Applied Numerical Mathematics 62 (2012) 155–165

Contents lists available at SciVerse ScienceDirect

Applied Numerical Mathematics

www.elsevier.com/locate/apnum

Multidimensional pseudo-spectral methods on lattice grids

Hans Munthe-Kaas, Tor Sørevik *

Dept. of Mathematics, University of Bergen, Johannes Brunsgt. 12, Bergen, Norway

ARTICLE INFO

Article history:
Received 17 August 2010
Received in revised form 9 November 2011
Accepted 14 November 2011
Available online 15 November 2011

Keywords:
Multidimensional Fourier expansion
Lattice rules
Pseudo-spectral methods

ABSTRACT

When multidimensional functions are approximated by a truncated Fourier series, the number of terms typically increases exponentially with the dimension s. However, for functions with more structure than just being L^2-integrable, the contributions from many of the N^s terms in the truncated Fourier series may be insignificant. In this paper we suggest a way to reduce the number of terms by omitting the insignificant ones. We then show how lattice rules can be used for approximating the associated Fourier coefficients, allowing a similar reduction in grid points as in expansion terms. We also show that using a lattice grid permits the efficient computation of the Fourier coefficients by the FFT algorithm. Finally we assemble these ideas into a pseudo-spectral algorithm and demonstrate its efficiency on the Poisson equation.

1. Introduction

For 1D pseudo-spectral methods to be efficient, the coefficients in the Fourier expansion need to decay rapidly. In that case we may approximate $f(x)$ accurately by a modest number of modes, those corresponding to frequency or index $|k| \leqslant d$, where the bound d depends on the decay of the Fourier coefficients and the accuracy of the approximation.

In the multidimensional case of periodic functions on the unit cube $[0,1)^s$, k becomes a multi-index, $\mathbf{k} = (k_1, \ldots, k_s) \in Z^s$, and the Fourier expansion is given as

$$f(\mathbf{x}) = \sum_{\mathbf{k} \in Z^s} \widehat{f}(\mathbf{k}) e^{2\pi i \mathbf{k}^T \mathbf{x}}. \tag{1}$$

Again, for efficient approximation by a truncated expansion, we assume that the coefficients decay with increasing $\mathbf{k}$, only this time we need to specify what we mean by *'increasing'*. The obvious answer is to view our multi-index as an integer vector and measure its size by an appropriate norm. Thus for approximation we truncate the sum in (1) according to $\|\mathbf{k}\|_p \leqslant d$. The norm we would like is the one which makes $|\widehat{f}(\mathbf{k})| \approx |\widehat{f}(\mathbf{j})|$ whenever $\|\mathbf{k}\|_p = \|\mathbf{j}\|_p$, which in turn will be highly problem-dependent. For isotropic functions one would expect the 2-norm to be the norm of choice, and as can be seen in Table 1; in that case only moderate savings can be expected. For anisotropic functions the situation may be different. In particular if the function lives in a Korobov space with a norm given by (2) below, we might achieve significant savings. This has been an underlying assumption for much of the work on sparse grids (for an overview see [3] and references therein). Of particular interest we note that H. Yserentant [31,32] has proved that the Hamiltonian operator of the electronic Schrödinger equation in fact does have the smoothness properties required to be in such a space, and consequently attempts to solve this problem on sparse grids have been made [8–10].

* Corresponding author.
E-mail addresses: hans.munthe-kaas@math.uib.no (H. Munthe-Kaas), tor.sorevik@math.uib.no (T. Sørevik).

0168-9274/$36.00
doi:10.1016/j.apnum.2011.11.002

Table 1
The volume of the 4 different topologies for the index spaces, as a function of s and d.

The s-cube	The s-ball	The s-cross-polytope	The s-star
$(2d)^s$	$C_s d^s$	$\frac{(2d)^s}{s!}$	$O(2^s d(\log d)^{s-1})$

The present paper proposes methods based on lattice grids as an alternative to sparse grid methods for such problems. The complexities of these two approaches are similar, whereas FFT based methods on lattice grids are structurally significantly simpler than those based on sparse grids.

A standard way of computing multidimensional Fourier transforms is to simply make a tensor product of the 1D basis functions and truncate in each dimension by retaining, in each dimension i, all $|k_i| \leqslant d$. This corresponds to retaining all coefficients $\mathbf{k}$ such that $\|\mathbf{k}\|_\infty = \max_{i=1,\ldots,s} |k_i| \leqslant d$, even if many of them are infinitesimally small and make no significant contribution to the approximation. The cost for this excessive use of terms is that the number of expansion terms, $N = (2d+1)^s$, increases exponentially with the dimension, and our spectral approximation suffers from *the curse of dimensionality*. The obvious question then becomes: May this curse be, if not avoided, at least mitigated by choosing another norm?

After a brief look at possible savings by switching to a different norm for truncation of the Fourier expansion, we focus on how to compute the associated expansion coefficients. Lattice rules provide a tool for doing so using less than $(2d+1)^s$ function values. For a saving in number of basis functions to materialize as computational saving, we must be able to compute the coefficients by the FFT algorithm as easily as in the standard case. In Section 3 we show the applicability of the FFT algorithm for function sampled on a lattice grid.

We finish by numerical experiments. We solve the Poisson equation in 3 and 4 dimensions, on different grids for three different solutions. For the solution which lies in a Korobov space (defined below), the theory predicts an asymptotic complexity of $O(d(\log d)^{s-1})$. We run a limited scaling experiment up to 7 dimensions, which indicates that this is indeed the case in practice too.

Our experiments reveal, not unexpectedly, that the success of the lattice grid versus standard Cartesian grid is problem-dependent. Cases where our new method performs better than the standard method can, in principle, be predicted upfront if one has knowledge about the decay of the Fourier coefficients.

