Higher Order Quadrature on Sparse Grids

Hans-Joachim Bungartz[1] and Stefan Dirnstorfer[2]

[1] IPVS, Universität Stuttgart, Universitätsstraße 38, D-70569 Stuttgart, Germany
bungartz@ipvs.uni-stuttgart.de
[2] Nagler & Company, D-92263 Schnaittenbach, Germany
dirnstor@web.de

Abstract. Sparse grids have turned out to be a very efficient discretization scheme that, to some extent, breaks the curse of dimensionality and, therefore, is especially well-suited for higher dimensional scenarios. Besides the classical sparse grid application, the numerical solution of partial differential equations, sparse grids have been used for various topics such as Fourier transform, image compression, numerical quadrature, or data mining, so far. In this paper, we summarize and assess recent results concerning the application of sparse grids to integrate functions of higher dimensionality, the focus being on the explicit and adaptive use of higher order basis polynomials.

1   Introduction

The representation of functions and the discretization of PDE by conventional methods is limited to time-depending $3\,\mathrm{D}$ problems due to storage requirements and computational cost. The reason is the so-called *curse of dimensionality*, saying that the cost to represent a function or to compute an approximation with some prescribed accuracy ε depends exponentially on the problem's dimensionality d. We, hence, encounter complexities of the order $O(\varepsilon^{-\alpha d})$ with $\alpha > 0$ depending on the respective approach (i.e. the polynomial degree of the ansatz functions, e.g.), the function's smoothness, and on the implementation. With sparse grids, we circumvent this problem by considering spaces of functions with bounded mixed derivatives [6], only, which is not a serious restriction due to the possibility of adaptive mesh refinement. Starting from some suitable $1\,\mathrm{D}$ multilevel basis on the unit interval $[0, 1]$, we obtain a multilevel basis for $\Omega := [0, 1]^d$ by a simple tensor product construction. If the resulting hierarchical basis functions are arranged by gathering those of a support of size $2 \cdot \mathbf{h_l}$ for level $\mathbf{l} = (l_1, \ldots l_d)$, $h_{l_j} = 2^{-l_j}$, $j = 1, ..., d$, in a hierarchical subspace $W_{\mathbf{l}}$, we get the multilevel subspace splitting illustrated for $d = 2$ in Fig. 1.

Now, for each of the $W_{\mathbf{l}}$, its cost (in terms of degrees of freedom) and its benefit (in terms of its contribution to the overall interpolant) can be estimated and used as input of a cost-benefit optimisation process selecting the most important subspaces with their respective grid points. This a priori optimisation leads us to grid patterns and corresponding approximation spaces that consist of $O(N \cdot \log(N)^{d-1})$ or even $O(N)$ grid points, only, instead of the $O(N^d)$ of
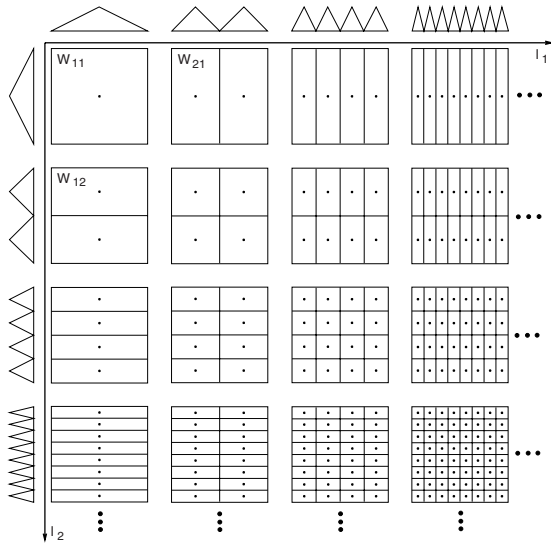
Fig. 1. Scheme of subspaces for $d = 2$: Each square represents one hierarchical subspace $W_{\mathbf{l}}$ with associated grid points and basis functions' support of size $2 \cdot \mathbf{h_l}$.

