Reliable Adaptive Cubature Using Digital Sequences

Fred J. Hickernell and Lluís Antoni Jiménez Rugama

In honor of Ilya M. Sobol'

Abstract Quasi-Monte Carlo cubature methods often sample the integrand using Sobol' (or other digital) sequences to obtain higher accuracy than IID sampling. An important question is how to conservatively estimate the error of a digital sequence cubature so that the sampling can be terminated when the desired tolerance is reached. We propose an error bound based on the discrete Walsh coefficients of the integrand and use this error bound to construct an adaptive digital sequence cubature algorithm. The error bound and the corresponding algorithm are guaranteed to work for integrands whose true Walsh coefficients satisfy certain cone conditions. Intuitively, these cone conditions imply that the ordered Walsh coefficients do not dip down for a long stretch and then jump back up. An upper bound on the cost of our new algorithm is given in terms of the *unknown* decay rate of the Walsh coefficients.

1 Introduction

Quasi-Monte Carlo cubature rules approximate multidimensional integrals over the unit cube by an equally weighted sample average of the integrand values at the first n nodes from some sequence $\{\boldsymbol{z}_i\}_{i=0}^{\infty}$. This node sequence should be chosen to minimize the error, and for this one can appeal to Koksma-Hlawka type error bounds of the form

$$\left| \int_{[0,1)^d} f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - \frac{1}{n} \sum_{i=0}^{n-1} f(\boldsymbol{z}_i) \right| \le D(\{\boldsymbol{z}_i\}_{i=0}^{n-1}) V(f). \tag{1}$$

Fred J. Hickernell · Lluís Antoni Jiménez Rugama
Department of Applied Mathematics, Illinois Institute of Technology, 10 W. 32nd Street, E1-208, Chicago, IL 60616, USA e-mail: hickernell@iit.edu, e-mail: ljimene1@hawk.iit.edu

The discrepancy, $D(\{\boldsymbol{z}_i\}_{i=0}^{n-1})$, measures how far the empirical distribution of the first n nodes differs from the uniform distribution. The variation, $V(f)$, is some semi-norm of the integrand, f. The definitions of the discrepancy and variation are linked to each other. Examples of such error bounds are given by [3, Chap. 2–3], [4], [11, Sec. 5.6], [12, Chap. 2–3], and [14, Chap. 9].

A practical problem is how large to choose n so that the absolute error is smaller than some user-defined tolerance, ε. Error bounds of the form (1) do not help in this regard because it is too hard to compute $V(f)$, which is typically defined in terms of integrals of mixed partial derivatives of f.

This article addresses the challenge of reliable error estimation for quasi-Monte Carlo cubature based on digital sequences, of which Sobol' sequences are the most popular example. The vector space structure underlying these digital sequences facilitates a convenient expression for the error in terms of the (Fourier)-Walsh coefficients of the integrand. Discrete Walsh coefficients can be computed efficiently, and their decay provides a reliable cubature error estimate. Underpinning this analysis is the assumption that the integrands lie in a cone defined in terms of their true Walsh coefficients; see (13).

The next section introduces digital sequences and their underlying algebraic structure. Sect. 3 explains how the cubature error using digital sequences as nodes can be elegantly formulated in terms of the Walsh series representation of the integrand. Our contributions begin in Sect. 4, where we derive a reliable data-based cubature error bound for a cone of integrands, (16), and an adaptive cubature algorithm based on that error bound, Algorithm 2. The cost of the algorithm is also represented in terms of the unknown decay of the Walsh series coefficients and the error tolerance in Theorem 1. A numerical example and discussion then conclude this article. A parallel development for cubature based on lattice rules is given in [9].

2 Digital Sequences

The integrands considered here are defined over the half open d-dimensional unit cube, $[0,1)^d$. For integration problems on other domains one may often transform the integration variable so that the problem is defined on $[0,1)^d$. See [1, 5, 6, 7, 8] for some discussion of variable transformations and the related error analysis. The example in Sect. 5 also employs a variable transformation.

Digital sequences are defined in terms of digitwise addition. Let b be a prime number; $b = 2$ is the choice made for Sobol' sequences. Digitwise addition, $\oplus$, and negation, $\ominus$, are defined as in terms of the proper b-ary expansions of points in $[0,1)^d$:

$$\boldsymbol{x} = \left(\sum_{\ell=1}^{\infty} x_{j\ell} b^{-\ell}\right)_{j=1}^{d}, \quad \boldsymbol{t} = \left(\sum_{\ell=1}^{\infty} t_{j\ell} b^{-\ell}\right)_{j=1}^{d}, \qquad x_{j\ell}, t_{j\ell} \in \mathbb{F}_b := \{0, \ldots, b-1\},$$

$$\boldsymbol{x} \oplus \boldsymbol{t} = \left(\sum_{\ell=1}^{\infty} [(x_{j\ell} + t_{j\ell}) \bmod b] b^{-\ell} \pmod 1 \right)_{j=1}^{d},$$

$$\ominus \boldsymbol{x} = \left(\sum_{\ell=1}^{\infty} [-x_{j\ell} \bmod b] b^{-\ell} \right)_{j=1}^{d}, \qquad a\boldsymbol{x} := \underbrace{\boldsymbol{x} \oplus \cdots \oplus \boldsymbol{x}}_{a \text{ times}} \ \forall a \in \mathbb{F}_b.$$

Here $\boldsymbol{x} \ominus \boldsymbol{t}$ means $\boldsymbol{x} \oplus (\ominus \boldsymbol{t})$.

We do not have associativity for all of $[0,1)^d$. For example, for $b = 2$,

$$1/3 = {}_2 0.010101\ldots, \quad 1/6 = {}_2 0.001010\ldots,$$
$$1/3 \oplus 1/3 = {}_2 0.00000\ldots = 0, \quad 1/3 \oplus 1/6 = {}_2 0.011111\ldots = 1/2,$$
$$(1/3 \oplus 1/3) \oplus 1/6 = 0 \oplus 1/6 = 1/6, \quad 1/3 \oplus (1/3 \oplus 1/6) = 1/3 \oplus 1/2 = 5/6.$$

This lack of associativity comes from the possibility of digitwise addition resulting in an infinite trail of digits $b-1$.

Define the Boolean operator that checks whether digitwise addition of two points does not result in an infinite trail of digits $b-1$:

$$\mathrm{ok}(\boldsymbol{x}, \boldsymbol{t}) = \begin{cases} \text{true}, & \min_{j=1,\ldots,d} \sup\{\ell : [(x_{j\ell} + t_{j\ell}) \bmod b] \neq b-1\} = \infty, \\ \text{false}, & \text{otherwise.} \end{cases} \tag{2}$$

If $\mathscr{P} \subset [0,1)^d$ is some set that is closed under $\oplus$ and $\mathrm{ok}(\boldsymbol{x}, \boldsymbol{t}) = \text{true}$ for all $\boldsymbol{x}, \boldsymbol{t} \in \mathscr{P}$, then associativity holds for all points in $\mathscr{P}$. Moreover, $\mathscr{P}$ is an Abelian group and also a vector space over the field $\mathbb{F}_b$.