1.1. The effect of norms

In addition to the standard norms we will also have a look at the α-*norm*. This norm is applicable to functions living in a *Korobov space*, K^α, that is complex valued L^2-functions, periodic on $[0,1)^s$ and with a finite α-*norm*

$$\|u\|_\alpha^2 = \sum_{\mathbf{k}\in Z^s} |\bar{k}_1 \cdot \cdots \cdot \bar{k}_s|^{2\alpha} \left|\widehat{u}(\mathbf{k})\right|^2; \quad \bar{k}_i = \max\bigl(1, |k_i|\bigr). \tag{2}$$

Approximation of functions in Korobov space by truncated Fourier expansion could then be obtained by retaining the terms corresponding to

$$\rho(\mathbf{k}) = |\bar{k}_1 \cdot \cdots \cdot \bar{k}_s| \leqslant d. \tag{3}$$

The corresponding criteria for the ∞-norm, 2-norm and 1-norm would be retaining terms satisfying:

$$\begin{aligned}
\infty\text{-norm:} \quad & \|\mathbf{k}\|_\infty = \max_{i=1,\ldots,s} |k_i| \leqslant d, \\
\text{2-norm:} \quad & \|\mathbf{k}\|_2 = \left(\sum_{i=1}^s k_i^2\right)^{1/2} \leqslant d, \\
\text{1-norm:} \quad & \|\mathbf{k}\|_1 = \sum_{i=1}^s |k_i| \leqslant d.
\end{aligned}$$

In Fig. 1 we illustrate which coefficients to retain in 2 dimensions for $d = 5$. We notice that the geometric shape of this index space takes very different forms and that the number of integer points is quite different. This difference increases with the dimension. For these examples we see that using the infinity norm implies using 121 terms in the expansion, while the corresponding numbers for the 2-norm, 1-norm and α-norm are 81, 61 and 61, respectively.

With the exception of the ∞-norm, no explicit formula exists for the number of integer points for general (s,d). However, approximations to these numbers are provided by the volume of the s-cube (Infinity-norm), the s-ball (2-norm), the s-cross-polytope (1-norm) and the 's-star' shaped geometry defined by (3). These numbers are shown in Table 1. Here

$$C_s = \begin{cases} \frac{\pi^{s/2}}{(s/2)!} & s \text{ even}, \\ \frac{2(2\pi)^{(s-1)/2}}{s!!} & s \text{ odd}. \end{cases} \tag{4}$$

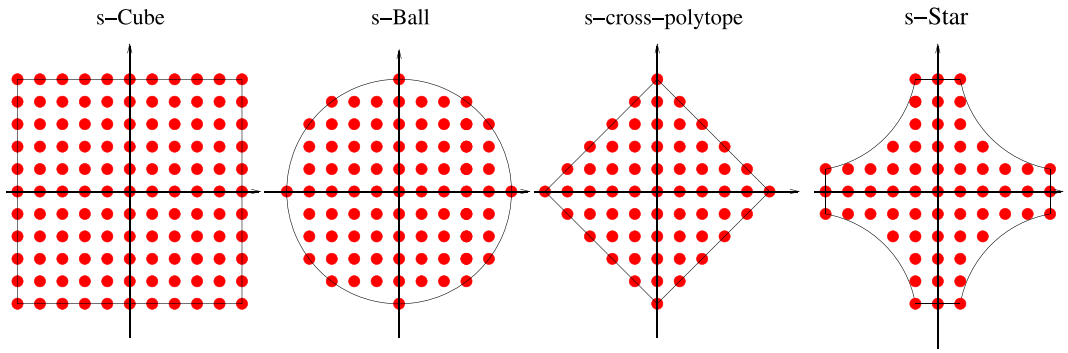

Fig. 1. Multi-indexes retained for different norms in 2 dimensions for $d = 5$. The different domains s-Cube, s-Ball, s-cross-polytope and s-star correspond to truncating the indexes by the norms: $\|\cdot\|_\infty$, $\|\cdot\|_2$, $\|\cdot\|_1$ and $\|\cdot\|_\alpha$, respectively.

The exact numbers of coefficients are of course slightly different due to the integer constraint, but nevertheless the volume formulas give a good indication of potential savings. In particular we notice that in the case of the 1-norm vs. the infinity norm a saving of $s!$ might be a possibility, while the α-norm turns the exponential dependency to a log-term.

1.2. Computing expansion coefficients

Having decided which terms to truncate and which to keep, one needs to evaluate the appropriate expansion coefficients. Using the orthogonality of the basis functions, the coefficients are given as

$$\widehat{f}(\mathbf{k}) = \int_{[0,1)^s} f(\mathbf{x}) e^{-2\pi i \mathbf{k}^T \mathbf{x}} \, d\mathbf{x}. \tag{5}$$

If the Fourier series is truncated according to the ∞-norm, these are usually approximated by the product trapezoidal rule

$$\widehat{f}(\mathbf{k}) \approx \frac{1}{(2d)^s} \sum_{\substack{0 \leqslant j_\ell < 2d \\ 1 \leqslant \ell \leqslant s}} f(\mathbf{x_j}) e^{-2\pi i \mathbf{k}^T \mathbf{x_j}}; \quad \mathbf{x_j} = \left(\frac{j_1}{2d}, \frac{j_2}{2d}, \ldots, \frac{j_s}{2d} \right). \tag{6}$$

The collocation points now form a Cartesian product of the underlying equidistant 1-dim grid. All coefficients $\|k\|_\infty \leqslant d$ can be computed by the s-dimensional FFT algorithm, and the algorithm for the s-dimensional case becomes a straightforward extension of the one-dimensional algorithm. This simplicity and the availability of the efficient FFT algorithm are probably the reason that apparently all multidimensional pseudo-spectral methods are based on truncating the Fourier expansion by the infinity norm.

The product trapezoidal rule can of course be used for computing the coefficients corresponding to truncation by other norms, but with this approach the number of collocation points remains $N = (2d+1)^s$, so no real saving will be obtained.

In this paper we advocate the use of lattice rules for evaluating multidimensional Fourier coefficients. With the appropriate choice of a lattice rule we may obtain almost the same saving in the number of collocation points as in the number of terms in the Fourier series. Moreover we show how a multidimensional FFT becomes as straightforward to use for a lattice grid as for the Cartesian grid.

The idea of sampling the function on a lattice grid for efficient and accurate high-dimensional approximation was introduced and analyzed in [13] and [14]. Their quality measure of lattices and subsequent generation of good lattices are different from what we present here.

Our ultimate goal is to develop pseudo-spectral methods for high-dimensional problems which defeat 'the curse of dimensionality'. We conclude this paper by some numerical examples for spectral solution of the Poisson's equation on a lattice grid and compare the results with those obtained for the solution on a Cartesian grid.

2. Lattice rules

Lattice rules are uniform-weight numerical integration rules for integrating over the s-dimensional unit hypercube $[0,1)^s$. They take as their abscissae the points from an integration lattice lying inside the unit cube [27,29,16,28].

