta_{k_d}^1 \otimes Q_{l+d-|\mathbf{k}|_1}^1) f.$$

These two formulations are frequently used to prove the properties of Smolyak's construction presented in the following sections.

4.2 Sparse grids

In a straightforward implementation, Smolyak's construction as well as the combination technique require multiple function evaluations at some abscissas. If function evaluations are costly, e.g. themselves integrals (see for example Section 6.5), it is necessary to modify the algorithm. For the case that the one–dimensional quadrature formulas $Q_l^1 f$ are nested, we consider the one–dimensional difference grids

$$\Theta_l^1 := \Gamma_l^1 \setminus \Gamma_{l-1}^1,$$

with Γ_l^1 being the grids of the one–dimensional formulas and $\Gamma_0^1 := \emptyset$. We denote the elements in Θ_l^1 by y_{li}, $1 \leq i \leq m_l$. In the non–nested case, we set

$$\Theta_l^1 := \Gamma_l^1,$$

$y_{li} := x_{li}$ and $m_l = n_l$. The points of the multivariate Smolyak formula then form a so–called sparse grid [58] given by the union over the pairwise disjoint grids $\Theta_{k_1}^1 \times \ldots \times \Theta_{k_d}^1$,

$$\Gamma_l^d = \biguplus_{|\mathbf{k}|_1 \leq l+d-1} \Theta_{k_1}^1 \times \ldots \times \Theta_{k_d}^1.$$

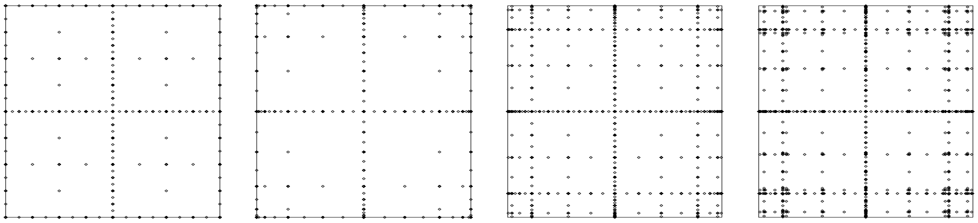

Figure 1: Sparse grids corresponding to the trapezoidal, Clenshaw–Curtis, Gauss–Patterson, and Gauss–Legendre rules for $d = 2, l = 6$.

Examples for sparse grids are shown in Figure 1. We have also plotted a sparse grid corresponding to the non–nested Gauss–Legendre formula in which the number of abscissas of the one–dimensional formulas is the same as in the Gauss–Patterson case. Note again, that the number of abscissas of the Gauss and Gauss–Patterson formulas is approximately twice the number of abscissas of the trapezoidal and Clenshaw–Curtis formulas.

As an important consequence, the multivariate formulas are nested, if the one–dimensional formulas are [37], that is,

$$\Gamma_l^d \subset \Gamma_{l+1}^d.$$

The number of points in a sparse grid can be determined as

$$n_l^d = \sum_{|\mathbf{k}|_1 \le l+d-1} m_{k_1} \cdot \ldots \cdot m_{k_d}.$$

If $n_l^1 = O(2^l)$ the order of n_l^d is (see [39])

$$n_l^d = O(2^l \cdot l^{d-1}).$$

This property is in contrast to product rules in which the number of grid points is of order $O(2^{ld})$. In order to reduce the increase of the number of grid points in high dimensional problems it is especially important to have $n_1^1 = 1$. Note, that the order of n_l^d is the same in the nested as well as in the non–nested case, but the constants are considerably larger in the non–nested case if the same number of abscissas for the univariate quadrature formulas are used.

4.3 Computation of the weights

Smolyak's algorithm can now be written as

$$Q_l^d f := \sum_{|\mathbf{k}|_1 \le l+d-1} \sum_{j_1=1}^{m_{k_1}} \cdots \sum_{j_d=1}^{m_{k_d}} w_{\mathbf{kj}} \cdot f(\mathbf{x}_{\mathbf{kj}}).$$

with $\mathbf{x}_{\mathbf{kj}} := (x_{k_1 j_1}, \ldots, x_{k_d j_d})$. In the nested case the weights are given by

$$w_{\mathbf{kj}} = \sum_{|\mathbf{k}+\mathbf{q}|_1 \le l+2d-1} v_{(k_1+q_1)j_1} \cdot \ldots \cdot v_{(k_d+q_d)j_d},$$

with $\mathbf{q} \in \mathbb{N}^d$ and

$$v_{(k+q)j} := \begin{cases} w_{kj} & \text{if} \quad q = 1 \\ w_{(k+q-1)r} - w_{(k+q-2)s} & \text{if} \quad q > 1, x_{kj} = x_{(k+q-1)r} = x_{(k+q-2)s}. \end{cases}$$

Explicit formulas for the weights $w_{\mathbf{kj}}$ exist only in special cases, such as the rectangle rule [41]. For the non–nested case, we have directly

$$w_{\mathbf{kj}} = w_{k_1 j_1} \cdot \ldots \cdot w_{k_d j_d}.$$

The weights can be precomputed in both cases, so there is no practical difference concerning the overall cost of the quadrature formulas.

Smolyak quadrature formulas can contain negative weights even if the underlying univariate quadrature formulas are positive. Convergence is guaranteed however, because the absolute values of the weights remain relatively small. It can be shown [38] that as well in the nested as in the non–nested case

$$\sum_{|\mathbf{k}|_1 \leq l+d-1} \sum_{j_1=1}^{m^1_{k_1}} \ldots \sum_{j_d=1}^{m^1_{k_d}} |w_{\mathbf{kj}}| = O((\log(n^d_l))^{d-1}).$$

Nevertheless, due to the existence of negative weights, it is especially important to avoid numerical cancellation. Instead of a summation with increasing l, that is,

$$Q^d_l f := \sum_{m=1}^{l} \sum_{|\mathbf{k}|_1 = m+d-1} \sum_{j_1=1}^{m^1_{k_1}} \ldots \sum_{j_d=1}^{m^1_{k_d}} w_{\mathbf{kj}} \cdot f(\mathbf{x_{kj}}),$$

we recommend to sum up coordinate–wise, i.e.

$$Q^d_l f = \sum_{k_1=1}^{l} \sum_{k_2=1}^{l-k_1} \ldots \sum_{k_d=1}^{l-k_1-\ldots-k_{d-1}} \sum_{j_1=1}^{m^1_{k_1}} \ldots \sum_{j_d=1}^{m^1_{k_d}} w_{\mathbf{kj}} \cdot f(\mathbf{x_{kj}}).$$

Numerical experiments show that rounding errors are much smaller in this case. Similarly, in the nested case the weights $w_{\mathbf{kj}}$ should be computed with the same strategy.