Suppose that $\mathscr{P}_\infty = \{\boldsymbol{z}_i\}_{i=0}^{\infty} \subset [0,1)^d$ is such a vector space that satisfies the following additional conditions:

$$\{\boldsymbol{z}_1, \boldsymbol{z}_b, \boldsymbol{z}_{b^2}, \ldots\} \text{ is a set of linearly independent points,} \tag{3a}$$

$$\boldsymbol{z}_i = \sum_{\ell=0}^{\infty} i_\ell \boldsymbol{z}_{b^\ell}, \qquad \text{where } i = \sum_{\ell=0}^{\infty} i_\ell b^\ell \in \mathbb{N}_0, \quad i_\ell \in \mathbb{F}_b, \tag{3b}$$

Such a $\mathscr{P}_\infty$ is called a *digital sequence*. Moreover, any $\mathscr{P}_m := \{\boldsymbol{z}_i\}_{i=0}^{b^m-1}$ is a subspace of $\mathscr{P}_\infty$ and is called a *digital net*. From this definition it is clear that

$$\mathscr{P}_0 = \{\boldsymbol{0}\} \subset \mathscr{P}_1 = \{\boldsymbol{0}, \boldsymbol{z}_1, \ldots, (b-1)\boldsymbol{z}_1\} \subset \mathscr{P}_2 \subset \cdots \subset \mathscr{P}_\infty = \{\boldsymbol{z}_i\}_{i=0}^{\infty}.$$

The Sobol' sequence works in base $b = 2$ and makes a careful choice of the basis $\{\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{z}_4, \ldots\}$ so that the points are evenly distributed. Figure 1a) displays the initial points of the two-dimensional Sobol' sequence. In Figure 1b) the Sobol' sequence has been linearly scrambled to obtain another digital sequence and then digitally shifted.

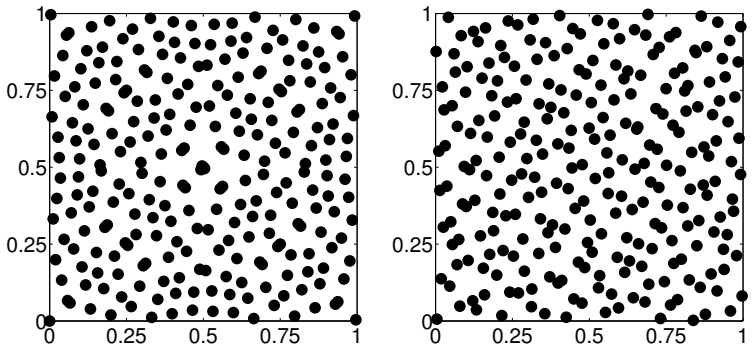

Fig. 1 a) 256 Sobol' points, b) 256 scrambled and digitally shifted Sobol' points.

3 Walsh Series

Non-negative integer vectors are used to index the Walsh series for the integrands. The set $\mathbb{N}_0^d$ is a vector space under digitwise addition, $\oplus$, and the field $\mathbb{F}_b$. Digitwise addition and negation are defined as follows for all $\boldsymbol{k},\boldsymbol{l} \in \mathbb{N}_0^d$:

$$\boldsymbol{k} = \left(\sum_{\ell=0}^{\infty} k_{j\ell} b^\ell\right)_{j=1}^d, \quad \boldsymbol{l} = \left(\sum_{\ell=0}^{\infty} l_{j\ell} b^\ell\right)_{j=1}^d, \qquad k_{j\ell}, l_{j\ell} \in \mathbb{F}_b,$$

$$\boldsymbol{k} \oplus \boldsymbol{l} = \left(\sum_{\ell=0}^{\infty} [(k_{j\ell} + l_{j\ell}) \bmod b] b^\ell\right)_{j=1}^d,$$

$$\ominus \boldsymbol{k} = \left(\sum_{\ell=0}^{\infty} (b - k_{j\ell}) b^\ell\right)_{j=1}^d, \qquad a\boldsymbol{k} := \underbrace{\boldsymbol{k} \oplus \cdots \oplus \boldsymbol{k}}_{a \text{ times}} \ \forall a \in \mathbb{F}_b.$$

The set $\mathbb{N}_{0,m} := \{0, \dots, b^m - 1\}$ is a subspace of $\mathbb{N}_0^d$.

For each wavenumber $\boldsymbol{k} \in \mathbb{N}_0^d$ a function $\langle \boldsymbol{k}, \cdot \rangle : [0,1)^d \to \mathbb{F}_b$ is defined as

$$\langle \boldsymbol{k}, \boldsymbol{x} \rangle := \sum_{j=1}^{d} \sum_{\ell=0}^{\infty} k_{j\ell} x_{j,\ell+1} \pmod b. \tag{4a}$$

For all points $\boldsymbol{t}, \boldsymbol{x} \in [0,1)^d$, wavenumbers $\boldsymbol{k}, \boldsymbol{l} \in \mathbb{N}_0^d$, and $a \in \mathbb{F}_b$, it follows that

$$\langle \boldsymbol{k}, \boldsymbol{0} \rangle = \langle \boldsymbol{0}, \boldsymbol{x} \rangle = 0, \tag{4b}$$

$$\langle \boldsymbol{k}, a\boldsymbol{x} \oplus \boldsymbol{t} \rangle = a \langle \boldsymbol{k}, \boldsymbol{x} \rangle + \langle \boldsymbol{k}, \boldsymbol{t} \rangle \pmod b \quad \text{if ok}(a\boldsymbol{x}, \boldsymbol{t}) \tag{4c}$$

$$\langle a\boldsymbol{k} \oplus \boldsymbol{l}, \boldsymbol{x} \rangle = a \langle \boldsymbol{k}, \boldsymbol{x} \rangle + \langle \boldsymbol{l}, \boldsymbol{x} \rangle \pmod b, \tag{4d}$$

$$\langle \boldsymbol{k}, \boldsymbol{x} \rangle = 0 \; \forall \boldsymbol{k} \in \mathbb{N}_0^d \implies \boldsymbol{x} = \boldsymbol{0}. \tag{4e}$$

The digital sequences $\mathscr{P}_\infty = \{\boldsymbol{z}_i\}_{i=0}^\infty$ considered here are assumed to contain sufficient points so that

$$\langle \boldsymbol{k}, \boldsymbol{z}_i \rangle = 0 \; \forall i \in \mathbb{N}_0 \implies \boldsymbol{k} = \boldsymbol{0}. \tag{5}$$

The *dual net* corresponding to the net $\mathscr{P}_m$ is the set of all wavenumbers for which $\langle \boldsymbol{k}, \cdot \rangle$ maps the whole net to 0:

$$\begin{aligned} \mathscr{P}_m^\perp &:= \{\boldsymbol{k} \in \mathbb{N}_0^d : \langle \boldsymbol{k}, \boldsymbol{z}_i \rangle = 0, \; i \in \mathbb{N}_{0,m}\} \\ &= \{\boldsymbol{k} \in \mathbb{N}_0^d : \langle \boldsymbol{k}, \boldsymbol{z}_{b^\ell} \rangle = 0, \; \ell = 0, \ldots, m-1\}. \end{aligned}$$

The properties of the bilinear transform defined in (4) imply that the dual nets $\mathscr{P}_m^\perp$ are subspaces of each other:

$$\mathscr{P}_0^\perp = \mathbb{N}_0^d \supset \mathscr{P}_1^\perp \supset \cdots \supset \mathscr{P}_\infty^\perp = \{\boldsymbol{0}\}.$$