A *lattice* is a discrete set of points $\Lambda \subset \mathbb{R}^s$ closed under addition and subtraction. The *unit lattice*, Λ_0, is the set of all integer points. Integration lattices should always contain $\Lambda_0 \subset \Lambda$ as a sublattice. This property ensures the periodicity of an integration lattice. Let $\Lambda/\Lambda_0 = \Lambda \cap [0,1)^s$. A lattice integration rule is defined as the following approximation to the continuous integral

$$Qf = \frac{1}{N_\Lambda} \sum_{\mathbf{x}_j \in \Lambda/\Lambda_0} f(\mathbf{x}_j) \approx If = \int_{[0,1)^s} f(\mathbf{x})\, d\mathbf{x},$$

where N_Λ is the number of lattice points in Λ/Λ_0. The lattice rule can always be presented in a more explicit form

$$Qf = \frac{1}{d_1 d_2 \cdots d_t} \sum_{j_1=0}^{d_1-1} \sum_{j_2=0}^{d_2-1} \cdots \sum_{j_t=0}^{d_t-1} f\left(\left\{ \sum_{i=1}^{t} j_i \frac{\mathbf{z}_i}{d_i} \right\} \right), \tag{7}$$

where $t \leqslant s$, d_i are positive integers and $\mathbf{z}_i \in \Lambda_0$. By the notation $\{\mathbf{x}\}$ we understand the *fractional part* of $\mathbf{x}$. Thus $\{\mathbf{x}\} \in [0,1)^s$.

Eq. (7) is not a unique representation of a lattice rule. There are many different choices of $t, d_1, \ldots, d_t$ and $\mathbf{z}_i$ which describe the same lattice rule. But when written with the minimal t and $\gcd(d_i, \mathbf{z}_i) = 1$ the rule is *non-repetitive*, meaning that each abscissa occurs exactly once. The number of function values needed is then $N_\Lambda = d_1 d_2 \cdots d_t$, and t is called the *rank* of the lattice rule [17]. A convenient way to specify an integration lattice is by an $s \times t$ integer matrix $A = (\mathbf{z}_1 | \cdots | \mathbf{z}_t)$ and a $t \times t$ diagonal matrix $D = \operatorname{diag}(d_1, \ldots, d_t)$. Then

$$\mathbf{x} \in \Lambda \quad \Leftrightarrow \quad \mathbf{x} = AD^{-1}\mathbf{j}; \quad \mathbf{j} \in Z^t. \tag{8}$$

Letting

$$Z_D = Z_{d_1} \times \cdots \times Z_{d_t} \tag{9}$$

denote the finite t-dimensional integer lattice with period d_i in direction i, we identify $\Lambda/\Lambda_0 \simeq Z_D$ through (8). What makes lattice rules interesting candidates for us is that for functions with an absolutely convergent Fourier series, their error functional takes the simple form [27,28]

$$Ef = If - Qf = \sum_{\mathbf{k} \in \Lambda^\perp \setminus \{0\}} \widehat{f}(\mathbf{k}), \tag{10}$$

where $\Lambda^\perp \subset Z^s$ is the dual lattice of our integration lattice, Λ. The dual lattice is defined as:

$$\Lambda^\perp = \left\{ \mathbf{k} \in \mathbb{R}^s \;\middle|\; \mathbf{k}^T \mathbf{x} \in Z;\ \forall \mathbf{x} \in \Lambda \right\} = \left\{ \mathbf{k} \in Z^s \;\middle|\; A^T \mathbf{k} = D\ell \text{ for some } \ell \in Z^t \right\}. \tag{11}$$

The dual lattice is itself a lattice, and when Λ is an integration lattice the dual lattice is contained in Z^s.

Using (10) it is possible to define the kind of lattice rule we need for our computation. For an efficient computation we seek the lattice with the minimal number of points N_Λ, such that all non-zero points in the dual lattice have a norm greater than d,

$$\min\left\{ N_\Lambda \;\middle|\; \|\mathbf{k}\|_p > d;\ \forall \mathbf{k} \in \Lambda^\perp \setminus \{0\} \right\}. \tag{12}$$

This general strategy allows us to design rules particularly appropriate for the function at hand. Unfortunately this optimization problem has no simple, general solution. Only for the infinity norm the solution is known. Here the answer is simply the product trapezoidal rule.

In the case of the 1-norm and the α-norm, huge computer searches have been applied to the problem. A numerical integration scheme is of *trigonometric degree* d if it integrates exactly all trigonometric polynomials of degree less or equal to d. For lattice rules, the trigonometric degree correspond to the 1-norm of the non-zero point in the dual lattice closest to the origin. Optimal rules of arbitrary dimension and degree are not known, but there exists a lower bound on N [6], and there exist quadrature rules which obtain this bound for $s = 2$ or $d \leqslant 3$. In addition the lattice rule of $N_\Lambda = 38$, $d = 5$, $s = 3$ based on Minkowski's celebrated critical lattice [24] is known to be optimal. All rules of optimal trigonometric degree are lattice rules. Moreover, all other best-known quadrature rules, with respect to trigonometric degree for fixed dimension and degree, are lattice rules. In [5,22] lattice rules of high trigonometric degree in 3, 4 and 5 dimensions are reported. These are found by exhaustive searches. These rules are optimal as they minimize the number of abscissas needed for computing one integral. When many related integrals are needed as in the Fourier transform, other choices may turn out to yield higher computational efficiency. In particular since a t-dim FFT is needed (see Section 3) one might expect lattice rules where $d_1, d_2, \ldots, d_t$ are all powers of 2 to be particularly attractive. We will, however, not address this issue here, as we consider that a technicality. Furthermore, efficient FFTs for arbitrary N are now readily available.

For optimal numerical integration rules of functions in Korobov spaces, the design criterion has been the *Zaremba rho-index* $\rho(\mathbf{k})$ defined by Eq. (3). Long lists of optimal lattice rules found by computer search based on this criterion are reported in [23,12,2,18–20].

The searches reported in [23,12,2] are restricted to lattice rules where $t = 1$. This restriction does not apply to those reported in [18–20,5,22]. A technical hurdle with the last group is that the lattices listed here are described by the generator matrix of their dual lattice, while we need it in the t–$\mathbf{z}$ form (7). How to obtain a t–$\mathbf{z}$ form from the generator matrix is explained in [17].

Only when restricting the search to subsets of integration lattices, a general pattern has been found. In these cases formulas for generating lattice rules are known. One such family of rules are those generated by skew-circulant matrices. Formulas for these rules of dimensions 3, 4, 5 and 6 of arbitrary degree are given in [21,4]. For convenience, these rules have been used in our computation where the 1-norm is the target. For rules optimized for the α-norm we have used the best rules listed in [23,12,18–20].