a regular full grid (N denoting the maximum number of grid points in one direction). These grids correspond to simplizoidal selections of subspaces $W_{\mathbf{l}}$ in Fig. 1, and they are known as *sparse* grids (see Fig. 2). It turns out that the number of degrees of freedom needed for some prescribed accuracy does (up
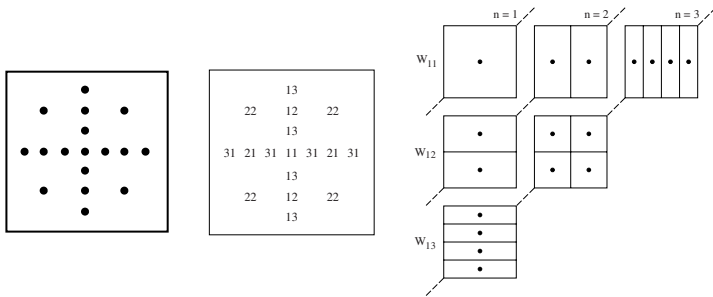

Fig. 2. 2 D regular sparse grid of level 3 and assignment of grid points to subspaces.

to logarithmic factors) no longer depend on d exponentially. This allows either to treat problems of low dimensionality substantially faster or to tackle higher dimensional problems. Extensions of the piecewise linear hierarchical basis to general polynomial degree p as well as interpolets or (pre-)wavelets have been successfully used as 1 D ingredient for the tensor product (see [4,7]).

Since its introduction [17], the sparse grid concept has been applied to most discretization schemes for PDE such as finite elements, finite differences, finite volumes, spectral methods, and splitting extrapolation [8]. In the FE context, special attention has been paid to adaptive mesh refinement [1,4] and to fast solvers [3]. Concerning fields of application, fluid flow was the first focus of sparse grids. Meanwhile, however, sparse grids are also used for problems from quantum mechanics, for problems in the context of stochastic differential equations, or for the discretization of differential forms in the context of the Maxwell equations. Apart from PDE, sparse grids have been and are applied to a variety of other problem classes. Among these problems are integral equations, general operator equations, eigenvalue problems, data mining, and numerical quadrature [10,11, 5], which is the topic of this contribution. For a more detailed introduction to sparse grids, we refer to the extensive survey in [7] and to the references therein.

2   Hierarchical Lagrangian Polynomials

In [4], the piecewise linear hierarchical basis was generalized to hierarchical bases of (piecewise) arbitrary polynomial degree. The main idea of the so-called hierarchical Lagrangian interpolation is to hold to (globally) $\mathcal{C}^0$ functions and elements with one degree of freedom, only, but to define local basis polynomials
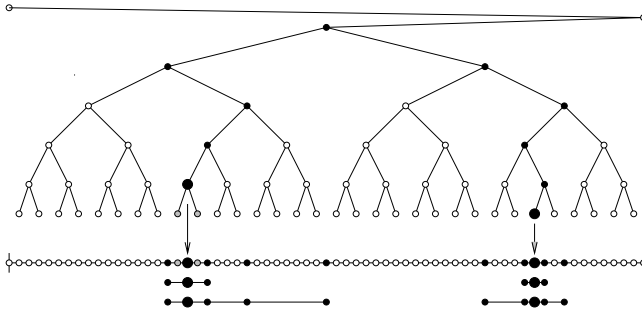

Fig. 3. Ancestors (solid) used for quartic hierarchical Lagrangian interpolation and descendants (dotted) of two grid points.

