Automatic Bayesian Cubature manuscript No.
(will be inserted by the editor)

Fast Automatic Bayesian Cubature Using Sobol Sampling

R. Jagadeeswaran · Fred J. Hickernell

Received: date / Accepted: date

Abstract Automatic cubatures approximate integrals to user-specified error tolerances. For high dimensional problems, it is difficult to adaptively change the sampling pattern, but one can automatically determine the sample size, n, given a reasonable, fixed sampling pattern. We take this approach here using a Bayesian perspective. We postulate that the integrand is an instance of a Gaussian stochastic process parameterized by a constant mean and a covariance kernel defined by a scale parameter times a parameterized function specifying how the integrand values at two different points in the domain are related. These hyperparameters are inferred or integrated out using integrand values via one of three techniques: empirical Bayes, full Bayes, or generalized cross-validation. The sample size, n, is increased until the half-width of the credible interval for the Bayesian posterior mean is no greater than the error tolerance.

The process outlined above typically requires a computational cost of $O(N_{\text{opt}} n^3)$, where N_{opt} is the number of optimization steps required to identify the hyperparameters. Our innovation is to pair low discrepancy nodes with matching covariance kernels to lower the computational cost to $O(N_{\text{opt}} n \log n)$. This approach is demonstrated explicitly with rank-1 lattice sequences and shift-invariant kernels. Our algorithm is implemented in the Guaranteed Automatic Integration Library (GAIL).

Keywords Bayesian cubature · Fast automatic cubature · GAIL · Probabilistic numeric methods

R. Jagadeeswaran
Department of Applied Mathematics,
Illinois Institute of Technology
10 W. 32nd St., Room 208, Chicago IL 60616
E-mail: jrathin1@iit.edu

Fred J. Hickernell
Center for Interdisciplinary Scientific Computation and
Department of Applied Mathematics
Illinois Institute of Technology
10 W. 32nd St., Room 208, Chicago IL 60616
E-mail: hickernell@iit.edu

1 Introduction

Your text comes here. Separate text sections with

2 Sobol' Nets and Walsh Kernels

The previous section shows an automatic Bayesian cubature algorithm using rank-1 lattice nodes and shift-invariant kernels. In this chapter, we demonstrate a second approach to formulate fast Bayesian transform using matching kernel and point sets. Scrambled Sobol' nets and Walsh kernels are paired to achieve $\mathcal{O}(n^{-1+\epsilon})$ order error convergence where n is the sample size. Sobol' nets **?** are low discrepancy points, used extensively in numerical integration, simulation, and optimization. The results of this chapter can be summarized as a theorem,

Theorem 1 *Any symmetric, positive definite, digital shift-invariant covariance kernel of the form* (5) *scaled to satisfy* (**??**)*, when matched with digital net data-sites, satisfies assumptions* (**??**)*. The* fast Walsh-Hadamard transform *(FWHT) can be used to expedite the estimates of $\boldsymbol{\theta}$ in* (**??**) *and the credible interval widths* (**??**) *in $\mathcal{O}(n \log n)$ operations. The cubature, $\widehat{\mu}$, is just the sample mean.*

We introduce the necessary concepts and prove this theorem in the remaining of this chapter.

3 Sobol' Nets

Nets were developed to provide deterministic sample points for quasi-Monte Carlo rules **?**. Nets are defined geometrically using elementary intervals, which are subintervals of the unit cube $[0,1)^d$. The (t,m,d)-nets in base b, introduced by Niederreiter, whose quality is governed by t. Lower values of t correspond to (t,m,d)-nets of higher quality **?**.

Definition 1 *Let $\mathcal{A}$ be the set of all elementary intervals $\mathcal{A} \subset [0,1)^d$ where $\mathcal{A} = \prod_{\ell=1}^{d} [\alpha_\ell b^{-\gamma_\ell}, (\alpha_\ell + 1) b^{-\gamma_\ell})$, with $d, b, \gamma_\ell \in \mathbb{N}, b \geq 2$ and $b^{\gamma_\ell} > \alpha_\ell \geq 0$. For $m, t \in \mathbb{N}, m \geq t \geq 0$, the point set $\mathcal{P}_m \in [0,1)^d$ with $n = b^m$ points is a (t,m,d) – net in base b if every $\mathcal{A}$ with volume b^{t-m} contains b^t points of $\mathcal{P}_m$.*

Digital (t,m,d)-nets are a special case of (t,m,d)-nets, constructed using matrix-vector multiplications over finite fields. Digital sequences are infinite length digital nets, i.e., the first $n = b^m$ points of a digital sequence comprise a digital net for all integer $m \in \mathbb{N}_0$.