3. Computational issues

3.1. The FFT on lattice grids

Using a lattice rule to evaluate the Fourier coefficients in (5), we obtain approximate Fourier coefficients $\widehat{f}_\Lambda(\mathbf{k})$ as

$$\widehat{f}(\mathbf{k}) \approx \widehat{f}_\Lambda(\mathbf{k}) = \frac{1}{N_\Lambda} \sum_{\mathbf{x}\in\Lambda/\Lambda_0} f(\mathbf{x}) e^{-2\pi i \mathbf{k}^T \mathbf{x}} \tag{13}$$

$$= \frac{1}{N_\Lambda} \sum_{\mathbf{j}\in Z_D} f(\mathbf{x}) e^{-2\pi i \mathbf{k}^T A D^{-1} \mathbf{j}} \quad \text{for } \mathbf{k} \in Z^s. \tag{14}$$

The approximate Fourier coefficients satisfy $\widehat{f}_\Lambda(\mathbf{k}) = \widehat{f}_\Lambda(\tilde{\mathbf{k}})$ whenever $\mathbf{k} - \tilde{\mathbf{k}} \in \Lambda^\perp$.

The standard DFT over the t-dimensional finite lattice Z_D is given as

$$\widehat{f}_{\mathrm{DFT}}(\mathbf{m}) = \frac{1}{N_\Lambda} \sum_{\mathbf{j}\in Z_D} f(\mathbf{x}_j) e^{-2\pi i \mathbf{m}^T D^{-1} \mathbf{j}} \quad \text{for } \mathbf{m} \in Z_D, \tag{15}$$

where $\mathbf{x}_\mathbf{j} = AD^{-1}\mathbf{j}$. This is computed in $O(N_\Lambda \log N_\Lambda)$ operations with a t-dimensional FFT. Letting

$$\mathbf{m} = A^T \mathbf{k} + D\ell \quad \text{for some } \mathbf{k} \in Z^s \text{ and } \ell \in Z^t, \tag{16}$$

we find

$$\widehat{f}_\Lambda(\mathbf{k}) = \widehat{f}_{\mathrm{DFT}}(\mathbf{m}). \tag{17}$$

Note that the map $\mathbf{k} \mapsto \mathbf{m} : Z^s \to Z_D$ is surjective, but not injective. From (11) we see that two points $\mathbf{k}$ and $\tilde{\mathbf{k}}$ correspond to the same point $\mathbf{m} \in Z_D$ if and only if one of the points is an alias of the other, that is $\mathbf{k} - \tilde{\mathbf{k}} \in \Lambda^\perp$. Thus $\widehat{f}_{\mathrm{DFT}}(\mathbf{m})$ for $\mathbf{m} \in Z_D$ correspond to non-aliased approximate Fourier coefficients $\widehat{f}_\Lambda(\mathbf{k})$ in a 1–1 manner.

Having chosen our design criterion for the lattice rule and found the optimal rule for fixed dimension and degree, the typical situation is that this rule will use more lattice points, $N_\Lambda(s,d)$ than the number of coefficients we require it to evaluate. This simply means that our FFT routine not only computes the value of all $\|\mathbf{k}\|_1 \leqslant d$ or $\rho(\mathbf{k}) \leqslant d$, but some additional ones where $\|\mathbf{k}\|_1 > d$. As the underlying assumption is that the terms in Fourier series associated with these additional coefficients give no significant contribution, we can in principle ignore them. However, using our Fourier expansion in a pseudo-spectral method, we will sooner or later like to return to 'grid-point' space by applying an inverse FFT. We would then need to pad the coefficients in frequency space by zeros if coefficients are thrown out. However, if these extra coefficients are computed, we might as well keep them, as there is no extra cost associated with it. Keeping these terms will never do any harm, and will generally improve the accuracy slightly.

The obvious choice is to extend our index-set or frequency domain with $\mathbf{k}$'s having smallest possible norm larger than d. For these terms the mapping from $\mathbf{m}$ index to $\mathbf{k}$-index is not unique, as there might be tie-breaking in the smallest norm condition. In our present code we simply assign these coefficients on a "first-come-first-served" basis, while at the same time making sure that each $\mathbf{m}$ corresponds to one and only one $\mathbf{k}$ not on the dual lattice. Other assignment procedures are of course possible here. We will call our index set S and we note that by construction the order of S is $|S| = N_\Lambda$ and that $\{\mathbf{k} \mid \|\mathbf{k}\|_p < d\} \subseteq S$, implying that the terms we intend to represent are included.

3.2. Spectral differentiation on lattice grids

We may now construct an interpolant for f by sampling it on our lattice Λ and approximating $\hat{f}(\mathbf{k})$ by using a lattice rule for the integration

$$L^f_\Lambda(\mathbf{x}) = \sum_{\mathbf{k}\in S} \widehat{f}_\Lambda(\mathbf{k}) e^{2\pi i \mathbf{k}^T \mathbf{x}}; \qquad \widehat{f}_\Lambda(\mathbf{k}) = \frac{1}{N_\Lambda} \sum_{\mathbf{x}_j \in \Lambda} f(\mathbf{x}_j) e^{-2\pi i \mathbf{k}^T \mathbf{x}_j}. \tag{18}$$

To see that $L^f_\Lambda(\mathbf{x})$ interpolates $f(\mathbf{x})$ on the lattice, we recall the discrete orthogonality of exponentials

$$\sum_{\mathbf{k}\in S} e^{2\pi i \mathbf{k}^T (\mathbf{x}_p - \mathbf{x}_j)} = N_\Lambda \delta_{p,j}. \tag{19}$$

Thus for $\mathbf{x}_p \in \Lambda$ we have:

$$L^f_\Lambda(\mathbf{x}_p) = \sum_{\mathbf{k}\in S} \widehat{f}_\Lambda(\mathbf{k}) e^{2\pi i \mathbf{k}^T \mathbf{x}_p} = \sum_{\mathbf{k}\in S} \frac{1}{N_\Lambda} \sum_{\mathbf{x}_j\in\Lambda} f(\mathbf{x}_j) e^{2\pi i \mathbf{k}^T(\mathbf{x}_p-\mathbf{x}_j)}$$
$$= \frac{1}{N_\Lambda} \sum_{\mathbf{x}_j\in\Lambda} f(\mathbf{x}_j) \sum_{\mathbf{k}\in S} e^{2\pi i \mathbf{k}^T(\mathbf{x}_p-\mathbf{x}_j)} = \frac{1}{N_\Lambda} \sum_{\mathbf{x}_j\in\Lambda} f(\mathbf{x}_j) N_\Lambda \delta_{p,j} = f(\mathbf{x}_p).$$

The error in $\widehat{f}_\Lambda(\mathbf{k})$ as an approximation to $\hat{f}(\mathbf{k})$ is the aliasing error.

$$\widehat{f}_\Lambda(\mathbf{k}) - \hat{f}(\mathbf{k}) = \sum_{\mathbf{l}\in\Lambda^\perp\setminus\{0\}} \hat{f}(\mathbf{l}+\mathbf{k}), \quad \mathbf{k}\in S. \tag{20}$$