The integrands are assumed to belong to some subset of $L^2([0,1)^d)$, the space of square integrable functions. The L^2 inner product is defined as

$$\langle f, g \rangle_2 = \int_{[0,1)^d} f(\boldsymbol{x}) \overline{g(\boldsymbol{x})} \, \mathrm{d}\boldsymbol{x}.$$

The Walsh functions $\{\exp(2\pi\sqrt{-1}\,\langle \boldsymbol{k}, \cdot \rangle / b) : \boldsymbol{k} \in \mathbb{N}_0^d\}$ [3, Appendix A] are a complete orthonormal *basis* for $L^2([0,1)^d)$. Thus, any function in L^2 may be written in series form as

$$f(\boldsymbol{x}) = \sum_{\boldsymbol{k} \in \mathbb{N}_0^d} \hat{f}(\boldsymbol{k}) \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{k}, \boldsymbol{x} \rangle / b}, \quad \text{where } \hat{f}(\boldsymbol{k}) := \left\langle f, \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{k}, \cdot \rangle / b} \right\rangle_2, \tag{6}$$

and the L^2 inner product of two functions is the ℓ^2 inner product of their Walsh series coefficients:

$$\langle f, g \rangle_2 = \sum_{\boldsymbol{k} \in \mathbb{N}_0^d} \hat{f}(\boldsymbol{k}) \overline{\hat{g}(\boldsymbol{k})} =: \left\langle \left(\hat{f}(\boldsymbol{k})\right)_{\boldsymbol{k} \in \mathbb{N}_0^d}, \left(\hat{g}(\boldsymbol{k})\right)_{\boldsymbol{k} \in \mathbb{N}_0^d} \right\rangle_2.$$

Since the digital net $\mathscr{P}_m$ is closed under $\oplus$, one may derive a useful formula for the average of a Walsh function sampled over a net. For all wavenumbers $\boldsymbol{k} \in \mathbb{N}_0^d$ and all $\boldsymbol{x} \in \mathscr{P}_m$ one has

$$\begin{aligned} 0 &= \frac{1}{b^m} \sum_{i=0}^{b^m-1} \left[\mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{k}, \boldsymbol{z}_i \rangle / b} - \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{k}, \boldsymbol{z}_i \oplus \boldsymbol{x} \rangle / b}\right] \\ &= \frac{1}{b^m} \sum_{i=0}^{b^m-1} \left[\mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{k}, \boldsymbol{z}_i \rangle / b} - \mathrm{e}^{2\pi\sqrt{-1}\{\langle \boldsymbol{k}, \boldsymbol{z}_i \rangle + \langle \boldsymbol{k}, \boldsymbol{x} \rangle\} / b}\right] \quad \text{by (4c)} \end{aligned}$$

$$= [1 - \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{k},\boldsymbol{x}\rangle/b}] \frac{1}{b^m} \sum_{i=0}^{b^m-1} \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{k},\boldsymbol{z}_i\rangle/b}.$$

By this equality it follows that the average of the sampled Walsh function values is either one or zero, depending on whether the wavenumber is in the dual net or not:

$$\frac{1}{b^m} \sum_{i=0}^{b^m-1} \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{k},\boldsymbol{z}_i\rangle/b} = \mathbb{1}_{\mathscr{P}_m^\perp}(\boldsymbol{k}) = \begin{cases} 1, & \boldsymbol{k} \in \mathscr{P}_m^\perp \\ 0, & \boldsymbol{k} \in \mathbb{N}_0^d \setminus \mathscr{P}_m^\perp. \end{cases} \tag{7}$$

Multivariate integrals may be approximated by the average of the integrand sampled over a digitally shifted digital net, namely,

$$\hat{I}_m(f) := \frac{1}{b^m} \sum_{i=0}^{b^m-1} f(\boldsymbol{z}_i \oplus \boldsymbol{\Delta}). \tag{8}$$

Under the assumption that $\mathrm{ok}(\boldsymbol{z}_i, \boldsymbol{\Delta}) = \text{true}$ (see (2)) for all $i \in \mathbb{N}_0$, it follows that the error of this cubature rule is the sum of the Walsh coefficients of the integrand for those wavenumbers in the dual net:

$$\begin{aligned} \left| \int_{[0,1)^d} f(\boldsymbol{x}) \,\mathrm{d}\boldsymbol{x} - \hat{I}_m(f) \right| &= \left| \hat{f}(\boldsymbol{0}) - \sum_{\boldsymbol{k} \in \mathbb{N}_0^d} \hat{f}(\boldsymbol{k}) \hat{I}_m\left(\mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{k},\cdot\rangle/b}\right) \right| \\ &= \left| \hat{f}(\boldsymbol{0}) - \sum_{\boldsymbol{k} \in \mathbb{N}_0^d} \hat{f}(\boldsymbol{k}) \mathbb{1}_{\mathscr{P}_m^\perp}(\boldsymbol{k}) \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{k},\boldsymbol{\Delta}\rangle/b} \right| \\ &= \left| \sum_{\boldsymbol{k} \in \mathscr{P}_m^\perp \setminus \{\boldsymbol{0}\}} \hat{f}(\boldsymbol{k}) \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{k},\boldsymbol{\Delta}\rangle/b} \right|. \end{aligned} \tag{9}$$

Adaptive Algorithm 2 that we construct in Sect. 4 works with this expression for the cubature error in terms of Walsh coefficients.

Although the true Walsh series coefficients are generally not known, they can be estimated by the discrete Walsh transform, defined as follows:

$$\begin{aligned} \tilde{f}_m(\boldsymbol{k}) &:= \hat{I}_m\left(\mathrm{e}^{-2\pi\sqrt{-1}\langle \boldsymbol{k},\cdot\rangle/b} f(\cdot)\right) = \frac{1}{b^m} \sum_{i=0}^{b^m-1} \mathrm{e}^{-2\pi\sqrt{-1}\langle \boldsymbol{k},\boldsymbol{z}_i \oplus \boldsymbol{\Delta}\rangle/b} f(\boldsymbol{z}_i \oplus \boldsymbol{\Delta}) \\ &= \frac{1}{b^m} \sum_{i=0}^{b^m-1} \left[\mathrm{e}^{-2\pi\sqrt{-1}\langle \boldsymbol{k},\boldsymbol{z}_i \oplus \boldsymbol{\Delta}\rangle/b} \sum_{\boldsymbol{l} \in \mathbb{N}_0^d} \hat{f}(\boldsymbol{l}) \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{l},\boldsymbol{z}_i \oplus \boldsymbol{\Delta}\rangle/b} \right] \\ &= \sum_{\boldsymbol{l} \in \mathbb{N}_0^d} \hat{f}(\boldsymbol{l}) \frac{1}{b^m} \sum_{i=0}^{b^m-1} \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{l} \ominus \boldsymbol{k},\boldsymbol{z}_i \oplus \boldsymbol{\Delta}\rangle/b} \\ &= \sum_{\boldsymbol{l} \in \mathbb{N}_0^d} \hat{f}(\boldsymbol{l}) \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{l} \ominus \boldsymbol{k},\boldsymbol{\Delta}\rangle/b} \frac{1}{b^m} \sum_{i=0}^{b^m-1} \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{l} \ominus \boldsymbol{k},\boldsymbol{z}_i\rangle/b} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{\boldsymbol{l}\in\mathbb{N}_0^d} \hat{f}(\boldsymbol{l}) \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{l}\ominus\boldsymbol{k},\boldsymbol{\Delta}\rangle/b} \mathbb{1}_{\mathscr{P}_m^\perp}(\boldsymbol{l}\ominus\boldsymbol{k}) \\
&= \sum_{\boldsymbol{l}\in\mathscr{P}_m^\perp} \hat{f}(\boldsymbol{k}\oplus\boldsymbol{l}) \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{l},\boldsymbol{\Delta}\rangle/b} \\
&= \hat{f}(\boldsymbol{k}) + \sum_{\boldsymbol{l}\in\mathscr{P}_m^\perp\setminus\{\boldsymbol{0}\}} \hat{f}(\boldsymbol{k}\oplus\boldsymbol{l}) \mathrm{e}^{2\pi\sqrt{-1}\langle \boldsymbol{l},\boldsymbol{\Delta}\rangle/b}, \qquad \forall \boldsymbol{k}\in\mathbb{N}_0^d. \qquad (10)
\end{aligned}
$$