of higher degree as well as local interpolants by using interpolation conditions in a sufficient number of hierarchical ancestors of the respective grid point *outside* the local element, as illustrated in Fig. 3. That is, we use conditions outside the local support to define the basis polynomials, but we, afterwards, omit the polynomials' parts outside their respective support according to Fig. 1. Depending on the relative position of the ancestor points taken into account, different basis polynomials are obtained. Fig. 4 illustrates this construction principle as well as the resulting four different basis polynomials for the quartic case $p = 4$. Hence, hierarchical Lagrangian interpolation provides a straightforward access
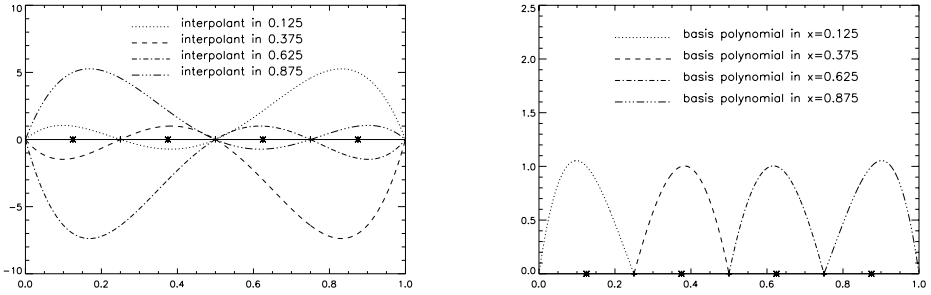
Fig. 4. Basis polynomials for $p = 4$: construction via hierarchical Lagrangian interpolation (left) and used restriction to the respective hierarchical support (right).

to higher order approximation without investing more than one degree of freedom per element. Actually, using such 1 D polynomial bases of degree p in the tensor product leads to accuracies of $O(p)$ [4].

3   Numerical Quadrature on Sparse Grids

Due to the roots of sparse grids going back to Archimedes' quadrature of a parabola and to work of Smolyak [16], quadrature has always been a hot topic in sparse grid research. Starting with the explicit use of the piecewise linear functions to calculate integrals [2], two principal strategies have been pursued. The first approach is based upon the direct application of Smolyak's formula

$$Q_n^{(d)} f \; := \; \left(\sum_{i=0}^{n} \left(Q_i^{(1)} - Q_{i-1}^{(1)} \right) \otimes Q_{n-i}^{(d-1)} \right) f \, , \tag{1}$$

where $Q_n^{(d)}$ denotes a d-dimensional quadrature formula, to standard 1 D rules $Q_n^{(1)}$ such as the Newton-Cotes or Clenshaw-Curtis rules, the Gauss formulas [15], or the incremental Gauss-Patterson rules [10]. Recently, in [11] a generalization of the conventional sparse grid approach which is able to assess the dimensions adaptively according to their importance was developed.

The second approach does not start from 1 D rules as input for Smolyak's formula (1), but, basically, computes a sparse grid interpolant of the integrand and accumulates all occurring basis functions' contributions to the integral. To tackle large values of d and to mimic the increased degree of accuracy of Gaussian quadrature, the Lagrangian basis from the previous section has to be modified. Actually, the basis used in [5] does without boundary points, introducing the grid points' positions as degrees of freedom, themselves (see Fig. 5 for the 1 D case). In the following numerical examples of Sect. 4, the non-equidistant grids and bases (shown in the right part of Fig. 5 for 1 D and resulting from these for larger d) will be used, leading us to the so-called *piecewise Gauss quadrature*.
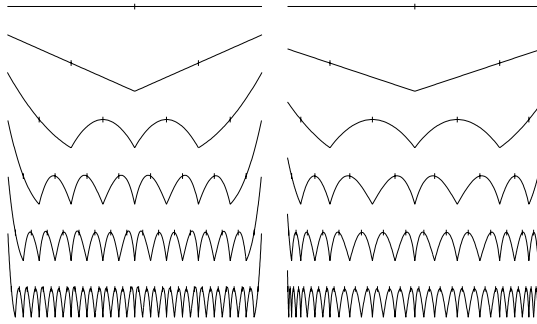
Fig. 5. Hierarchical polynomial bases according to [5]: equidistant (left) and piecewise Gauss version (right), both without boundary points.