The important point to notice is that this error is restricted to terms we are not able to resolve on our grid. Those terms we assume are insignificant. Now differentiating both expansions and taking the differences we get:

$$\frac{\partial}{\partial x_j} f(\mathbf{x}) - \frac{\partial}{\partial x_j} L^f_\Lambda(\mathbf{x}) = \frac{\partial}{\partial x_j}\big(f(\mathbf{x}) - L^f_\Lambda(\mathbf{x})\big) = \frac{\partial}{\partial x_j}\Big(\sum_{\mathbf{k}\in Z^s} \hat{f}(\mathbf{k}) e^{2\pi i \mathbf{k}^T\mathbf{x}} - \sum_{\mathbf{k}\in S} \widehat{f}_\Lambda(\mathbf{k}) e^{2\pi i \mathbf{k}^T\mathbf{x}}\Big)$$
$$= \sum_{\mathbf{k}\in Z^s} \hat{f}(\mathbf{k}) 2\pi i k_j e^{2\pi i \mathbf{k}^T\mathbf{x}} - \sum_{\mathbf{k}\in S} \widehat{f}_\Lambda(\mathbf{k}) 2\pi i k_j e^{2\pi i \mathbf{k}^T\mathbf{x}} = +2\pi i \sum_{\mathbf{k}\notin S} \hat{f}(\mathbf{k})(k_j - l_j) e^{2\pi i \mathbf{k}^T\mathbf{x}},$$

where l_j corresponds to the j-th component of the $\mathbf{l}\in S$ which aliases with $\mathbf{k}$. As expected the error only occurs in terms we are not able to resolve on our grid.

By extension we see that the effect of applying the s-dim Laplacian to the Fourier expansion corresponds to multiplying each term by $-(2\pi)^2(k_1^2+k_2^2+\cdots+k_s^2) = -(2\pi)^2\|\mathbf{k}\|_2^2$.

However, here our computed Fourier coefficients $\widehat{f}_{\mathrm{DFT}}(\mathbf{m})$ are given in terms of the t-dim index $\mathbf{m}$, related to $\mathbf{k}$ through Eq. (16).

In our code we produce all possible $\mathbf{k}$ in order of increasing 1-norm (or ρ-index) and compute the corresponding $\mathbf{m}$ using (16) until all $\mathbf{m}$'s are assigned a corresponding $\mathbf{k}$.

For a lattice rule of degree $2d-1$ this simple procedure guarantees that all $\|\mathbf{k}\|_1 < d$ are assigned a Fourier coefficient, which is what we require. In addition some extra terms in the expansion are also included in the approximation.

4. Numerical experiments

As our test case we solve the s-dim Poisson equation on $[0,1)^s$ with periodic boundary conditions

$$\Delta u(\mathbf{x}) = f(\mathbf{x}); \quad \mathbf{x}\in[0,1)^s,$$
$$u(x_i = 0) = u(x_i = 1); \quad i = 1,\ldots,s.$$

The pseudo-spectral algorithm is simply:

1. Sample $f(\mathbf{x})$ on a lattice grid.
2. Compute $\widehat{f}_{\mathrm{DFT}}(\mathbf{m})$ by (15) (using an FFT).
3. $\widehat{u}_{\mathrm{DFT}}(\mathbf{m}) = -\widehat{f}_{\mathrm{DFT}}(\mathbf{m})/(2\pi\|\mathbf{k}\|_2)^2$, where $\mathbf{k}$ and $\mathbf{m}$ are related by (16) and $\|\mathbf{k}\|_2^2$ is minimal among the aliasing $\mathbf{k}$'s.
4. Find $u(\mathbf{x})$ from $\widehat{u}_{\mathrm{DFT}}(\mathbf{m})$ using the inverse FFT.

Here the zero'th Fourier coefficient is discarded, as it is undetermined for periodic solutions.

We do three different solutions listed in the table:

Test function 1	Test function 2	Test function 3
Gaussian	Periodic	Polynomial
$u(\mathbf{x}) = e^{-100\sum_{i=1}^s (x_i-0.5)^2}$	$u(\mathbf{x}) = \prod_{i=1}^s (e^{\sin 2\pi x_i} - 1)$	$u(\mathbf{x}) = \prod_{i=1}^s (x_i-1)^2 x_i^2$

These test functions are chosen because their Fourier expansions have different characteristics and as such can reveal the strengths and weaknesses of the three different grids.

In 2D we can plot the Fourier spectrum of these solutions. These plots show that for *Test function 1* Fourier coefficients of equal magnitude form a circle, indicating that coefficients of equal 2-norm are of equal magnitude. For such a problem there is no reason to believe that our 1-norm lattice grid should do better than the standard Cartesian grid. The α-norm lattice grid may be a bad choice here.

Table 2
The error is measured by the discrete L^2-norm.

s	N_Λ	$\mathbf{z}$	Rel. err.
2	149	(1, 44)	0.00122
3	547	(1, 338, 397)	0.00162
4	2011	(1, 1125, 698, 820)	0.00423
5	7411	(1, 2721, 6781, 259, 6475)	0.00322
6	27 329	(1, 19 037, 5175,19 525, 10 217, 26 546)	0.00353
7	99 991	(1, 61 204, 42 555, 94 308, 16935, 91 256, 53 985)	0.00341

Table 3
The error is measured by the discrete L^2-norm.

s	Lattice grid	Cartesian grid	
	$N_\Lambda = 4091$	$N_\Lambda = 4096$	
2	0.0000028	(64×64)	0.0004236
3	0.0000774	$(16 \times 16 \times 16)$	0.0090503
4	0.0012902	$(8 \times 8 \times 8 \times 8)$	0.0615971
5	0.0064880		
6	0.0227155	$(4 \times 4 \times 4 \times 4 \times 4 \times 4)$	0.3837561
7	0.0762831		
8	0.1274343		

For *Test function 2* the coefficients of equal size seem to line up somewhere between the circular form and the cross polytope form. This should give an advantage to the 1-norm lattice grid. For *Test function 3* contour lines of equal sized coefficients appear to have the "star-shaped" form of the α-norm. In this case using the full tensor-product would be a terrible waste, and we expect a restriction to terms possible to determine by the 1-norm lattice grid to perform much better, almost by a factor of $s!$ in s-dimensions. The α-norm lattice grid should do even better.

Our code can handle any dimension, and we have run experiments for all three test functions and the three different grid types on a large span of grid sizes in 2, 3 and 4 dimensions. In Fig. 3 we show plots of our results in 3D and 4D.