The discrete transform, $\tilde{f}_m(\boldsymbol{k})$ is equal to the true Walsh transform, $\hat{f}(\boldsymbol{k})$, plus *aliasing* terms proportional to $\hat{f}(\boldsymbol{k}\oplus\boldsymbol{l})$ where $\boldsymbol{l}$ is a nonzero wavenumber in the dual net.

4 Error Estimation and an Adaptive Cubature Algorithm

4.1 Wavenumber Map

Since the discrete Walsh transform has aliasing errors, some assumptions must be made about how quickly the true Walsh coefficients decay and which coefficients are more important. This is done by way of a map of the non-negative integers *onto* the space of all wavenumbers, $\tilde{\boldsymbol{k}}:\mathbb{N}_0\to\mathbb{N}_0^d$, according to the following algorithm.

Algorithm 1. Given a digital sequence, $\mathscr{P}_\infty=\{\boldsymbol{z}_i\}_{i=0}^\infty$ define $\tilde{\boldsymbol{k}}:\mathbb{N}_0\to\mathbb{N}_0^d$ as follows:

Step 1. Define $\tilde{\boldsymbol{k}}(0)=\boldsymbol{0}$.

Step 2. For $m=0,1,\ldots$

For $\kappa=1,\ldots,b^m-1$

Choose the values of $\tilde{\boldsymbol{k}}(\kappa+b^m),\ldots,\tilde{\boldsymbol{k}}(\kappa+(b-1)b^m)$ from the sets

$$\{\boldsymbol{k}:\boldsymbol{k}\ominus\tilde{\boldsymbol{k}}(\kappa)\in\mathscr{P}_m^\perp,\langle\boldsymbol{k}\ominus\tilde{\boldsymbol{k}}(\kappa),\boldsymbol{z}_{b^m}\rangle=a\},\quad a=1,\ldots,b-1,$$

but not necessarily in that order.

There is some flexibility in the choice of this map. One might choose $\tilde{\boldsymbol{k}}$ to map smaller values of κ to smaller values of $\boldsymbol{k}$ based on some standard measure of size such as that given in [3, (5.9)]. The motivation is that larger κ should generally lead to smaller $\hat{f}(\tilde{\boldsymbol{k}}(\kappa))$. We use Algorithm 3 below to construct this map implicitly.

Introducing the shorthand notation $\hat{f}_\kappa:=\hat{f}(\tilde{\boldsymbol{k}}(\kappa))$ and $\tilde{f}_{m,\kappa}:=\tilde{f}_m(\tilde{\boldsymbol{k}}(\kappa))$, the aliasing relation (10) may be written as

$$\tilde{f}_{m,\kappa}=\hat{f}_\kappa+\sum_{\lambda=1}^\infty \hat{f}_{\kappa+\lambda b^m}\mathrm{e}^{2\pi\sqrt{-1}\langle\tilde{\boldsymbol{k}}(\kappa+\lambda b^m)\ominus\tilde{\boldsymbol{k}}(\kappa),\boldsymbol{\Delta}\rangle/b}, \qquad (11)$$

and the cubature error in (9) may be bounded as

$$\left|\int_{[0,1)^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}-\hat{I}_m(f)\right|=\left|\sum_{\lambda=1}^\infty \hat{f}_{\lambda b^m}\mathrm{e}^{2\pi\sqrt{-1}\langle\tilde{\boldsymbol{k}}(\lambda b^m),\boldsymbol{\Delta}\rangle}/b\right|\le\sum_{\lambda=1}^\infty\left|\hat{f}_{\lambda b^m}\right|. \qquad (12)$$

We will use the discrete transform, $\tilde{f}_{m,\kappa}$, to estimate true Walsh coefficients, $\hat{f}_\kappa$, for m is significantly larger than $\lfloor \log_b(\kappa) \rfloor$.

4.2 Sums of Walsh Series Coefficients and Cone Conditions

Consider the following sums of the true and approximate Walsh series coefficients. For $\ell, m \in \mathbb{N}_0$ and $\ell \le m$ let

$$S_m(f) = \sum_{\kappa=\lfloor b^{m-1} \rfloor}^{b^m-1} \bigl|\hat{f}_\kappa\bigr|, \qquad \widehat{S}_{\ell,m}(f) = \sum_{\kappa=\lfloor b^{\ell-1} \rfloor}^{b^\ell-1} \sum_{\lambda=1}^{\infty} \bigl|\hat{f}_{\kappa+\lambda b^m}\bigr|,$$

$$\check{S}_m(f) = \widehat{S}_{0,m}(f) + \cdots + \widehat{S}_{m,m}(f) = \sum_{\kappa=b^m}^{\infty} \bigl|\hat{f}_\kappa\bigr|, \qquad \widetilde{S}_{\ell,m}(f) = \sum_{\kappa=\lfloor b^{\ell-1} \rfloor}^{b^\ell-1} \bigl|\tilde{f}_{m,\kappa}\bigr|.$$

The first three sums, $S_m(f)$, $\widehat{S}_{\ell,m}(f)$, and $\check{S}_m(f)$, cannot be observed because they involve the true series coefficients. But, the last sum, $\widetilde{S}_{\ell,m}(f)$, is defined in terms of the discrete Walsh transform and can easily be computed in terms of function values. The details are described in the Appendix.

We now make critical assumptions about how certain sums provide upper bounds on others. Let $\ell_* \in \mathbb{N}$ be some fixed integer and $\widehat{\omega}$ and $\check{\omega}$ be some non-negative valued functions with $\lim_{m\to\infty} \check{\omega}(m) = 0$. Define the cone of integrands

$$\mathscr{C} := \{f \in L^2([0,1)^d) : \widehat{S}_{\ell,m}(f) \le \widehat{\omega}(m-\ell)\check{S}_m(f), \ \ \ell \le m, \\ \check{S}_m(f) \le \check{\omega}(m-\ell) S_\ell(f), \ \ \ell_* \le \ell \le m\}. \quad (13)$$

This is a cone because $f \in \mathscr{C} \implies af \in \mathscr{C}$ for all real a.