A comparison of both approaches reveals the pros and cons of either method. The direct application of (1) allows to use 1 D quadrature rules of arbitrary *global* accuracy, such as the hierarchical Gauss-Kronrod or Gauss-Patterson rules (cf. [10]), whereas the piecewise Gauss quadrature only allows for *locally* higher order, due to the underlying hierarchical Lagrangian approach. On the other hand, in the Smolyak formula, there are no possibilities of a local grid adaptation, which is, in contrast to that, straightforward with piecewise Gauss quadrature. It is not clear how important local mesh refinement may be in the really high dimensional case of, say, $d \gg 20$, where, anyway, only a very small number of steps of refinement is feasible. However, since most of those scenarios are characterized by big differences in the dimensions' importance, some kind of *dimension adaptivity* as studied in [11] and in work in progress [12] in the sequel of [5] will be essential for efficiently tackling such problems.

4   Numerical Results

Next, we apply both sparse gird approaches of Sect. 3 to typical problems of moderate to high d, and we compare the results with standard techniques normally used in these scenarios, such as Monte Carlo quadrature or quasi Monte Carlo quadrature [14], also known as quadrature based on low-discrepancy sequences.

As a first example, we study a simple transport problem from computational physics (see [13] for details), described by the integral equation

$$y(x) \;=\; x + \int_x^1 \gamma y(z) dz \,, \tag{2}$$

for which the exact solution is known. Think of a particle travelling through a 1 D (z) slab of length one. In each step, the particle covers a distance which is uniformly distributed on $[0, 1]$. This may cause it to exit the slab; otherwise, it may be absorbed with probability $1 - \gamma$. The function $y(x)$ denotes the probability for some particle at current position x to leave the slab. The solution of the integral equation (2) can also be represented as an infinite dimensional integral,

$$y(x) \; = \; \int_{[0,1]^\infty} \sum_{n=0}^{\infty} F_n(x, \mathbf{z}) d\mathbf{z} \,, \tag{3}$$

where the vector $\mathbf{z}$ contains the leap lengths and where F_n denotes the probability for the particle to leave the slab after exactly n steps. F_n, basically, is a product of two Heaviside functions, such that the overall integrand has a discontinuity along the diagonal (see Fig. 6), which is kind of a worst case for a regular sparse grid, of course. Concerning dimensionality, since higher dimensions do not contribute considerably to the integral, the sum can be truncated. Fig. 6 (right) shows numerical results for $y(0)$ with $\gamma = 0.5$ in twenty dimensions. The performance of the adaptive sparse grid lies within the range of the standard Monte Carlo method, but can not compete with quasi Monte Carlo. At least, our adaptivity rescues Monte Carlo performance in this worst case scenario.
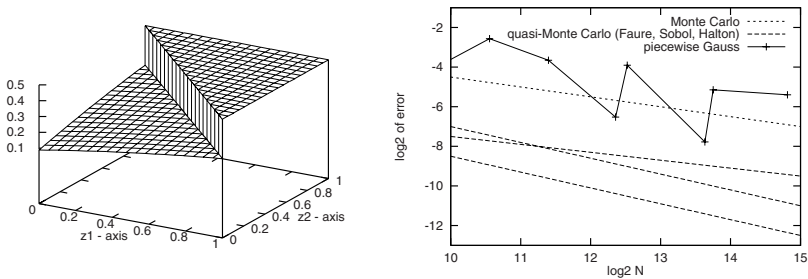

Fig. 6. The discontinuous integrand $\sum_{n=0}^{\infty} F_n(0, \mathbf{z})$ in the first two dimensions (left) and results for numerical quadrature taking into account twenty dimensions according to standard Monte Carlo, quasi-Monte Carlo, and piecewise Gauss quadrature (right).