For modest N-values all 3 grids perform equally well on the Gaussian test functions. Fig. 2 indicates that this should be the case for Cartesian grid and 1-norm lattice grid. It may be more surprising that α-norm lattice grid shows equal performance. Only for high values of N the α-norm grid falls behind and also the 1-norm grid shows somewhat weaker behavior than standard regular Cartesian grid. "High values of N" are dimension-dependent, so this effect is best seen in 3 dimensions.

For the second test function, the periodic one, Fig. 2 indicates that the 1-norm lattice grid should have an edge, and indeed it has. In particular for high values of N. In this case the grid based on α-norm lattice performs badly.

The Fourier coefficients of the third test function have the star shape of the α-norm. This is where the α-norm lattice grid should excel, and indeed it does. However, the 1-norm lattice grid is almost as good, while the Cartesian grid performs badly.

In all cases the error decays monotonically with increasing grid density for the Cartesian grid, while the decay is much less smooth for the two versions of lattice grids. There are at least two reasons for this. First of all, if the error should decay smoothly it would be as a function of d, not as a function of N, and for the lattice grids we are using here the relation between d and N is not smooth. Secondly, when increasing d, we not only increase the grid density, but also change the shape of the grid.

To demonstrate the potential of our method in higher dimensions, we have done some limited experiments for test function no. 3, as this case represents the problem type where we claim that lattice grids outperform Cartesian grids. The lists of lattice rules based on solving (12) for the α-norm do not go beyond dimension 5, and in dimension 5 only few entries exist. There does, however, exist an algorithm for computing good, rank 1, lattice rules for functions in Korobov spaces [30]. This is a greedy algorithm which for a prime N_Λ computes the components of the $\mathbf{z}$-vector component by component. The s-th component is computed as the one which minimize the worst case error for functions in the Korobov space, assuming the first $s-1$ components are fixed. We have implemented this algorithm and used lattices produced by this for our experiments. In the first experiment, reported in Table 2, we have chosen N_Λ to be a prime such that $N_\Lambda \approx 40(\ln 40)^{s-1}$.

We observed that the error remains almost constant, implicating that the computational cost is indeed $O(d(\log d)^{s-1})$ for this problem when using a lattice grid.

For comparison with the standard Cartesian grid, we have computed solutions to the same problem on a $N_\Lambda = 2^{12} = 4096$ Cartesian grid, and compared it with the solution obtained on a lattice grid with $N_\Lambda = 4091$ for various dimensions. The results are reported in Table 3.

Table 3 demonstrates that for this particular problem we can compute the solution of the 5-dimensional problem on a lattice grid with better accuracy than the corresponding 3-dimensional problem on the Cartesian grid for the same cost.

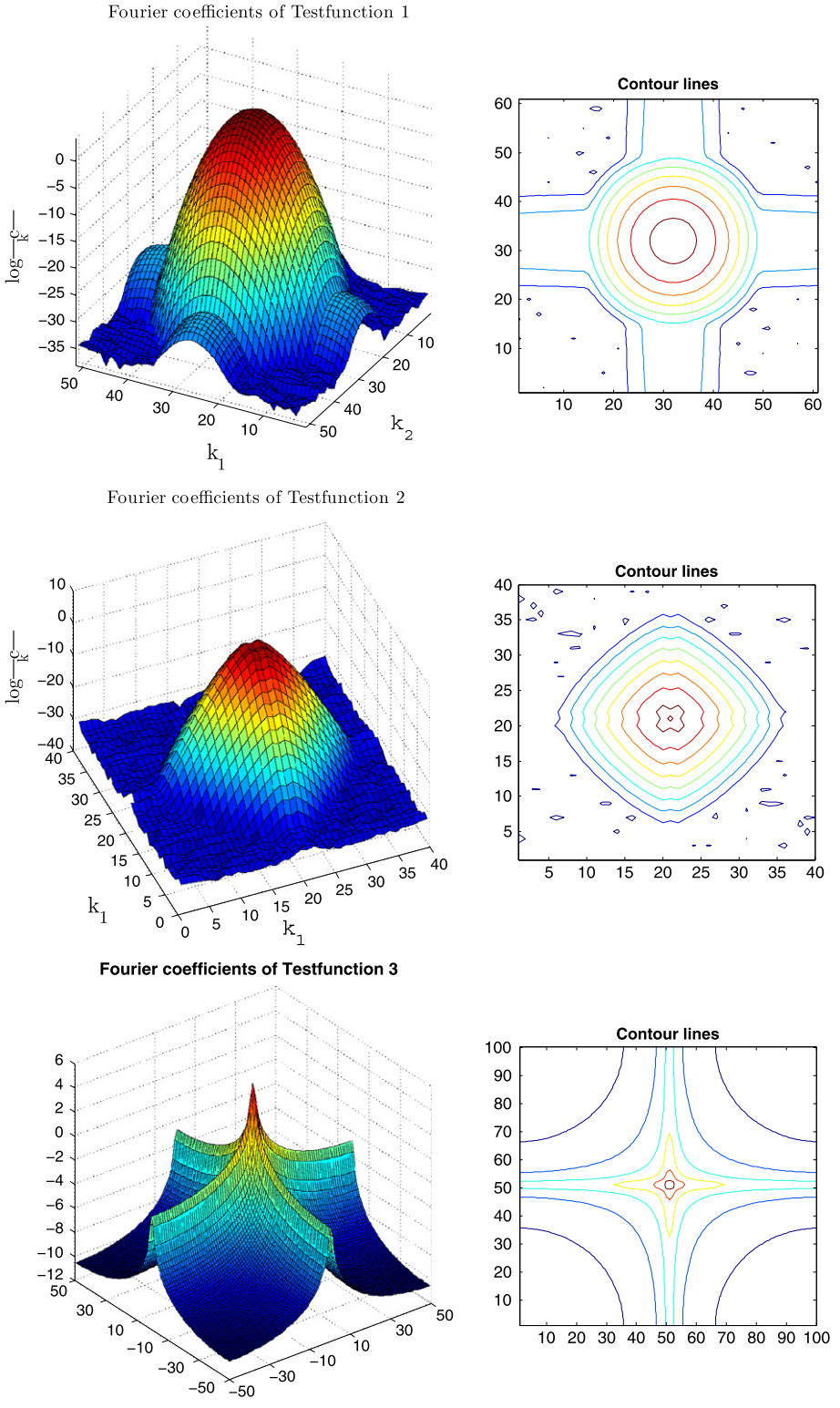

Fig. 2. The size of the Fourier coefficients for the 2D version of our 3 test functions. In the left column these are shown as surface-plot of $\log|\widehat{f}_{\Lambda}(\mathbf{k})|$. In the right column the same is shown as contour plots.