4.4 Error Bounds

We first look at the polynomial degree of exactness of a Smolyak quadrature formula. Let $\mathcal{P}^1_l$ be the space of one–dimensional polynomials of degree $\leq l$. In the multivariate case we consider the polynomial spaces

$$\mathcal{P}^d_l := \{\mathcal{P}^1_{k_1} \otimes \ldots \otimes \mathcal{P}^1_{k_d}, |\mathbf{k}|_1 = l + d - 1\}.$$

If Q^1_l is exact for $\mathcal{P}^1_l$, then Q^d_l is exact for $\mathcal{P}^d_l$ [37], that is

$$E^d_l f = 0 \;\text{ for all }\; f \in \mathcal{P}^d_l.$$

The classical polynomial exactness (based on the spaces $\mathcal{P}_{k_1}^1 \otimes \ldots \otimes \mathcal{P}_{k_d}^1$, $|\mathbf{k}|_\infty = l$), however, behaves for interpolatory quadrature formulas in Smolyak's construction for $l < d$ only linearly like $2l - 1$ [8, 39].

A quadrature formula Q_l^1 is called symmetric, if for $x \in \Gamma_l^1$ also $-x \in \Gamma_l^1$ and the weights for these abscissas are the same. If all Q_l^1 are symmetric, then Q_l^d is exact for all $\mathcal{P}_{k_1} \otimes \ldots \otimes \mathcal{P}_{k_d}$ with at least one k_j odd [37].

In order to formulate error bounds for Smolyak's formula, we start with the error bound for one–dimensional quadrature formulas for functions $f \in \mathcal{C}^r$,

$$|E_l^1 f| = O((n_l^1)^{-r}).$$

This bound holds for example for all interpolatory quadrature formulas with positive weights, such as the Clenshaw–Curtis, Gauss–Patterson and Gauss–Legendre formulas. Taking one such quadrature formula as one–dimensional basis, if $f \in \mathcal{W}_d^r$ and $n_l^1 = O(2^l)$, the error of Smolyak's quadrature formula is of order [57]

$$|E_l^d f| = O(2^{-lr} \cdot l^{(d-1)\cdot(r+1)}).$$

Similar results exist for Korobov spaces E_d^r [2]. For the classical spaces $\mathcal{C}_d^r$ error bounds can also be derived [38], but they indicate an exponential dependence on the dimension.

5 Extensions

In this section, we consider speed–up techniques and strategies for computing singular integrands.

5.1 Extrapolation

Extrapolation methods use a linear combination of nested quadrature formulas to construct a quadrature formula which has lower order error terms eliminated. We will consider here only the extrapolation of the iterated trapezoidal rule.

In the univariate case, the coefficients c_{lj} are determined such that the extrapolated quadrature formula Λ_l^1

$$\Lambda_l^1 f = \sum_{j=1}^{l} c_{lj} \cdot Q_j^1 f.$$

is of higher order. In the case of the iterated trapezoidal rule, the Euler–McLaurin summation formula yields the coefficients

$$c_{lj} = \prod_{\rho=1, \rho \neq j}^{l} \frac{4^\rho}{4^\rho - 4^j}.$$

Since the iterated trapezoidal rule is of order $O(2^{-2l})$, the extrapolated version is of order $O(2^{-2l^2})$ for $f \in \mathcal{C}^{2l}$. Therefore, by replacing r by $2l$ analogous error bounds as for the Gauss and Clenshaw–Curtis formulas can be obtained. In the multivariate case, it is possible to apply the combination technique to get the extrapolated quadrature formula [3]

$$Q_l^d f := \sum_{l \leq |\mathbf{k}|_1 \leq l+d-1} (-1)^{|\mathbf{k}|_1 - 1} \cdot \left(\begin{array}{c} d-1 \\ l - |\mathbf{k}|_\mathbf{1} \end{array} \right) (\Lambda_{k_1}^1 \otimes \ldots \otimes \Lambda_{k_d}^1) f.$$

The error bounds for the extrapolated quadrature formula can be obtained from section 4.4.

5.2 Adaptivity

Adaptive quadrature formulas spend more points in areas where f is not smooth and less points where f is smooth. In principle, it is possible to consider two kinds of adaptivity strategies. The first approach uses a–priori knowledge of the smoothness of the integrand, while in the second approach the algorithm adapts itself automatically during the quadrature process. Examples of such strategies for Smolyak's construction have been presented in [3].

If it is known that the smoothness of the integrand depends on the direction, it is possible to apply different univariate quadrature formulas in each direction. This way, less costly quadrature formulas (even of different type) can be applied in smoother directions and therefore the overall cost is reduced. This is especially important when dealing with high–dimensional integrals. It is also possible to use grid–adapted univariate quadrature formulas, for example if the integrand has singular behaviour in certain directions.

Alternatively, it is possible to modify Smolyak's original construction and define non–isotropic formulas [41] of the form

$$Q_l^d f := \sum_{|\mathbf{k}|_\mathbf{v} \leq l+d-1} (\Delta_{k_1}^1 \otimes \ldots \otimes \Delta_{k_d}^1) f,$$

with

$$|\mathbf{k}|_\mathbf{v} := \sum_{j=1}^{d} v_j \cdot k_j, \;\; v_j > 0.$$

This corresponds to a weighting of the directions and therefore small values for v_j can be used in smoother directions.

This construction is in fact a special case of a wider class of quadrature formulas which can be generated by the following generalization of Smolyak's construction. We denote the j–th unit vector as $\mathbf{e}_j$ and define index sets $\mathcal{I}_l$ such that for all $\mathbf{k} \in \mathcal{I}_l$,

$$\mathbf{k} - \mathbf{e}_j \in \mathcal{I}_l \;\; \text{for} \;\; 1 \leq j \leq d, \;\; k_j > 1,$$

holds. This way, a general Smolyak quadrature formula is defined by

$$Q_l^d f := \sum_{\mathbf{k} \in \mathcal{I}_l} (\Delta_{k_1}^1 \otimes \ldots \otimes \Delta_{k_d}^1) f.$$

To describe the general combination technique formally, we define the characteristic function $\chi_{\mathcal{I}_l}$ of $\mathcal{I}_l$ as

$$\chi_{\mathcal{I}_l}(\mathbf{k}) = \begin{cases} 1 & \text{if } \mathbf{k} \in \mathcal{I}_l, \\ 0 & \text{else}, \end{cases}$$

and have

$$Q_l^d f = \sum_{\mathbf{k} \in \mathcal{I}_l} \left(\sum_{z_1=0}^{1} \ldots \sum_{z_d=0}^{1} (-1)^{|\mathbf{z}|_1} \cdot \chi_{\mathcal{I}_l}(\mathbf{k}+\mathbf{z}) \right) (Q_{k_1}^1 \otimes \ldots \otimes Q_{k_d}^1) f.$$

An appropriate choice of $\mathcal{I}_l$ can reduce the computing cost significantly for special integrands especially in high–dimensional problems, as can be seen in Section 6.6. The index sets $\mathcal{I}_l$ can be obtained a–priori or by a self–adaptive procedure.