Much better results can be obtained by changing the formulation once more. $F_n(x, \mathbf{z})$ in (3) can be replaced by a polynomial $F_n^*(x, \mathbf{z})$ describing the contribution of each jump, which, then, results in a smooth integrand. Now, the sparse grid properties can be exploited. Results for $d = 20$ are shown in Fig. 7. While quasi Monte Carlo gains about two digits of accuracy, the sparse grid gains eight. With 30000 evaluated points, the adaptive sparse grid outperforms quasi Monte Carlo by about four digits. Fig. 7 also gives the convergence behaviour if we further reduce dimensionality up to eight. The non-adaptive Gauss-Patterson on sparse grids according to [10] beats quasi Monte Carlo, but not the adaptive sparse grid. Hence, in spite of the smooth integrand, the different importance of the involved directions requires some adaptive strategy.

As a second example, we consider a problem from computational finance, a so-called collateralized mortgage obligation or CMO problem (see [9] for details). There, the present value (PV) of some security is defined as the expectation value over random variables involved in the interest rate fluctuations, each dimension
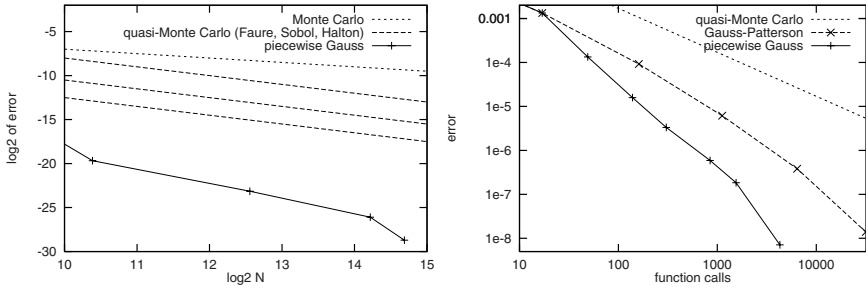
Fig. 7. Results for the smooth integrand in 20 (left) and 8 dimensions (right).

representing one month. After suitable transformations, PV can be written as

$$PV \;=\; \int_{[0,1]^d} v(G(x_1), \cdots, G(x_d))\, dx_1 \cdots dx_d \,, \tag{4}$$

where $d \geq 120$, typically. A more efficient way to compute (4) was suggested in [9], where the authors reduced the effective number of dimensions by representing the interest rate fluctuations as a Brownian motion $b(t)$. Here, a future value $b(t + \Delta t)$ can be generated from the current value $b(t)$ by a random jump:

$$b(t + \Delta t) = b(t) + \sqrt{\Delta t}\, \xi_{t+\Delta t} \,, \tag{5}$$

where ξ_t denotes some random noise. Using the so-called Brownian bridge (BB) discretization, where $b(t + \Delta t)$ is computed from both a past and a future value of b, most of the involved functions' variation can be put into the dimensions defining the first few values of b. In a first scenario with $d = 256$, we compare the adaptive piecewise Gauss and the Gauss-Patterson rule. Fig. 8 (left) shows how the BB formulation improves the performance of adaptive schemes. To make this accessible to Gauss-Patterson, too, a higher order formula is used in the 30 most important directions, found by some setup run up to level two (cf. [10]). The piecewise Gauss rule can find these important directions itself, but shows drawbacks where no use of adaptivity can be made, i.e. at the very beginning of the process and without the BB construction. Although the piecewise Gauss grid performs best for more than 10^6 points, there is no doubt that some dimension adaptivity as suggested in [11] will lead to further improved results. In our second CMO scenario ($d = 360$), we compare our approach with published results for Sobol's quasi Monte Carlo method. The right part of Fig. 8 shows that the adaptive sparse grid can not keep pace for the investigated sample sizes. An increasing downward slope can be expected with even more function evaluations. However, for such large d, the logarithmic factors in the standard (L_2-based) sparse grids' complexity cause problems. A remedy might be the energy-based sparse grids discussed in [6], doing without the logarithmic factors.