Definition 2 *For any non-negative integer $i = \ldots i_3 i_2 i_1 (\text{base}\, b)$, define the $\infty \times 1$ vector $\boldsymbol{\imath}$ as the vector of its digits, that is, $\boldsymbol{\imath} = (i_1, i_2, \ldots)^T$. For any point $z = 0.z_1 z_2 \ldots (\text{base}\, b) \in [0,1)$, define the $\infty \times 1$ vector of the digits of z, that is, $\mathbf{z} = (z_1, z_2, \ldots)^T$. Let $\mathsf{G}_1, \ldots, \mathsf{G}_d$ denote predetermined $\infty \times \infty$ generator matrices. The digital sequence in* base b *is $\{\boldsymbol{z}_0, \boldsymbol{z}_1, \boldsymbol{z}_2, \ldots\}$, where each $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{id})^T \in [0,1)^d$ is defined by*

$$\mathbf{z}_{i\ell} = \mathsf{G}_\ell\, \boldsymbol{\imath}, \quad \ell = 1, \ldots, d, \quad i = 0, 1, \ldots \,.$$

The value of t as mentioned in Definition 1 depends on the choice of G_ℓ.

Digital nets have a group structure under digitwise addition, which is a very useful property exploited in our algorithm, especially to develop a fast Bayesian transform that speedups computations. Digitwise addition, $\oplus$, and subtraction $\ominus$, are defined in terms of b-ary expansions of points in $[0,1)^d$,

$$\boldsymbol{z} \oplus \boldsymbol{y} = \left(\sum_{j=1}^{\infty} [z_{\ell j} + y_{\ell j} \bmod b] b^{-j} \bmod 1 \right)_{\ell=1}^{d},$$

$$\boldsymbol{z} \ominus \boldsymbol{y} = \left(\sum_{j=1}^{\infty} [z_{\ell j} - y_{\ell j} \bmod b] b^{-j} \bmod 1 \right)_{\ell=1}^{d},$$

where

$$\boldsymbol{z} = \left(\sum_{j=1}^{\infty} z_{\ell j} b^{-j} \right)_{\ell=1}^{d}, \quad \boldsymbol{y} = \left(\sum_{j=1}^{\infty} y_{\ell j} b^{-j} \right)_{\ell=1}^{d}, \quad z_{\ell j}, y_{\ell j} \in \{0, \cdots, b-1\}.$$

Similarly for integer values in $\mathbb{N}_0^d$, the digitwise addition, $\oplus$, and subtraction $\ominus$, are defined in terms of their b-ary expansions,

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

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

where

$$\boldsymbol{k} = \left(\sum_{j=0}^{\infty} k_{\ell j} b^{j} \right)_{\ell=1}^{d}, \quad \boldsymbol{l} = \left(\sum_{j=0}^{\infty} l_{\ell j} b^{j} \right)_{\ell=1}^{d}, \quad \boldsymbol{k}_{\ell j}, \boldsymbol{l}_{\ell j} \in \{0, \cdots, b-1\}.$$

Let $\{\boldsymbol{z}_i\}_{i=0}^{b^m-1}$ be a digital net. Then

$$\forall i_1, i_2 \in \{0, \cdots, b^m-1\}, \quad \boldsymbol{z}_{j_1} \oplus \boldsymbol{z}_{i_2} = \boldsymbol{z}_{i_3}, \quad \text{for some } i_3 \in \{0, \cdots, b^m-1\}.$$

The following very useful result, which will be further used to obtain the fast Bayesian transform, arises from the fundamental property of digital nets.

Lemma 1 *Let $\{\boldsymbol{z}_i\}_{i=0}^{b^m-1}$ be the digital-net and the corresponding digitally shifted net be $\{\boldsymbol{x}_i\}_{i=0}^{b^m-1}$, i.e.,*

$$\mathbf{x}_{i\ell} = \mathbf{z}_{i\ell} + \boldsymbol{\Delta}_l \bmod 1,$$

where $\mathbf{x}_{i\ell}$ is the ℓth component of ith digital net and $\boldsymbol{\Delta}_\ell$ is the digital shift for the ℓth component. Then,

$$\boldsymbol{x}_i \ominus \boldsymbol{x}_j = \boldsymbol{z}_i \ominus \boldsymbol{z}_j = \boldsymbol{z}_{i \ominus j}, \quad \forall i, j \in \mathbb{N}_0. \tag{1}$$