Adaptive schemes during the quadrature process are based on an error estimator which refines regions according to a partitioning scheme [18, 55]. Nested quadrature formulas allow a simple global error estimator based on the difference of two subsequent quadrature formulas, i.e. for $l \geq 2$,

$$|E_l^d f| \approx |(Q_l^d - Q_{l-1}^d) f|.$$

Refinement strategies can then be obtained by locally applying error estimators of above form. However note that self–adaptive strategies for high–dimensional integrals are very costly and involved.

5.3 General Weight Functions

We consider the weighted integral

$$I^d f := \int_\Omega w(\mathbf{x}) f(\mathbf{x}) \; d\mathbf{x},$$

where

$$w(\mathbf{x}) := w_1(x_1) \cdot \ldots \cdot w_d(x_d).$$

In this case, Smolyak's construction can then be applied directly using appropriate one–dimensional quadrature formulas corresponding to the weight functions $w_1, \ldots, w_d$.

For general weight functions, the theory of nested univariate quadrature formulas is far less developed. Patterson [44] constructs extended Gaussian quadrature formulas for general constant signed weight functions. The computation of the abscissas is done on the basis of the associated 3–term recurrence relation for the orthogonal polynomials corresponding to the weight function.

However, the existence of real abscissas and the interlacing property is not always guaranteed, and is heavily dependent on the weight function. For Hermite and Laguerre weight functions, even the first (Kronrod) extension is known only for a few special cases. On the other hand, extensions for Gauss–Chebyshev formulas (of the first and second kind) seem to exist for all n and even have the same or nearly the same degree of exactness as the corresponding Gauss formulas.

One way for the derivation of extensions in the cases where the existence of Patterson extensions is not guaranteed are suboptimal extensions [45, 49]. These formulas aim at a smaller degree of exactness than the highest possible but try to construct quadrature formulas with real and interlacing abscissas.

6 Numerical Examples

In the following, we consider several multivariate integration problems arising in various applications which can be solved using the sparse grid method. The dimension of these integrals varies from low (3-6) up to theoretically infinite. In order to show the performance of the methods presented, we chose in all cases except for the CMO problem examples where the exact value of the integral is known. Several examples are taken from Morokoff and Caflisch [34] which allows the comparison with Quasi–Monte Carlo methods.

6.1 Test Functions

First, we consider some test integrals which have a rather simple structure. Although these functions might not be of great use in real applications, they still allow some insight in the properties of sparse grid quadrature formulas. Previous comparisons with other multivariate quadrature formulas [3, 37, 38, 39] showed the superiority of sparse grid quadrature formulas for smooth functions. So here we compare only the performance of sparse grid formulas with different univariate bases. Let us consider the following test integral taken from [34], which is an example for a function whose variation grows exponentially,

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

	Trapez		Clenshaw		Patterson		Gauss	
l	Fcalls	Error	Fcalls	Error	Fcalls	Error	Fcalls	Error
1	1	2.44e-01	1	2.44e-01	1	2.44e-01	1	2.44e-01
2	11	1.08e-00	11	6.38e-01	11	8.94e-03	11	8.94e-03
3	61	7.58e-02	61	1.44e-01	71	8.07e-04	81	8.38e-04
4	241	2.86e-01	231	1.24e-01	351	2.07e-04	471	8.74e-05
5	801	1.08e-01	801	6.65e-03	1471	2.26e-05	2341	7.57e-06
6	2433	8.00e-02	2433	1.06e-02	5503	1.42e-06	10363	9.38e-08
7	6993	5.03e-02	6993	1.74e-03	18943	3.44e-09	41913	1.94e-07
α	0.22		0.61		1.63		1.54	

Table 1: Computational results for the test integral

The exact value of this integral is 1. In Table 1 we compare the numerical results for Smolyak quadrature formulas with the trapezoidal rule (*trapez*), the Clenshaw–Curtis formulas (*clenshaw*), the Gauss–Patterson formulas (*patterson*), and the Gauss–Legendre (*gauss*) formulas as univariate basis integration routines for dimension $d = 5$. At the bottom of this and the following tables we denote the computed exponential factor α for the error

$$E_l^d f = c \cdot N^{-\alpha}$$

obtained by a least squares fit of the data.

We see that the Patterson formula performs best considering the ratio of error to function calls. The fitted convergence rate for a Quasi–Monte–Carlo method using a Sobol sequence is for this example only $\alpha \approx 0.5$ [34]. The Clenshaw–Curtis and trapezoidal rules perform particularly bad probably because they require function evaluations in the origin where the singularity is located.

Further tests of Smolyak's construction based on various univariate quadrature formulas using the Genz testing package have been made in [3, 8, 37]. In summary, the results show that the Filippi formulas perform in general worse than the Patterson formulas and the trapezoidal and extrapolated trapezoidal rules usually perform worst. The superiority of Patterson formulas over Clenshaw–Curtis formulas in Smolyak's construction decreases with rising dimension d. This can be accounted to the fact that the Clenshaw–Curtis formula requires less function evaluations for the same classical polynomial exactness for $l < d$.