The first inequality asserts that the sum of the larger indexed Walsh coefficients bounds a partial sum of the same coefficients. For example, this means that $\widehat{S}_{0,12}$, the sum of the values of the large black dots in Figure 2, is no greater than some factor times $\check{S}_{12}(f)$, the sum of the values of the darker gray smaller dots. Choosing $\widehat{\omega}(m) := 1$ works, but one might also consider choosing $\widehat{\omega}(m) = Cb^{-\alpha m}$ for some $C > 1$ and $0 \le \alpha \le 1$. The second inequality asserts that the sum of the smaller indexed coefficients provides an upper bound on the sum of the larger indexed coefficients. In other words, the fine scale components of the integrand are not unduly large compared to the gross scale components. In Figure 2 this means that $\check{S}_{12}(f)$ is no greater than some factor times $S_8(f)$, the sum of the values of the large black squares. This implies that $\bigl|\hat{f}_\kappa\bigr|$ does not dip down and then bounce back up too dramatically as $\kappa \to \infty$. The reason for enforcing the second inequality only for $\ell \ge \ell_*$ is that for small ℓ, one might have a coincidentally small $S_\ell(f)$, while $\check{S}_m(f)$ is large.

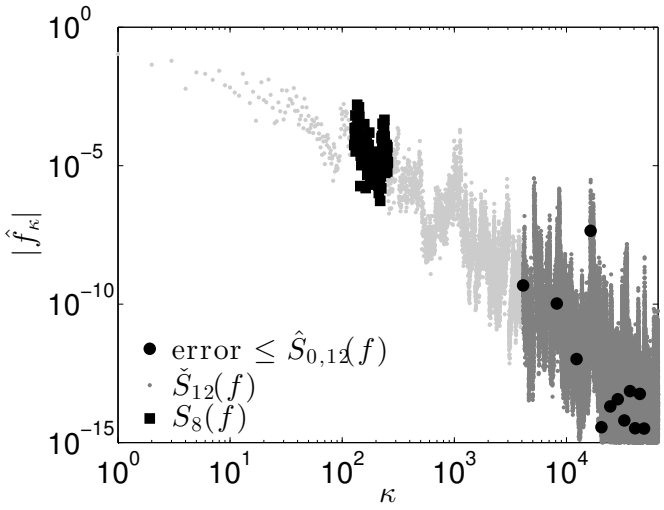

Fig. 2 The magnitudes of true Walsh coefficients for some integrand.

The cubature error bound in (12) can be bounded in terms of $S_l(f)$, a certain finite sum of the Walsh coefficients for integrands f in the cone $\mathscr{C}$. For $\ell, m \in \mathbb{N}$, $\ell_* \leq \ell \leq m$, it follows that

$$\begin{aligned}\left|\int_{[0,1)^d} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} - \hat{I}_m(f)\right| &\leq \sum_{\lambda=1}^{\infty} \left|\hat{f}_{\lambda b^m}\right| = \widehat{S}_{0,m}(f) \qquad \text{by (12)} \\ &\leq \widehat{\omega}(m)\check{S}_m(f) \leq \widehat{\omega}(m)\mathring{\omega}(m-\ell)S_\ell(f). \end{aligned} \tag{14}$$

Thus, the faster $S_\ell(f)$ decays as $\ell \to \infty$, the faster the cubature error must decay.

Unfortunately, the true Walsh coefficients are unknown. Thus we must bound $S_\ell(f)$ in terms of the observable sum of the approximate coefficients, $\widetilde{S}_{\ell,m}(f)$. This is done as follows:

$$\begin{aligned} S_\ell(f) &= \sum_{\kappa=b^{\ell-1}}^{b^\ell-1} \left|\hat{f}_\kappa\right| \\ &= \sum_{\kappa=b^{\ell-1}}^{b^\ell-1} \left|\tilde{f}_{m,\kappa} - \sum_{\lambda=1}^{\infty} \hat{f}_{\kappa+\lambda b^m} \mathrm{e}^{2\pi\sqrt{-1}\left\langle \tilde{\boldsymbol{k}}(\kappa+\lambda b^m)\ominus\tilde{\boldsymbol{k}}(\kappa), \boldsymbol{\Delta}\right\rangle/b}\right| \qquad \text{by (11)} \\ &\leq \sum_{\kappa=b^{\ell-1}}^{b^\ell-1} \left|\tilde{f}_{m,\kappa}\right| + \sum_{\kappa=b^{\ell-1}}^{b^\ell-1}\sum_{\lambda=1}^{\infty} \left|\hat{f}_{\kappa+\lambda b^m}\right| = \widetilde{S}_{\ell,m}(f) + \widehat{S}_{\ell,m}(f) \\ &\leq \widetilde{S}_{\ell,m}(f) + \widehat{\omega}(m-\ell)\mathring{\omega}(m-\ell)S_\ell(f) \qquad \text{by (13)}, \end{aligned}$$

$$S_\ell(f) \leq \frac{\widetilde{S}_{\ell,m}(f)}{1-\widehat{\omega}(m-\ell)\mathring{\omega}(m-\ell)} \qquad \text{provided that } \widehat{\omega}(m-\ell)\mathring{\omega}(m-\ell) < 1. \tag{15}$$

Combining (14) with (15) leads to the following conservative upper bound on the cubature error for $\ell, m \in \mathbb{N}$, $\ell_* \le \ell \le m$:

$$\left| \int_{[0,1)^d} f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} - \hat{I}_m(f) \right| \le \frac{\widetilde{S}_{\ell,m}(f) \widehat{\omega}(m) \mathring{\omega}(m-\ell)}{1 - \widehat{\omega}(m-\ell)\mathring{\omega}(m-\ell)}. \tag{16}$$

This error bound suggests the following algorithm.

4.3 An Adaptive Cubature Algorithm and Its Cost

Algorithm 2 (Adaptive Digital Sequence Cubature, `cubSobol_g`). Given the parameter $\ell_* \in \mathbb{N}$ and the functions $\widehat{\omega}$ and $\mathring{\omega}$ that define the cone $\mathscr{C}$ in (13), choose the parameter $r \in \mathbb{N}$ such that $\widehat{\omega}(r)\mathring{\omega}(r) < 1$. Let $\mathfrak{C}(m) := \widehat{\omega}(m)\mathring{\omega}(r)/[1 - \widehat{\omega}(r)\mathring{\omega}(r)]$ and $m = \ell_* + r$. Given a tolerance, ε, and a routine that produces values of the integrand, f, do the following:

Step 1. Compute the sum of the discrete Walsh coefficients, $\widetilde{S}_{m-r,m}(f)$ according to Algorithm 3.

Step 2. Check whether the error tolerance is met, i.e., whether $\mathfrak{C}(m)\widetilde{S}_{m-r,m}(f) \le \varepsilon$. If so, then return the cubature $\hat{I}_m(f)$ defined in (8) as the answer.

Step 3. Otherwise, increment m by one, and go to Step 1.

There is a balance to be struck in the choice of r. Choosing r too large causes the error bound to depend on the Walsh coefficients with smaller indices, but choosing r too large makes $\mathfrak{C}$ large.