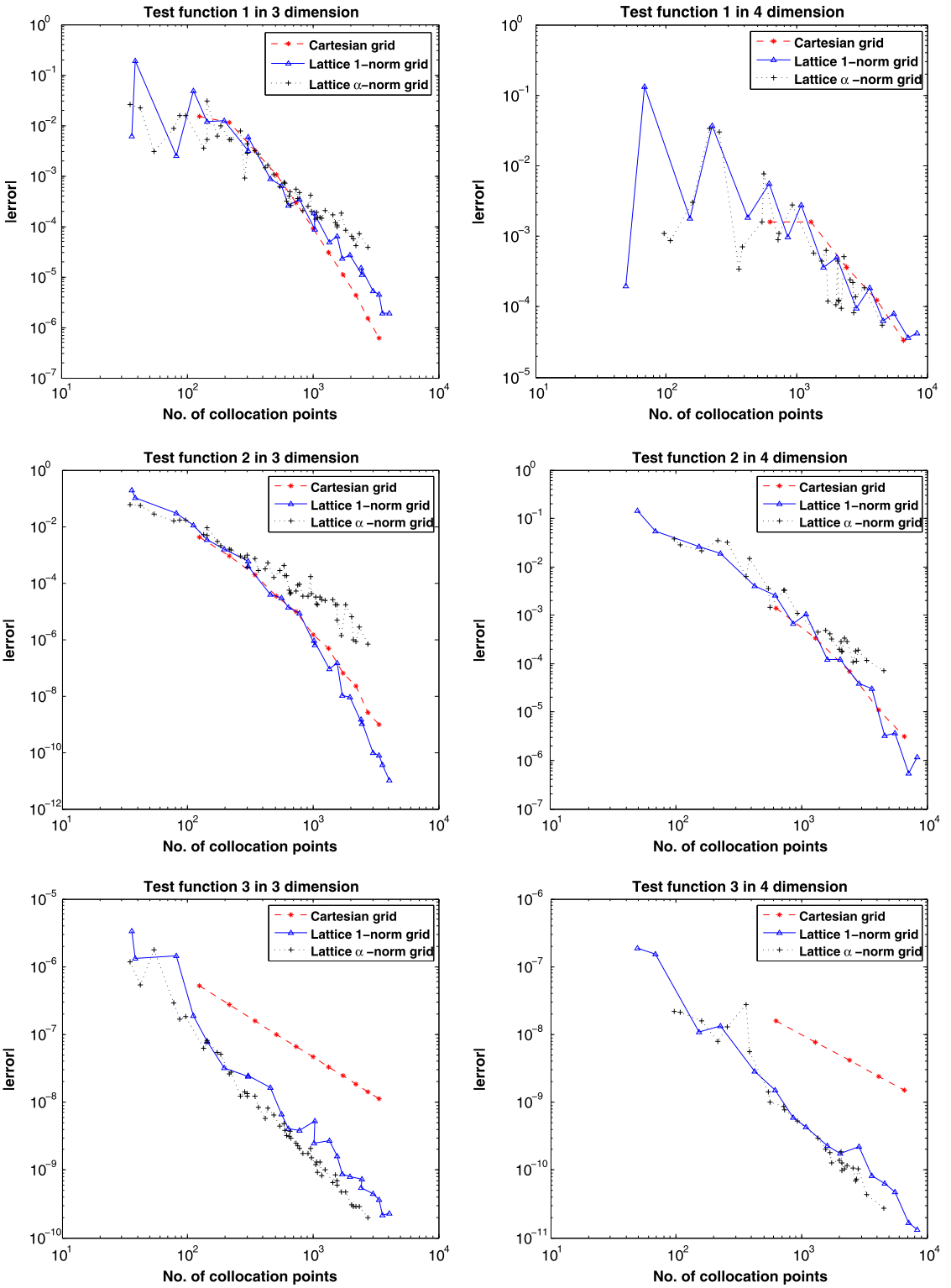

Fig. 3. LogLog-plot of the error as function of grid points for the 3 different types of grid in dimensions 3 and 4. The error is measured as $\|u_{ext} - u_{app}\|_2$.

5. Conclusions and related work

In this paper we have shown that lattice grids may help to reduce the exponential growth in complexity for spectral methods when moving to higher dimensions. The rate of success depends on the function we want to approximate. With knowledge of how its Fourier coefficients decay, we should choose our lattice grid, either based on lattice rules optimized for the α-norm or for the 1-norm. Theoretically, the savings in number of lattice points is $s!$ for truncation by the 1-norm, while truncation by the α-norm may move the exponential dependency of the complexity from $O(d^s)$ to $O(d(\log d)^s)$.

A practical problem is that lattice rules of optimal or near optimal trigonometric degree are not known for high dimensions and computer search for them is extremely expensive. The same is true for lattice rules of optimal *Zaremba rho-index*. However, in the latter case there does exist an alternative, the component-by-component algorithm suggested by Sloan and Reztsov [30]. This algorithm finds lattice rules designed to perform well for function in Korobov space. The difference is that the optimization criterion here is the worst case error, not the Zaremba rho-index. Thus we do not anymore explicitly eliminate terms in the error functional. However, the limited experiments we did in Section 4.1 indicate that this is no problem in practice. An advantage of this construction technique is that it can be extended to allow for adaptive weighting of the importance of the different directions [7]. As we expect the proposed technique to be superior to the traditional Cartesian grid for anisotropic problem, one might expect that in these cases one has a clear idea of the relative importance of the different directions. If this is the case this knowledge can easily be built into the rule construction algorithm without extra computational costs. Lately Nyuens and Cools [26] have improved the computational complexity of the component-by-component algorithm. By rewriting the algorithm as s matrix-vector multiplies and noticing that the matrix is circulant, they reduced the computational cost to $O(sN_\Lambda \log N_\Lambda)$ using an FFT algorithm.

An alternative to the present approach is to truncate the Fourier expansion according to the 2-norm. This approach is pursued in [1]. Optimal lattice points are then given by *densest lattice packing* points. In dimensions up to 8, these are given by certain root lattices [25]. The savings are 1.15, 1.4, 2.0, 2.8, 4.6, 8.0 and 16.0 in dimensions 2, 3, …, 8, thus smaller than the 1-norm savings. An other difference in this case is that these lattices do not contain the integer lattice Λ_0 as a sub-lattice. Thus periodicity is not with respect to the unit cube. E.g. for the 2D case, the periodicity is with respect to an equilateral hexagon and in 3D the periodic domain forms a rhombic dodecahedron. If the periodic problem is generated from an underlying infinite non-periodic problem by periodization, hexagonal and dodecahedral periodic domains may, however, yield smaller periodization errors than the unit cube periodization.