6.2 Integral Equation

The discretization of multivariate partial differential or integral equations using finite element or boundary element methods often requires the evaluation of multivariate integrals. In case the exact value of these integrals is too expensive to compute, numerical algorithms are commonly used. We consider the Dirichlet screen (square plate) problem

$$\Delta u = 0 \quad \text{in} \quad \mathbb{R}^3 \setminus \bar{\Omega}$$

$$u = f \quad \text{on} \quad \Omega$$

with $\Omega = [0,1]^2$ and $u = O(\frac{1}{|\mathbf{z}|})$ for $|\mathbf{z}| \to \infty$. This problem can be transformed into an integral equation using the single layer potential V, i.e.

$$V\Psi(\mathbf{x}) := \frac{1}{4\pi} \int_\Omega \frac{1}{|\mathbf{x} - \mathbf{y}|} \Psi(\mathbf{y}) \; ds_{\mathbf{y}} = f(\mathbf{x}), \;\; \mathbf{x} \in \Omega.$$

The boundary element Galerkin approach using a finite–dimensional approximating subspace X_h of the appropriate Sobolev space leads to the linear system

$$(V\Psi_h, \phi_h) = (f_h, \phi_h)$$

with $\phi_h, \Psi_h \in X_h$. We consider for X_h the space of piecewise bilinear functions on an equidistant grid Ω_h with $h = 2^{-l}$ and use a nodal (Lagrange) basis $\{\phi_h\}$. The entries of the $n^2 \times n^2, n = 2^l + 1$ matrix $A = (a_{\mathbf{b},\mathbf{c}}) := (V\Psi_h, \phi_h), \mathbf{b}, \mathbf{c} = (0,0), \ldots, (n,n)$ are then given by

$$a_{\mathbf{b},\mathbf{c}} = \int_{(b_1-1)h}^{(b_1+1)h} \int_{(b_2-1)h}^{(b_2+1)h} \int_{(c_1-1)h}^{(c_1+1)h} \int_{(c_2-1)h}^{(c_2+1)h} \frac{\phi_h^{\mathbf{b}}(\mathbf{x}) \phi_h^{\mathbf{c}}(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|} \; d\mathbf{y} \; d\mathbf{x},$$

where

$$\phi_h^{\mathbf{b}}(\mathbf{x}) = \begin{cases} \varphi_h(\mathbf{x} - h\mathbf{b}) & \text{for } \mathbf{x} - h\mathbf{b} \in \Omega \\ 0 & \text{else} \end{cases}$$

and

$$\varphi_h(\mathbf{x}) = \max\{(1 - |x_1|/h) \cdot (1 - |x_2|/h), 0\}.$$

The exact value of these integrals can be computed using the computationally rather expensive recursion formulas presented in [29]. For the numerical computation each integral is subdivided along $\mathbf{x} = \mathbf{b}$ and $\mathbf{y} = \mathbf{c}$ into the 16 parts where the integrand is differentiable. Away from the singularity it is possible to apply Smolyak's construction directly. For the computation of the singular integrals the above mentioned recursion formulas or the Duffy transformation [12] can be used.

In Table 2 we compare the performance of several Smolyak quadrature formulas using different one–dimensional basis integration routines for one selected matrix entry. We show the computational results for $h = \frac{1}{32}$, $\mathbf{a} = (0,0)^T$ and $\mathbf{b} = (0,3)^T$ since this is one of the least smooth non–singular integrals.

	Trapez		Clenshaw		Patterson		Gauss	
l	Fcalls	Error	Fcalls	Error	Fcalls	Error	Fcalls	Error
1	1	7.28e-03	1	7.28e-03	1	7.28e-03	1	7.28e-03
2	9	4.71e-03	9	7.15e-04	9	2.00e-04	9	2.00e-04
3	41	1.51e-03	41	2.41e-06	49	4.63e-05	57	4.63e-05
4	137	4.80e-04	137	1.67e-05	209	3.85e-06	289	3.85e-06
5	401	5.43e-05	401	4.42e-06	769	2.90e-07	1265	2.90e-07
6	1105	2.30e-05	1105	5.57e-07	2561	8.52e-09	4969	8.52e-09
7	2929	6.00e-06	2929	5.18e-07	7973	8.55e-11	17945	8.55e-11
α	0.94		1.21		1.97		1.80	

Table 2: Computational results for the integral equation

We see that the errors are the same for the Gauss–Legendre and Gauss–Patterson formulas. In fact they are not identical but differ in the fourth and following digits. This can be accounted to the fact that polynomials in the range for which the Gauss–Legendre formula is exact and the Gauss–Patterson formula is not are integrated by the Gauss–Patterson formula with about this accuracy. However, the Gauss–Patterson formulas require much less function evaluations and therefore have the best rate of convergence. The trapezoidal and Clenshaw–Curtis rules require still less function evaluations but also exhibit a much larger error.

Once all matrix entries have been computed the linear system can be solved using a preconditioned conjugate gradient iteration. This yields the solution shown in Figure 2.

Further integral equations arise in computer graphics, examples are the radiosity and volume rendering equations. There are also higher dimensional generalizations, for example volume integral equations, which require the evaluation of six dimensional integrals.

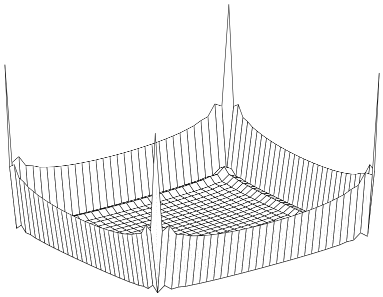

Figure 2: Numerically computed screen ($h = \frac{1}{32}$)

6.3 Boltzmann Collision Integral

The gain in the distribution function for the density of rarefied gas can be described by an integral over velocity $\mathbf{w}$ and the two collision angles χ and ε. We consider here a special case, where the exact value of the integral is known [34]. We consider the five–dimensional integral

$$I^d f = \frac{1}{\pi^3} \int_{\mathbb{R}^3} \int_0^{2\pi} \int_0^{\pi} |\mathbf{w}|_2 \cdot \sin\chi \cdot e^{h(u,\mathbf{w},\chi,\varepsilon)} \cdot e^{-|\mathbf{w}|_2^2} \, d\chi d\varepsilon d\mathbf{w},$$

where $w_{23} = \sqrt{w_2^2 + w_3^2}$ and

$$\begin{aligned} h(u,\mathbf{w},\chi,\varepsilon) &= -2u^2 + u(w_1 + w_2 + (w_2 - w_1)\cos\chi) \\ &\quad + u \cdot \sin\chi \left(w_{23} \cdot \sin\varepsilon + \frac{|\mathbf{w}|_2 w_3 \cos\varepsilon + w_1 w_2 \sin\varepsilon}{w_{23}} \right). \end{aligned}$$

The exact value of this integral is given by the series

$$I^d f = \frac{4}{\pi} \cdot e^{-2u^2} \left(\sum_{k=0}^{\infty} c_k \cdot u^{2k} \right),$$

with

$$c_k = \frac{3^{k+1} - 1}{1 \cdot 3 \cdot \ldots \cdot (2k+1)}.$$

The integral over $\mathbb{R}^3$ can be transformed into an integral over the unit cube using the inverse distribution function thus eliminating the $e^{-|\mathbf{w}|_2^2}$ term. The computational results for $u = 0.25$ are shown in Table 3.