Theorem 1. *If the integrand, f, lies in the cone, $\mathscr{C}$, then the Algorithm 2 is successful:*

$$\left| \int_{[0,1)^d} f(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} - \hat{I}_m(f) \right| \le \varepsilon.$$

The number of integrand values required to obtain this answer is 2^m, where the following upper bound on m depends on the tolerance and unknown decay rate of the Walsh coefficients.

$$m \le \min\{m' \ge \ell_* + r : \mathfrak{C}(m')[1 + \widehat{\omega}(r)\mathring{\omega}(r)]S_{m'-r}(f) \le \varepsilon\}$$

The computational cost of this algorithm beyond that of obtaining the integrand values is $\mathcal{O}(mb^m)$ to compute the discrete Walsh transform.

Proof. The success of this algorithm comes from applying (16). To bound the number of integrand values required note that argument leading to (15) can be modified to provide an upper bound on $\widetilde{S}_{\ell,m}(f)$ in terms of $S_\ell(f)$:

$$\widetilde{S}_{\ell,m}(f) = \sum_{\kappa=b^{\ell-1}}^{b^\ell-1} \left| \tilde{f}_{m,\kappa} \right|$$

$$
\begin{aligned}
&= \sum_{\kappa=b^{\ell-1}}^{b^\ell-1} \left| \hat{f}_\kappa + \sum_{\lambda=1}^{\infty} \hat{f}_{\kappa+\lambda b^m} \mathrm{e}^{2\pi\sqrt{-1}\left\langle \tilde{\boldsymbol{k}}(\kappa+\lambda b^m) \ominus \tilde{\boldsymbol{k}}(\kappa), \boldsymbol{\Delta} \right\rangle / b} \right| \qquad \text{by (11)} \\
&\le \sum_{\kappa=b^{\ell-1}}^{b^\ell-1} \left| \hat{f}_\kappa \right| + \sum_{\kappa=b^{\ell-1}}^{b^\ell-1} \sum_{\lambda=1}^{\infty} \left| \hat{f}_{\kappa+\lambda b^m} \right| = S_\ell(f) + \widehat{S}_{\ell,m}(f) \\
&\le [1 + \widehat{\omega}(m-\ell)\mathring{\omega}(m-\ell)] S_\ell(f) \qquad \text{by (13).}
\end{aligned}
$$

Thus, the upper bound on the error in Step 2 of Algorithm 2, is itself bounded above by $\mathfrak{C}(m)[1+\widehat{\omega}(r)\mathring{\omega}(r)]S_{m-r}(f)$. Therefore, the stopping criterion in Step 2 must be satisfied no later than when this quantity falls below ε.

The computation of the discrete Walsh transform and $\widetilde{S}_{m-r,m}(f)$ is described in Algorithm 3 in the Appendix. The cost of this algorithm is $\mathcal{O}(mb^m)$ operations. □

5 Numerical Experiments

Algorithm 2 has been implemented in MATLAB code as the function `cubSobol_g`. It is intended to be included in the next release of our Guaranteed Automatic Integration Library (GAIL) [2]. Our `cubSobol_g` utilizes MATLAB's built-in Sobol' sequences, so $b = 2$. The default algorithm parameters are

$$
\ell_* = 6, \qquad r = 4, \qquad \mathfrak{C}(m) = 5 \times 2^{-m}.
$$

We have tried `cubSobol_g` on an example from [10]:

$$
I = \int_{\mathbb{R}^d} \mathrm{e}^{-\|\boldsymbol{t}\|^2} \cos(\|\boldsymbol{t}\|) \, \mathrm{d}\boldsymbol{t} = \pi^{d/2} \int_{[0,1)^d} \cos\left(\sqrt{\frac{1}{2} \sum_{j=1}^{d} [\Phi^{-1}(x_j)]^2} \right) \mathrm{d}\boldsymbol{x}, \qquad (17)
$$

where Φ is the standard Gaussian distribution function. We generated 1000 IID random values of the dimension $d = \lfloor \mathrm{e}^D \rfloor$ with D being uniformly distributed between 0 and $\log(20)$. Each time `cubSobol_g` was run, a different randomized Sobol' sequence was used. The tolerance was met about 97% of the time. Failures were more likely among the higher dimensions.

6 Discussion

There are few quasi-Monte Carlo cubature algorithms available that adaptively determine the sample size needed based on integrand values. The chief reason is that reliable error estimation for quasi-Monte Carlo is difficult. Quasi-standard error has serious drawbacks, as explained in [15]. Internal replications have no explicit theory.

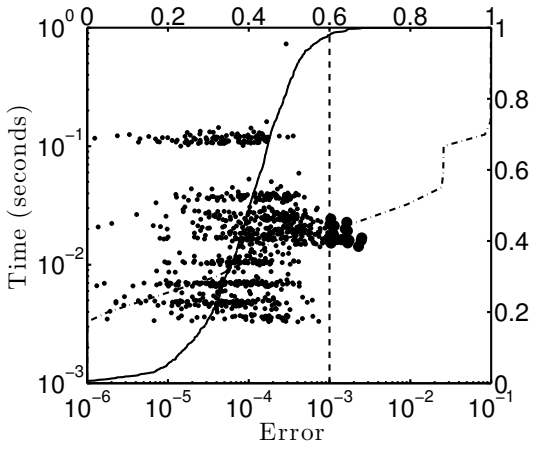

Fig. 3 Time required and error observed for `cubSobol_g` (Algorithm 2) for the Keister example, (17). Small dots denote the time and error when the tolerance of $\varepsilon = 0.001$ was met. Large dots denote the time and error when the tolerance was not met. The solid line denotes the empirical distribution function of the error, and the dot-dashed line denotes the empirical distribution function of the time.

IID replications of randomized quasi-Monte Carlo rules are sometimes used, but one does not know how many replications are needed.

The proposed error bound and adaptive algorithm here are practical and have theoretical justification. By focusing on a non-convex cones of integrands, as defined in (13), we put ourselves in a setting where adaption has the opportunity to be beneficial.

Problems requiring further consideration include how to choose the default parameters for Algorithm 2. We would also like to extend our algorithm and theory to the case of relative error.

Acknowledgements This work was partially supported by US National Science Foundation grants DMS-1115392 and DMS-1357690. The authors thank Ronald Cools and Dirk Nuyens for organizing MCQMC 2014. We thank Sergei Kucherenko and Art Owen and for organizing the special session in honor of Ilya M. Sobol'. We are grateful for Professor Sobol's many contributions to MCQMC and related fields. The suggestions made by Sou-Cheng Choi, Yuhan Ding, and Lan Jiang to improve this manuscript are greatly appreciated.