Finally, note that aspects of implementation are crucial for a memory- and runtime-efficient maintaining and processing of adaptive sparse grids, see [10,5].
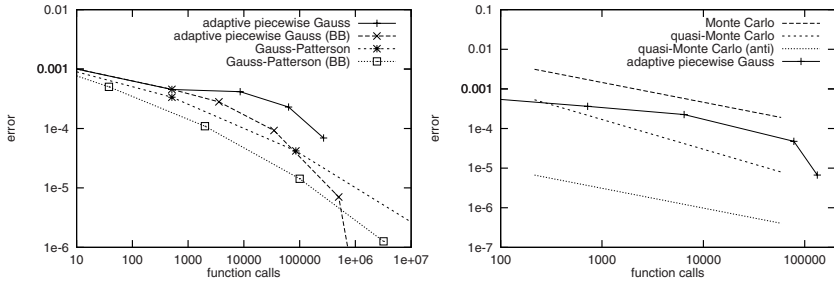
Fig. 8. The CMO problem for $d = 256$ (left), with and without the Brownian bridge, and for $d = 360$ with Brownian bridge (right, comparison with (quasi) Monte Carlo).

References

1. S. Achatz. Higher order sparse grids methods for elliptic partial differential equations with variable contraints. *Computing*, 71(1):1–15, 2003.
2. T. Bonk. A new algorithm for multi-dimensional adaptive numerical quadrature. In W. Hackbusch and G. Wittum, editors, *Adaptive Methods – Algorithms, Theory, and Applications*, volume 46 of *NNFM*, pages 54–68. Vieweg, 1994.
3. H.-J. Bungartz. A multigrid algorithm for higher order finite elements on sparse grids. *ETNA*, 6:63–77, 1997.
4. H.-J. Bungartz. *Finite Elements of Higher Order on Sparse Grids*. Shaker, 1998.
5. H.-J. Bungartz and S. Dirnstorfer. Multivariate quadrature on adaptive sparse grids. *Computing*, 71(1):89–114, 2003.
6. H.-J. Bungartz and M. Griebel. A note on the complexity of solving Poisson's equation for spaces of bounded mixed derivatives. *J. Complex.*, 15:167–199, 1999.
7. H.-J. Bungartz and M. Griebel. Sparse grids. *Acta Numerica*, 2004.
8. H.-J. Bungartz, M. Griebel, and U. Rüde. Extrapolation, combination, and sparse grid techniques for elliptic boundary value problems. *Comput. Meth. Appl. Mech. Eng.*, 116:243–252, 1994.
9. R. Caflisch, W. Morokoff, and A. Owen. Valuation of mortgage backed securities using brownian bridges to reduce effective dimension. *J. Comput. Finance*, 1, 1997.
10. T. Gerstner and M. Griebel. Numerical integration using sparse grids. *Numerical Algorithms*, 18:209–232, 1998.
11. T. Gerstner and M. Griebel. Dimension-adaptive tensor-product quadrature. *Computing*, 71:65–87, 2003.
12. T. Kern. Dünngittertechniken zur hochdimensionalen numerischen Quadratur. Master's thesis, Universität Stuttgart, 2003.
13. W. Morokoff and R. Caflisch. Quasi-monte carlo integration. *J. Comp. Phys.*, 122:218–230, 1995.
14. H. Niederreiter. *Random Number Generation and Quasi-Monte-Carlo Methods*. SIAM, Philadelphia, 1992.
15. E. Novak and K. Ritter. High dimensional integration of smooth functions over cubes. *Numer. Math.*, 75(1):79–98, 1996.
16. S. Smolyak. Quadrature and interpolation formulas for tensor products of certain classes of functions. *Soviet Math. Dokl.*, 4:240–243, 1963.
17. C. Zenger. Sparse grids. In W. Hackbusch, editor, *Parallel Algorithms for Partial Differential Equations*, volume 31 of *NNFM*. Vieweg, 1991.