	Patterson		Gauss	
l	Fcalls	Error	Fcalls	Error
1	1	1.00e+00	1	1.00e+00
2	11	9.74e-01	11	9.74e-01
3	71	2.09e-01	81	2.10e-01
4	351	9.28e-03	471	9.29e-03
5	1471	8.70e-05	2341	8.69e-05
α	1.23		1.16	

Table 3: Computational results for the Boltzmann collision integral

The trapezoidal and the Clenshaw–Curtis formulas cannot be used in this case since they require function evaluations on the boundary of the interval. Of course, there are transformations which can overcome this drawback, but they usually change the smoothness class of the integrand. In case of the Clenshaw–Curtis formula it is possible to use Chebyshev knots of the second kind for the appropriate weight function.

Again, the Patterson and Gauss formulas show similar error behaviour while the Patterson formulas are computationally much cheaper. The fitted convergence rates α are relatively small in both cases, but about twice as high as those of Quasi–Monte Carlo methods which lie in the range of 0.61–0.65 for this example [34].

6.4 Absorption Problem

We consider the numerical computation of a transport problem given by the integral equation

$$y(x) = x + \int_x^1 \gamma \cdot y(z)\ dz,$$

which describes the behaviour of a particle traveling through a one–dimensional slab of unit length. In each step, the particle travels forward a random distance uniform between 0 and 1. If it does not leave the slab this way, it may be absorbed with probability $1 - \gamma$. The solution of this problem is given by

$$y(x) = \frac{1}{\gamma} - \frac{1-\gamma}{\gamma} \cdot e^{\gamma(1-x)}.$$

This solution can also be represented by the infinite–dimensional integral

$$y(x) = \int_{[0,1]^\infty} \sum_{n=0}^{\infty} F_n(x, \mathbf{z})\ d\mathbf{z},$$

with

$$F_n(x, \mathbf{z}) = \gamma^n \cdot \theta\left((1-x) - \sum_{j=1}^{n} z_j\right) \cdot \theta\left(\sum_{j=1}^{n+1} z_j - (1-x)\right),$$

where θ is the Heaviside function

$$\theta(s) = \left\{ \begin{array}{ll} 1 & \text{for } s \geq 0 \\ 0 & \text{for } s < 0 \end{array} \right. .$$

Since it is rather unlikely that the particle can travel more than a few steps before being absorbed or leaving the slab, the infinite–dimensional integral may be approximated by truncation at a finite dimension d, i.e.

$$\tilde{y}(x) = \int_{[0,1]^d} \sum_{n=0}^{d-1} F_n(x, \mathbf{z})\ d\mathbf{z}.$$

However, in this formulation the integrand is discontinuous, which has a negative effect on the performance of sparse grid quadrature formulas. Alternatively, the same solution can be obtained by replacing F_n by F'_n in the infinite–dimensional integral where

$$F'_n(x, \mathbf{z}) = \gamma^n (1-x)^n \left(\prod_{j=1}^{n-1} (z_j)^{n-j} \right) \cdot \left(1 - (1-x) \cdot \prod_{j=1}^{n} z_j \right).$$

Afterwards, an analogous truncation at finite dimension d can be performed. In this formulation, the integrand is smooth. The numerical results for $\tilde{y}(0)$ and $d = 8$ are shown in Table 4.

	Trapez		Clenshaw		Patterson		Gauss	
l	Fcalls	Error	Fcalls	Error	Fcalls	Error	Fcalls	Error
1	1	2.02e-02	1	2.02e-02	1	2.02e-02	1	2.02e-02
2	17	8.12e-03	17	1.32e-03	17	1.33e-03	17	1.33e-03
3	145	3.86e-03	145	7.18e-05	161	9.19e-05	177	9.19e-05
4	849	9.74e-04	849	2.68e-06	1121	6.13e-06	1409	6.13e-06
5	3937	1.37e-04	3937	1.15e-06	6401	3.82e-07	9377	3.82e-07
6	15713	7.36e-06	15713	9.70e-08	31745	1.40e-08	54673	1.40e-08
α	0.76		1.28		1.35		1.29	

Table 4: Computational results for the absorption problem

The errors for the Gauss and Patterson formulas are again nearly the same. However, in this example the Clenshaw–Curtis formulas perform almost as good as the Gauss formulas. As already mentioned this can be accounted to the dependence of the classical polynomial exactness on the dimension.

6.5 Path Integral

We consider the initial value problem given by the linear parabolic differential equation

$$\frac{\partial u}{\partial t} = \frac{1}{2} \cdot \frac{\partial^2 u}{\partial x^2}(x,t) + v(x,t) \cdot u(x,t),$$

with initial condition $u(x, 0) = f(x)$. The solution of this problem is given by the Feynman–Kac formula [30] as

$$u(x,t) = E_{x,0} \left(f(\xi(t)) \cdot e^{\int_0^t v(\xi(r), t-r)\, dr} \right),$$

where ξ represents a Wiener path starting at $\xi(0) = x$. The expectation $E_{x,0}$ can be approximated by discretizing time in a finite number of time steps $t_i = i \cdot \Delta t$ and by approximating the integral in the exponent by a one–dimensional quadrature formula such as the trapezoidal rule for each time step. We consider the example

$$v(x,t) = \left(\frac{1}{t+1} + \frac{1}{x^2+1} - \frac{4x^2}{(x^2+1)^2} \right),$$

with initial condition $u(x,0) = \frac{1}{x^2+1}$. The exact solution of this example is then

$$u(x,t) = \frac{t+1}{x^2+1}.$$

The results for $\Delta t = 0.02$ and $x = 0$ are given in Table 5. Increasing the level and dimension separately improves the accuracy only to a certain extent, see Figure 3. This is another type of the curse of dimension we encounter here. In order to gain accuracy, it is necessary to increase both the level and the dimension of the quadrature formula. For $d = 10$ and $l = 0 \dots 5$ the fitted rate of convergence is about $\alpha = 0.93$. This rate can be improved by using a Brownian bridge discretization in the dimension, which is explained in the next example.