References

1. Caflisch, R.E.: Monte Carlo and quasi-Monte Carlo methods. Acta Numer. **7**, 1–49 (1998). DOI 10.1017/S0962492900002804
2. Choi, S.C.T., Ding, Y., Hickernell, F.J., Jiang, L., Zhang, Y.: GAIL: Guaranteed Automatic Integration Library (versions 1, 1.3). MATLAB software (2013–2014). URL http://code.google.com/p/gail

3. Dick, J., Pillichshammer, F.: Digital Nets and Sequences: Discrepancy Theory and Quasi-Monte Carlo Integration. Cambridge University Press, Cambridge (2010)
4. Hickernell, F.J.: A generalized discrepancy and quadrature error bound. Math. Comp. **67**, 299–322 (1998). DOI 10.1090/S0025-5718-98-00894-1
5. Hickernell, F.J., Sloan, I.H., Wasilkowski, G.W.: On strong tractability of weighted multivariate integration. Math. Comp. **73**, 1903–1911 (2004). DOI 10.1090/S0025-5718-04-01653-9
6. Hickernell, F.J., Sloan, I.H., Wasilkowski, G.W.: On tractability of weighted integration for certain Banach spaces of functions. In: Niederreiter [13], pp. 51–71
7. Hickernell, F.J., Sloan, I.H., Wasilkowski, G.W.: On tractability of weighted integration over bounded and unbounded regions in $\mathbb{R}^s$. Math. Comp. **73**, 1885–1901 (2004). DOI 10.1090/S0025-5718-04-01624-2
8. Hickernell, F.J., Sloan, I.H., Wasilkowski, G.W.: The strong tractability of multivariate integration using lattice rules. In: Niederreiter [13], pp. 259–273
9. Jiménez Rugama, Ll.A., Hickernell, F.J.: Adaptive multidimensional integration based on rank-1 lattices (2014). In preparation
10. Keister, B.D.: Multidimensional quadrature algorithms. Computers in Physics **10**, 119–122 (1996). DOI 10.1063/1.168565
11. Lemieux, C.: Monte Carlo and quasi-Monte Carlo Sampling. Springer Science+Business Media, Inc., New York (2009)
12. Niederreiter, H.: Random Number Generation and Quasi-Monte Carlo Methods. CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia (1992)
13. Niederreiter, H. (ed.): Monte Carlo and Quasi-Monte Carlo Methods 2002. Springer-Verlag, Berlin (2004)
14. Novak, E., Woźniakowski, H.: Tractability of Multivariate Problems Volume II: Standard Information for Functionals. No. 12 in EMS Tracts in Mathematics. European Mathematical Society, Zürich (2010)
15. Owen, A.B.: On the Warnock-Halton quasi-standard error. Monte Carlo Methods and Appl. **12**, 47–54 (2006). DOI 10.1515/156939606776886652

Appendix: Fast Computation of the Discrete Walsh Transform

Let $y_0, y_1, \ldots$ be some data. Define $Y_\nu^{(m)}$ for $\nu = 0, \ldots, b^m - 1$ as follows:

$$Y_\nu^{(m)} := \frac{1}{b^m} \sum_{i=0}^{b^m-1} \mathrm{e}^{-2\pi\sqrt{-1}\sum_{\ell=0}^{m-1} \nu_\ell i_\ell / b} y_i = \frac{1}{b^m} \sum_{i_{m-1}=0}^{b-1} \cdots \sum_{i_0=0}^{b-1} \mathrm{e}^{-2\pi\sqrt{-1}\sum_{\ell=0}^{m-1} \nu_\ell i_\ell / b} y_i,$$

where $i = i_0 + i_1 b + \cdots i_{m-1} b^{m-1}$ and $\nu = \nu_0 + \nu_1 b + \cdots \nu_{m-1} b^{m-1}$. For all $i_j, \nu_j \in \mathbb{F}_b$, $j, \ell = 0, \ldots, m-1$, recursively define

$$Y_{m,0}(i_0, \ldots, i_{m-1}) := y_i,$$

$$\begin{aligned} &Y_{m,\ell+1}(\nu_0, \ldots, \nu_\ell, i_{\ell+1}, \ldots, i_{m-1}) \\ &\qquad := \frac{1}{b} \sum_{i_\ell=0}^{b-1} \mathrm{e}^{-2\pi\sqrt{-1}\nu_\ell i_\ell / b} Y_{m,\ell}^{(m)}(\nu_1, \ldots, \nu_{\ell-1}, i_\ell, \ldots, i_{m-1}). \end{aligned}$$

This allows us to identify $Y_{\nu}^{(m)} = Y_{m,m}(\nu_0,\dots,\nu_{m-1})$. By this iterative process one can compute $Y_0^{(m)},\dots,Y_{b^m-1}^{(m)}$ in only $\mathcal{O}(mb^m)$ operations.

Note also, that $Y_{m+1,m}(\nu_0,\dots,\nu_{m-1},0) = Y_{m,m}(\nu_0,\dots,\nu_{m-1}) = Y_{\nu}^{(m)}$. This means that the work done to compute $Y_{\nu}^{(m)}$ can be used to compute $Y_{\nu}^{(m+1)}$.

Next, we relate the Y_{ν} to the discrete Walsh transform of the integrand f. For every $\boldsymbol{k} \in \mathbb{N}_0^d$ and every digital sequence $\mathscr{P}_\infty = \{\boldsymbol{z}_i\}_{i=0}^\infty$, let

$$\widetilde{\nu}_0(\boldsymbol{k}) := 0, \qquad \widetilde{\nu}_m(\boldsymbol{k}) := \sum_{\ell=0}^{m-1} \langle \boldsymbol{k}, \boldsymbol{z}_{b^\ell} \rangle \, b^\ell \in \mathbb{N}_{0,m}, \quad m \in \mathbb{N}. \tag{18}$$

If we set $y_i = f(\boldsymbol{z}_i + \boldsymbol{\Delta})$, and if $\widetilde{\nu}_m(\boldsymbol{k}) = \nu$, then

$$\begin{aligned}
\tilde{f}_m(\boldsymbol{k}) &= \frac{1}{b^m} \sum_{i=0}^{b^m-1} \mathrm{e}^{-2\pi\sqrt{-1}\langle \boldsymbol{k}, \boldsymbol{z}_i \oplus \boldsymbol{\Delta}\rangle/b} y_i \\
&= \frac{\mathrm{e}^{-2\pi\sqrt{-1}\langle \boldsymbol{k}, \boldsymbol{\Delta}\rangle/b}}{b^m} \sum_{i=0}^{b^m-1} \mathrm{e}^{-2\pi\sqrt{-1}\langle \boldsymbol{k}, \boldsymbol{z}_i\rangle/b} y_i \qquad \text{by (4c)} \\
&= \frac{\mathrm{e}^{-2\pi\sqrt{-1}\langle \boldsymbol{k}, \boldsymbol{\Delta}\rangle/b}}{b^m} \sum_{i=0}^{b^m-1} \mathrm{e}^{-2\pi\sqrt{-1}\left\langle \boldsymbol{k}, \sum_{j=0}^{m-1} i_j \boldsymbol{z}_{b^j}\right\rangle/b} y_i \qquad \text{by (3)} \\
&= \frac{\mathrm{e}^{-2\pi\sqrt{-1}\langle \boldsymbol{k}, \boldsymbol{\Delta}\rangle/b}}{b^m} \sum_{i=0}^{b^m-1} \mathrm{e}^{-2\pi\sqrt{-1}\sum_{j=0}^{m-1} i_j \langle \boldsymbol{k}, \boldsymbol{z}_{b^j}\rangle/b} y_i \qquad \text{by (4c)} \\
&= \frac{\mathrm{e}^{-2\pi\sqrt{-1}\langle \boldsymbol{k}, \boldsymbol{\Delta}\rangle/b}}{b^m} \sum_{i=0}^{b^m-1} \mathrm{e}^{-2\pi\sqrt{-1}\sum_{\ell=0}^{m-1} \nu_\ell i_\ell/b} y_i \qquad \text{by (18)} \\
&= \mathrm{e}^{-2\pi\sqrt{-1}\langle \boldsymbol{k}, \boldsymbol{\Delta}\rangle/b} Y_{\nu}^{(m)}.
\end{aligned} \tag{19}$$