For functions in K^α sparse grid or 'hyperbolic cross approximation' [3] provides another strategy for constructing efficient quadrature rules, and thus provides an alternative to α-norm lattice rules. The underlying grid would be quite different with this approach, but the overall complexity in terms of grid points needed for computing all $\widehat{f}(\mathbf{k})$; $\forall \|\mathbf{k}\|_\alpha \leqslant d$, is of the same order. Hallatschek [11] devised an FFT algorithm for sparse grid. Using this FFT algorithm Kupka [15] solves the Helmholtz equation by exactly the same procedure as we use in this work. Gradinaru [8,9] solves the time-dependent Schrödinger equation on sparse grids. He applies operator splitting to the Hamiltonian operator which splits into a Laplacian operator and a potential operator. The propagation by the Laplacian is done in Fourier space and he samples the wave function on a sparse grid and uses Hallatschek's sparse grid FFT algorithm to compute the necessary coefficients in Fourier space.

In the works mentioned above the sparse grid could have been replaced with an α-norm lattice grid and Hallatschek's FFT algorithm with our FFT algorithm. The asymptotic complexity of the two approaches would have been the same, as would the theoretical error bounds. It could have been interesting to run a back-to-back comparison on practical efficiency and accuracy of the two approaches on the same problem, but that is beyond the scope of this paper.

References

[1] R.H. Bisseling, R. Kosloff, Optimal choice of grid points in multidimensional pseudospectral Fourier methods, Journal of Computational Physics 76 (1988) 243–262.
[2] M. Bourdeau, A. Pitre, Tables of good lattices in four and five dimensions, Numerische Mathematik 47 (1985) 39–43.
[3] H. Bungartz, M. Griebel, Sparse grids, Acta Numerica 13 (2004) 1–123.
[4] R. Cools, H. Govaerts, Five- and six-dimensional lattice rules generated by structured matrices, Journal of Complexity 19 (2003) 715–729.
[5] R. Cools, J.N. Lyness, Three- and four-dimensional k-optimal lattice rules of moderate trigonometric degree, Mathematics of Computation 70 (236) (2001) 1549–1567.
[6] R. Cools, I. Sloan, Minimal cubature formulae of trigonometric degree, Mathematics of Computation 65 (1996) 1583–1600.
[7] J. Dick, I. Sloan, X. Wang, H. Woźniakowski, Good lattice rules in weighted Korobov spaces with general weights, Numerische Mathematik 103 (2006) 63–97.
[8] V. Gradinaru, Fourier transform on sparse grids: code design and the time dependent Schrödinger equation, Computing 80 (2007) 1–22.
[9] V. Gradinaru, Strang splitting for the time dependent Schrödinger equation on sparse grid, SIAM Journal on Numerical Analysis 46 (1) (2008) 103–123.
[10] M. Griebel, J. Hamaekers, Sparse grid for the Schrödinger equation, Mathematical Modelling and Numerical Analysis 41 (2007) 215–247.
[11] K. Hallatschek, Fouriertransformation auf dünnen Gittern mit hierarchischen Basen, Numerische Mathematik 63 (1992) 83–97.
[12] G. Kedem, S.K. Zaremba, A table of good lattice points in three dimensions, Numerische Mathematik 23 (1974) 175–180.
[13] F.Y. Kuo, I.H. Sloan, H. Wozniakowski, Lattice rules for multivariate approximation in the worst case setting, in: H. Niederreiter, D. Talay (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 2004, Springer-Verlag, Berlin, 2006, pp. 289–330.
[14] F.Y. Kuo, I.H. Sloan, H. Wozniakowski, Lattice rule algorithm for multivariate approximation in the average case setting, Journal of Complexity 24 (2) (2008) 283–323.
[15] F.G. Kupka, in: C.-H. Lai, P.E. Bjørstad, M. Cross, O.B. Widlund (Eds.), Sparse Grid Spectral Methods and some Results from Approximation Theory Domain Decomposition 11, Greenwich, UK, 1999, pp. 57–64.
[16] J.N. Lyness, An introduction to lattice rules and their generator matrices, IMA Journal of Numerical Analysis 9 (1989) 405–419.

[17] J.N. Lyness, P. Keast, Application of the Smith normal form to the structure of lattice rules, SIAM Journal on Matrix Analysis and Applications 16 (1) (1995) 218–231.
[18] J.N. Lyness, T. Sørevik, A search program for finding optimal integration lattices, Computing 47 (1991) 103–120.
[19] J.N. Lyness, T. Sørevik, An algorithm for finding optimal integration lattices of composite order, BIT 32 (4) (1992) 665–675.
[20] J.N. Lyness, T. Sørevik, Lattice rules by component scaling, Mathematics of Computation 61 (1993) 799–820.
[21] J.N. Lyness, T. Sørevik, Four-dimensional lattice rules generated by skew-circulant matrices, Mathematics of Computation 73 (245) (2003) 279–295.
[22] J.N. Lyness, T. Sørevik, Five-dimensional *k*-optimal lattice rules, Mathematics of Computation 75 (255) (2006) 1467–1480.
[23] D. Maisonneuve, Recherche et utilisation des 'bon trellis', programming et résultats numériques, in: S.K. Zaremba (Ed.), Applications of Number Theory to Numerical Analysis, Academic Press, London, 1972, pp. 121–201.
[24] H. Minkowski, Gesammelte Abhandlung, Chelsea Publishing Company, New York, 1967.
[25] H.Z. Munthe-Kaas, On group Fourier analysis and symmetry preserving discretizations of PDEs, Journal of Physics A: Mathematical and Theoretical 39 (19) (2006) 5563–5584.
[26] D. Nyuens, R. Cools, Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing Kernel Hilbert spaces, Mathematics of Computation 75 (254) (2006) 903–920.
[27] I.H. Sloan, Lattice methods for multiple integration, Journal of Computational and Applied Mathematics 12–13 (1985) 131–143.
[28] I.H. Sloan, S. Joe, Lattice methods for multiple integration, Oxford University Press, 1994.
[29] I.H. Sloan, P.J. Kachoyan, Lattice methods for multiple integration: theory, error analysis and examples, SIAM Journal on Numerical Analysis 24 (1987) 116–128.
[30] I.H. Sloan, A.V. Reztsov, Component-by-component construction of good lattice rules, Mathematics of Computation 71 (237) (2002) 263–273.
[31] H. Yserentant, On the regularity of the electronic Schrödinger equation in Hilbert spaces of mixed derivatives, Numerische Mathematik 98 (2004) 731–759.
[32] H. Yserentant, Sparse grid spaces for the numerical solution of the electronic Schrödinger, Numerische Mathematik 101 (2005) 381–389.