	$d = 4$		$d = 6$		$d = 8$		$d = 10$	
l	Fcalls	Error	Fcalls	Error	Fcalls	Error	Fcalls	Error
1	1	2.06e-02	1	2.06e-02	1	2.06e-02	1	2.06e-02
2	9	2.90e-03	13	2.86e-03	17	2.83e-03	21	2.82e-03
3	49	3.41e-04	97	3.29e-04	161	3.24e-04	241	3.21e-04
4	209	3.50e-05	545	3.22e-05	1121	3.11e-05	2001	3.05e-05
5	769	5.33e-06	2561	3.80e-06	6401	3.26e-06	13441	3.00e-06
6	2561	2.81e-06	10625	1.38e-06	31745	8.82e-07	77505	6.50e-07

Table 5: Computational results for the path integral

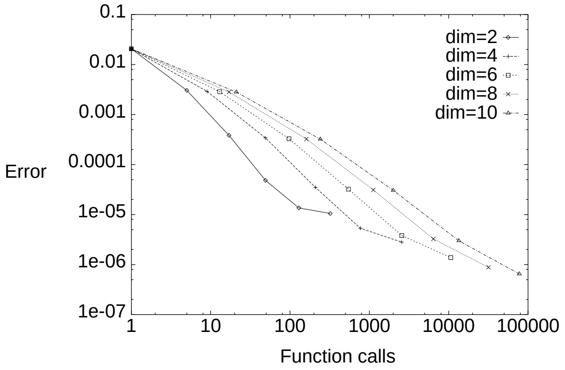

Figure 3: Graphical representation of the results in Table 5

6.6 CMO Problem

A typical collateralized mortgage obligation problem consists of several tranches which derive their cash flows from an underlying pool of mortgages [6, 42]. The problem is to estimate the expected value of the sum of present values of future cash flows for each of the tranches. If the pool of mortgages has a thirty–year maturity and cash flows are obtained monthly, this results in the evaluation of an integral which is of dimension $d = 360$ for each tranche, i.e.

$$PV := \int_{\mathbb{R}^d} v(\xi_1, \dots, \xi_d) \cdot g(\xi_1) \cdot \dots \cdot g(\xi_d) \quad d\xi_d \dots d\xi_1$$

with Gaussian weights $g(\xi_i) = (2\pi\sigma^2)^{-1/2} e^{-\xi_i^2/2\sigma^2}$. The present value v is defined by

$$v(\xi_1, \ldots, \xi_d) := \sum_{k=1}^{d} u_k m_k$$

with

$$\begin{aligned}
u_k &:= \prod_{j=0}^{k-1} (1 + i_j)^{-1}, \\
m_k &:= c r_k ((1 - w_k) + w_k c_k), \\
r_k &:= \prod_{j=1}^{k-1} (1 - w_j), \\
c_k &:= \sum_{j=0}^{d-k} (1 + i_0)^{-j}, \\
i_k &:= K_0^k e^{\xi_1 + \ldots + \xi_k} i_0, \\
w_k &:= K_1 + K_2 \arctan(K_3 i_k + K_4).
\end{aligned}$$

The variables u_k, m_k, and i_k are the discount factor, the cash flow and the interest rate for month k, respectively. The constant $K_0 := e^{-\sigma^2/2}$ is chosen to normalize the log–normal distribution, i.e. $E(i_k) = i_0$. The initial interest rate i_0, the monthly payment c, and K_1, K_2, K_3, K_4 are further constants of the model which are usually set to

$$(i_0, c, K_1, K_2, K_3, K_4, \sigma) := (0.007, 1.0, 0.01, -0.005, 10, 0.5, 0.0004)$$

For the numerical computation, the integral over $\mathbb{R}^d$ is transformed to an unweighted integral on $[0,1]^d$ by a mapping $\xi_i = G(x_i)$ with $G'(x_i) = g(\xi_i)$ which takes a uniform distributed variable x_i to an $N(0, \sigma)$ distributed variable ξ_i,

$$PV = \int_{[0,1]^d} v(G(x_1), \ldots G(x_d)) \; dx_d \ldots dx_1.$$

The most natural way of determining the interest rate fluctuations is by a random walk, i.e. using the recursive formula

$$i_k = K_0 e^{\xi_k} i_{k-1}.$$

In the Brownian bridge discretization [6], however, the interest rate is determined from a future and a past value

$$i_k = K_0^k e^{b(k)} i_0 := K_0^k e^{\frac{1}{2}(\xi_{k+t} + \xi_{k-t}) + \sqrt{t/2}\xi_k} i_0.$$

So starting with $b(0) := 0$, $b(d) := \sqrt{d} \cdot \xi_d$ the subsequent values to be computed are $b(d/2), b(d/4), b(3d/4), b(d/8), b(3d/8), \ldots$ and so forth. This leads to a concentration of the total variance in the first steps of the discretization which improves the rate of convergence of Quasi–Monte Carlo methods.

For sparse grid quadrature formulas there is no immediate advantage from the Brownian bridge discretization since all dimensions are of equal importance. However, it is possible to use adaptive strategies (see Section 5.2), which apply lower degree quadrature formulas in less important dimensions.

In order to detect the most important dimensions we proceed as follows. First we compute the integral using a Smolyak quadrature formula of level l. Then we look at the change in the integral value when in Smolyak's construction a quadrature formula of level $l+1$ is applied in the s-th dimension and quadrature formulas of level l in all other dimensions. Figure 4 (*left*) shows these changes of the integral value with $l = 1$ for a 256–dimensional problem where the dimensions are sorted in the order of the Brownian Bridge construction. We have used a problem for $21\frac{1}{3}$ years here since this simplifies the Brownian bridge discretization. The construction of course can be generalized to arbitrary dimensions.

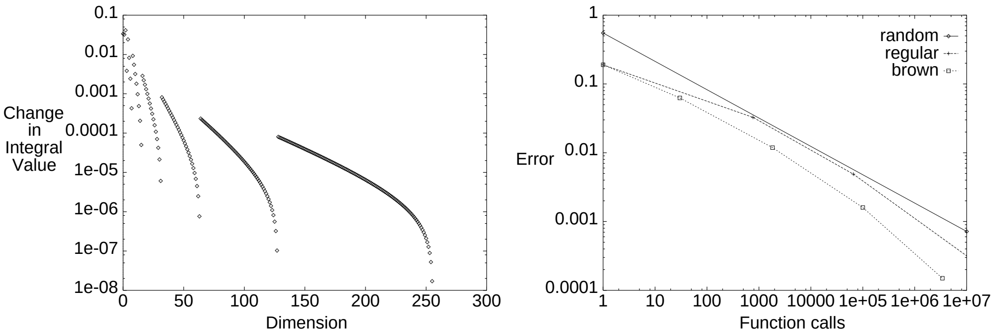

Figure 4: Importance of the dimensions in the Brownian bridge representation (*left*) and function calls vs. error for the CMO problem (*right*).

Figure 4 (*right*) shows the performance of Smolyak's construction using Gauss–Patterson formulas with and without the Brownian bridge discretization using quadrature formulas of level $l = 1 \ldots 5$ in the 30 most important dimensions (in the sense of Figure 4 (*left*)) and of level $l - 1$ in all other dimensions. With this construction it is possible to win about one digit of accuracy in comparison to the average error of a Monte Carlo method. The performance is comparable to Quasi–Monte Carlo methods [6], but more elaborate adaptive schemes can possibly further improve the results.