Using the notation in Sect. 4, for all $m \in \mathbb{N}_0$ define a pointer $\mathring{\nu}_m : \mathbb{N}_{0,m} \to \mathbb{N}_{0,m}$ as $\mathring{\nu}_m(\kappa) := \widetilde{\nu}_m(\tilde{\boldsymbol{k}}(\kappa))$. It follows that

$$\begin{aligned}
\tilde{f}_{m,\kappa} &= \tilde{f}_m(\tilde{\boldsymbol{k}}(\kappa)) = \mathrm{e}^{-2\pi\sqrt{-1}\langle \boldsymbol{k}, \boldsymbol{\Delta}\rangle/b} Y_{\mathring{\nu}_m(\kappa)}^{(m)}, \\
\widetilde{S}_{\ell,m}(f) &= \sum_{\kappa=b^{\ell-1}}^{b^\ell-1} \left|\tilde{f}_{m,\kappa}\right| = \sum_{\kappa=b^{\ell-1}}^{b^\ell-1} \left|Y_{\mathring{\nu}_m(\kappa)}^{(m)}\right|.
\end{aligned} \tag{20}$$

The quantity $\widetilde{S}_{m-r,m}(f)$ is the key to the stopping criterion in Algorithm 2.

If the map $\tilde{\boldsymbol{k}} : \mathbb{N}_0 \to \mathbb{N}_0^d$ defined in Algorithm 1 is known explicitly, then specifying $\mathring{\nu}_m$ is straightforward. However, in practice the bookkeeping involved in constructing $\tilde{\boldsymbol{k}}$ might be tedious, so we take a data-dependent approach to constructing the pointer $\mathring{\nu}_m(\kappa)$ for $\kappa \in \mathbb{N}_{0,m}$ directly, which then defines $\tilde{\boldsymbol{k}}$ implicitly.

Algorithm 3. Let $r \in \mathbb{N}$ be fixed. Given the input $m \in \mathbb{N}_0$, the discrete Walsh coefficients $Y_{\nu}^{(m)}$ for $\nu \in \mathbb{N}_{0,m}$, and also the pointer $\mathring{\nu}_{m-1}(\kappa)$ defined for $\kappa \in \mathbb{N}_{0,m-1}$, provided $m > 0$,

Step 1. If $m = 0$, then define $\mathring{\nu}(0) = 0$ and go to Step 4.

Step 2. Otherwise, if $m \geq 1$, then initialize $\mathring{\nu}_m(\kappa) = \mathring{\nu}_{m-1}(\kappa)$ for $\kappa \in \mathbb{N}_{0,m-1}$ and $\mathring{\nu}_m(\kappa) = \kappa$ for $\kappa = b^{m-1}, \ldots, b^m - 1$.

Step 3. For $\ell = m-1, m-2, \ldots, \max(1, m-r)$,
for $\kappa = 1, \ldots, b^\ell - 1$
Find a^* such that $\left|Y^{(m)}_{\mathring{\nu}_m(\kappa + a^* b^\ell)}\right| \geq \left|Y^{(m)}_{\mathring{\nu}_m(\kappa + a b^\ell)}\right|$ for all $a \in \mathbb{F}_b$.
Swap the values of $\mathring{\nu}_m(\kappa)$ and $\mathring{\nu}_m(\kappa + a^* b^\ell)$.

Step 4. Return $\mathring{\nu}_m(\kappa)$ for $\kappa \in \mathbb{N}_{0,m}$. If $m \geq r$, then compute $\widetilde{S}_{m-r,r}(f)$ according to (20), and return this value as well.

Lemma 1. *Let* $\mathscr{P}^\perp_{m,\kappa} := \{\boldsymbol{k} \in \mathbb{N}_0^d : \widetilde{\nu}_m(\boldsymbol{k}) = \mathring{\nu}_m(\kappa)\}$ *for* $\kappa \in \mathbb{N}_{0,m}$, $m \in \mathbb{N}_0$, *where* $\mathring{\nu}_m$ *is given by Algorithm 3. Then we implicitly have defined the map* $\tilde{\boldsymbol{k}}$ *in the sense that any map* $\tilde{\boldsymbol{k}} : \mathbb{N}_{0,m} \to \mathbb{N}_0^d$ *that chooses* $\tilde{\boldsymbol{k}}(0) = \boldsymbol{0} \in \mathscr{P}^\perp_{m,0}$, *and* $\tilde{\boldsymbol{k}}(\kappa) \in \mathscr{P}_{m,\kappa}$ *for all* $\kappa = 1, \ldots, b^m - 1$ *gives the same value of* $S_{m-r,r}(f)$. *It is also consistent with Algorithm 1 for* $\kappa \in \mathbb{N}_{0,m-r}$.

Proof. The constraint that $\tilde{\boldsymbol{k}}(\kappa) \in \mathscr{P}_{m,\kappa}$ implies that $S_{m-r,r}(f)$ is invariant under all $\tilde{\boldsymbol{k}}$ chosen according to the assumption that $\tilde{\boldsymbol{k}}(\kappa) \in \mathscr{P}_{m,\kappa}$. By definition $\boldsymbol{0} \in \mathscr{P}^\perp_{m,0}$ remains true for all m for Algorithm 3.

The remainder of the proof is to show that choosing $\tilde{\boldsymbol{k}}(\kappa)$ by the hypothesis of this lemma is consistent with Algorithm 1. To do this we show that for $m \in N_0$

$$\boldsymbol{k} \in \mathscr{P}^\perp_{m,\kappa},\ \boldsymbol{l} \in \mathscr{P}^\perp_{m,\kappa + a b^\ell} \implies \boldsymbol{k} \ominus \boldsymbol{l} \in \mathscr{P}^\perp_\ell \qquad \text{for all } \kappa = 1, \ldots, b^\ell, \ell < m, \quad (21)$$

and that

$$\mathscr{P}^\perp_{m,\kappa} \supset \mathscr{P}^\perp_{m+1,\kappa} \supset \cdots \qquad \text{for } \kappa \in \mathbb{N}_{0,m-r} \text{ provided } m \geq r. \quad (22)$$

The proof proceeds by induction. Since $\mathscr{P}^\perp_{0,0} = \mathbb{N}_0^d$, the above two conditions are satisfied automatically.

If they are satisfied for $m-1$ (instead of m), then the initialization stage in Step 2 of Algorithm 3 preserves (21) for m. The swapping of $\mathring{\nu}_m(\kappa)$ and $\mathring{\nu}_m(\kappa + a^* b^\ell)$ values in Step 3 also preserves (21). Step 3 may cause $\mathscr{P}^\perp_{m-1,\kappa} \cap \mathscr{P}^\perp_{m,\kappa} = \emptyset$ for some larger values of κ, but the constraint on the values of ℓ in Step 3 mean that (22) is preserved. □