7 Concluding Remarks

We have shown various constructions for multivariate quadrature formulas on sparse grids based on Newton–Cotes, Clenshaw–Curtis, Gauss and extended Gauss formulas. We stated known results concerning computing cost and error bounds and indicated a numerically stable implementation. We presented a generalization of Smolyak's construction which can take into account the smoothness properties of the integrand varying with the dimension.

We have seen that sparse grid quadrature formulas apply very well to many application problems which require the evaluation of multivariate integrals.

In most cases, the Patterson formulas perform best in comparison to Gauss–Legendre, Clenshaw–Curtis or trapezoidal formulas considering the ratio of necessary function evaluations and accuracy. Except for very high–dimensional problems they outperform Quasi–Monte Carlo methods by about a factor of 2 in the fitted convergence rate α. Quasi–Monte Carlo methods in turn show much better results than Monte Carlo methods.

From the error bounds in Section 4.4 one can expect an asymptotic exponential rate of convergence for smooth functions. This cannot be expected for Monte Carlo and Quasi–Monte Carlo methods since they cannot take into account the smoothness properties of the integrand. In our experiments we see the expected asymptotic exponential convergence in the downward slope of the error graphs, although the least squares fitted convergence rates α cannot reflect it. Furthermore, sparse grid quadrature formulas perform much better than other multivariate quadrature formulas already in the pre–asymptotic range, which is important for the practical computation of high–dimensional problems. This superiority of sparse grid quadrature formulas increases further if higher accuracies are required.

Note finally that for non–smooth integrands sparse grid quadrature formulas perform rather poor. Quasi–Monte Carlo methods obtain for these problems often better results with respect to the ratio of cost vs. accuracy since their smoothness requirements are lower in comparison to Smolyak's construction.

The comparison of the various univariate basis integration routines has shown that nested quadrature formulas are the best choice for Smolyak's construction. Of these, the Patterson formulas perform better than Clenshaw–Curtis formulas in our computational examples. However, this superiority degrades with the dimension due to the identical (classical) polynomial exactness of the formulas for $l < d$.

Especially for high–dimensional problems a careful adaption of the quadrature formula to the smoothness of the integrand or is required in order to limit the increase of the computational cost with the level l. For the path integral and the CMO problem one possibility is the Brownian bridge construction [6], which reduces the effective dimension of the problem. Further acceleration can be achieved by a (coarse grain) parallelization [24] of the method based on the grids $\Theta^1_{k_1} \times \ldots \times \Theta^1_{k_d}$ which is subject to future research.

References

[1] A.R. Barron: Approximation and estimation bounds for artificial neural networks, *Machine Learning* 14, pp. 115–133, 1994.

[2] G. Baszenski, F.–J. Delvos: Multivariate Boolean midpoint rules, in *Numerical Integration IV*, H. Brass, G. Hämmerlin (eds.), pp. 1–11, Birkhäuser, Basel, 1993.

[3] T. Bonk: A new algorithm for multi–dimensional adaptive numerical quadrature, in *Adaptive Methods – Algorithms, Theory and Applications*, W. Hackbusch, G. Wittum (eds.), pp. 54–68, Vieweg, Braunschweig, 1994.

[4] H. Brass, K.-J. Förster: On the estimation of linear functionals, *Analysis* 7, pp. 237–258, 1987.

[5] H.–J. Bungartz: Dünne Gitter und deren Anwendung bei der adaptiven Lösung der dreidimensionalen Poisson-Gleichung, *Dissertation*, Institut für Informatik, TU München, 1992.

[6] R.E. Caflisch, W.J. Morokoff, A. Owen: Valuation of mortgage backed securities using Brownian bridges to reduce effective dimension *J. Comput. Finance*, to appear, 1998.

[7] C.W. Clenshaw, A.R. Curtis: A method for numerical integration on an automatic computer, *Numerische Mathematik* 2, pp. 197–205, 1960.

[8] R. Cools, B. Maerten: Integration over an hypercube, *Preliminary Report*, Departement Computerwetenschappen, Katholieke Universiteit Leuven, 1998.

[9] P.J. Davis, P. Rabinowitz: Methods of Numerical Integration, Academic Press, New York, 1975.

[10] F.–J. Delvos: d–variate Boolean interpolation, *J. Approx. Theory* 34, pp. 99-114, 1982.

[11] F.–J. Delvos, W. Schempp: Boolean methods in interpolation and approximation, *Pitman Research Notes in Mathematics Series* 230, Longman, Essex, 1989.

[12] M.G. Duffy: Quadrature over a pyramid or cube of integrands with a singularity at a vertex, *SIAM J. Num. Anal.* 19, pp. 1260–1262, 1982.

[13] S. Elhay, J. Kautsky: A method for computing quadratures of the Kronrod–Patterson type, *Australian Computer Science Comm.* 6(1), 15.1–15.8, 1984.

[14] S. Elhay, J. Kautsky: Generalized Kronrod Patterson Type Imbedded Quadratures, *Applications of Mathematics*, 37(2), pp. 81–103, 1992.

[15] K. Frank, S. Heinrich: Computing discrepancies of Smolyak quadrature rules, *J. Complexity* 12, 1996.

[16] J. Garcke: Berechnung der kleinsten Eigenwerte der stationären Schroedingergleichung mit der Kombinationstechnik, Diplomarbeit, Institut für Angewandte Mathematik, Universität Bonn, 1998, to appear.

[17] W.M. Gentleman: Implementing Clenshaw–Curtis Quadrature, *Comm. ACM* 15, pp. 337-346, 1972.

[18] A. Genz, A.A. Malik: An Adaptive Algorithm for Numerical Integration over an N-Dimensional Rectangular Region, *J. Comp. Appl. Math.* 6, pp. 295–302, 1980.

[19] A. Genz: A package for testing multiple integration subroutines, in *Numerical Integration*, P. Keast, G. Fairweather (eds.), pp. 337–340, Kluwer, Dordrecht, 1987.

[20] T. Gerstner: Adaptive Hierarchical Methods for Landscape Representation and Analysis, in *Proc. Workshop on Process Modelling and Landform Evolution*, S. Hergarten (ed.), Springer, 1998, to appear.

[21] G.H. Golub, J. Kautsky: Calculation of Gauss quadratures with multiple free and fixed knots, *Numerische Mathematik* 41, pp. 147–163, 1983.

[22] W.J. Gordon: Blending function methods of bivariate and multivariate interpolation and approximation, *SIAM J. Num. Anal.* 8, pp. 158–177, 1971.

[23] M. Griebel: A parallelizable and vectorizable multi–level algorithm on sparse grids, in *Parallel Algorithms for Partial Differential Equations*, Hackbusch, W. (ed.), Notes on Numerical Fluid Mechanics 31, Vieweg, Braunschweig, 1991.

[24] M. Griebel: The combination technique for the sparse grid solution of PDEs on multiprocessor machines, *Parallel Processing Letters* 2(1), pp. 61–70, 1992.

[25] M. Griebel, M. Schneider, C. Zenger: A combination technique for the solution of sparse grid problems, in *Iterative Methods in Linear Algebra*, R. Bequwens, P. de Groen (eds.), pp. 263–281, Elsevier, North–Holland, 1992.

[26] M. Griebel, P. Oswald, T. Schiekofer: Sparse grids for boundary integral equations, *Numerische Mathematik*, submitted, 1998.

[27] M. Griebel, G. Zumbusch: Adaptive Sparse Grids for Hyperbolic Conservation Laws, in *Proc. of the 7th Int. Conf. on Hyperbolic Problems*, Birkhäuser, Basel, submitted, 1998.

[28] K. Hallatschek: Fouriertransformation auf dünnen Gittern mit hierarchischen Basen, *Numerische Mathematik* 63, pp. 83–97, 1992.

[29] N. Heuer, M. Maischak, E.P. Stephan: The hp–Version of the Boundary Element Method for Screen Problems, Preprint IFAM7, Universität Hannover, 1997.

[30] M. Kac: On some connections between probability theory and differential and integral equations, in *Proceedings 2nd Berkeley Symp. Math. Stat. Prob.*, J. Neyman (ed.), Univ. of California Press, Berkley, 1951.

[31] A.S. Kronrod: Nodes and Weights of Quadrature Formulas, Consultants Bureau, New York, 1965.

[32] H.N. Mhaskar: Neural Networks and Approximation Theory, *Neural Networks* 9, pp. 711–722, 1996.

[33] G. Monegato: Stieltjes polynomials, and related quadrature rules, *SIAM Rev.* 24(2), pp. 137–158, 1982.

[34] W.J. Morokoff, R.E. Caflisch: Quasi–Monte Carlo Integration, *J. Comp. Phys.* 122, pp. 218–230, 1995.

[35] H. Niederreiter: Random Number Generation and Quasi-Monte Carlo Methods, SIAM, Philadelphia, 1992.

[36] E. Novak, K. Ritter: Global optimization using hyperbolic cross points, in *State of the Art in Global Optimization*, C.A. Floudas, P.M. Parlados (eds.), pp. 19–33, Kluwer, Dordrecht, 1996.

[37] E. Novak, K. Ritter: High dimensional integration of smooth functions over cubes, *Numerische Mathematik* 75, pp. 79–97, 1996.

[38] E. Novak, K. Ritter: The curse of dimension and a universal method for numerical integration, in *Multivariate Approximation and Splines*, G. Nürnberger, J.W. Schmidt, G. Walz (eds.), to appear, 1998.

[39] E. Novak, K. Ritter: Simple cubature formulas for d–dimensional integrals with high polynomial exactness and small error, *Report*, Institut für Mathematik, Universität Erlangen–Nürnberg, 1997.

[40] E. Novak, K. Ritter, A. Steinbauer: A Multiscale Method for the Evaluation of Wiener Integrals, *J. Approx. Theory*, submitted, 1998.

[41] S. Paskov: Average case complexity of multivariate integration for smooth functions, *J. Complexity* 9, pp. 291–312, 1993.

[42] S. Paskov, J.F. Traub: Faster valuation of financial derivatives, *J. Portfolio Management* 22, pp. 113–120, 1995.

[43] T.N.L. Patterson: The Optimum Addition of Points to Quadrature Formulae, *Math. Comp.* 22, pp. 847–856, 1968.

[44] T.N.L. Patterson: Algorithm 672: Generation of interpolatory quadrature rules of the highest degree of precision with preassigned nodes for general weight functions, *ACM Trans. Math. Software* 15(2), pp. 137–143, 1989.

[45] T.N.L. Patterson: Modified optimal quadrature extensions, *Numerische Mathematik* 64, pp. 511-520, 1993.

[46] S.V. Pereverzev: An estimate of the complexity of the approximate solution of Fredholm equations of the second kind with differentiable kernels, *Ukr. Math. J.* 41(2), pp. 1225–1227, 1989.

[47] R. Piessens, M. Branders: A Note on the Optimum Addition of Abscissas to Quadrature Formulas of Gauss and Lobatto Type, *Math. Comp.* 28, pp. 135–140, 344–347, 1974.

[48] P. Rabinowitz, S. Elhay, J. Kautsky: Empirical mathematics: the first Patterson extension of Gauss-Kronrod rules, *Int. J. Computer Math.* 36, pp. 119–129, 1990.

[49] I. Robinson, A. Begumisa: Suboptimal Kronrod extension formulae for numerical quadrature, *Numerische Mathematik* 58, pp. 807–818, 1991.

[50] I.H. Sloan, S. Joe: Lattice Methods for Multiple Integration, Oxford University Press, Oxford, 1994.

[51] S.A. Smolyak: Quadrature and interpolation formulas for tensor products of certain classes of functions, *Dokl. Akad. Nauk SSSR* 4, pp. 240–243, 1963.

[52] F. Sprengel: Interpolation und Waveletzerlegung multivariater periodischer Funktionen, *Dissertation*, Institut für Mathematik, Universität Rostock, 1997.

[53] V.N. Temlyakov: Approximation of Periodic Functions, Nova Science Publishers, New York, 1994.

[54] J.F. Traub, G.W. Wasilkowski, H. Woźniakowski, Information–Based Complexity, Academic Press, New York, 1988.

[55] P. Van Dooren, L. De Ridder; An adaptive algorithm for numerical integration over an n–dimensional cube, *J. Comp. Appl. Math.* 2, pp. 207–217, 1976.

[56] G. Wahba: Interpolating surfaces: high order convergence rates and their associated design, with applications to X–ray image reconstruction, *Report*, Dept. of Statistics, University of Wisconsin, Madison, 1978.

[57] G.W. Wasilkowsi, H. Wozniakowski: Explicit cost bounds of algorithms for multivariate tensor product problems, *J. Complexity* 11, pp. 1–56, 1995.

[58] Zenger, C.: Sparse grids, in Hackbusch, W. (ed.): Parallel Algorithms for Partial Differential Equations, *Notes on Numerical Fluid Mechanics 31*, Vieweg, Braunschweig, 1991.