Also the digital subtraction is symmetric,

$$\boldsymbol{x}_i \ominus \boldsymbol{x}_i = \boldsymbol{0}, \qquad \boldsymbol{x}_i \ominus \boldsymbol{x}_j = \boldsymbol{x}_j \ominus \boldsymbol{x}_i, \quad \forall i, j \in \mathbb{N}_0. \tag{2}$$

Proof The proof can be obtained from the definition of digital nets which stated that the digital nets are obtained using generator matrices, $\mathbf{z}_{i\ell} = \mathsf{G}_\ell\, \boldsymbol{\imath} \bmod b$. Rewriting the subtraction using the generating matrix provides the result,

$$\begin{aligned} \mathbf{z}_{i\ell} - \mathbf{z}_{j\ell} \bmod b &= (\mathsf{G}_\ell \boldsymbol{\imath} \bmod b) - (\mathsf{G}_\ell \boldsymbol{\jmath} \bmod b) \\ &= (\mathsf{G}_\ell \boldsymbol{\imath} - \mathsf{G}_\ell \boldsymbol{\jmath}) \bmod b \\ &= \mathsf{G}_\ell (\boldsymbol{\imath} - \boldsymbol{\jmath}) \bmod b \end{aligned}$$

$$= \mathsf{G}_\ell(\overrightarrow{i \ominus j}) \bmod b$$
$$= \mathbf{z}_{i \ominus j\ \ell}.$$

The rest of the lemma is obvious from the definition of digital nets.

We chose digitally shifted and scrambled nets **?** for our Bayesian cubature algorithm. Digital shifts help to avoid having nodes at the origin, similar to the random shift used with lattice nodes. Scrambling helps to eliminate bias while retaining the low-discrepancy properties. A proof that a scrambled net preserves the property of (t, m, d)-net almost surely can be found in Owen **?**. The scrambling method proposed by Matoušek **?** is preferred since it is more efficient than the Owen's scrambling.

Sobol' nets **?** are a special case of (t, m, d)-nets when base $b = 2$. An example of 64 Sobol' nets in $d = 2$ is given in Figure **??**. The even coverage of the unit cube is ensured by a well chosen generating matrix. The choice of generating vector is typically done offline by computer search. See **?** and **?** for more on generating matrices. We use randomly scrambled and digitally shifted Sobol' sequences in this research **?**.

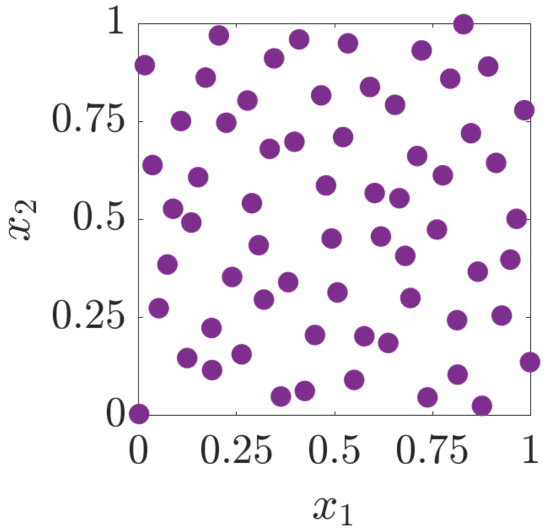

Fig. 1: Example of a scrambled Sobol' node set in $d = 2$. This plot can be reproduced using `PlotPoints.m`.

4 Walsh Kernels

Walsh kernels are product kernels based on the Walsh functions. We introduce the necessary concepts in this section.

4.1 Walsh functions

Like the Fourier transform used with lattice points (Section **??**), the Walsh-Hadamard transform, which we will simply call Walsh transform, is used for the digital nets. The Walsh transform is defined using Walsh functions. Recall $\mathbb{N}_0 := \{0, 1, 2, \cdots\}$. The one-dimensional Walsh functions in base b are defined as

$$\mathrm{wal}_{b,k}(x) := e^{2\pi\sqrt{-1}(x_1 k_0 + x_2 k_1 + \cdots)/b} = e^{2\pi\sqrt{-1}\mathbf{k}^T\mathbf{x}/b}, \tag{3}$$

for $x \in [0,1)$ and $k \in \mathbb{N}_0$ and the unique base b expansions $x = \sum_{j\geq 1} x_j b^{-i} = (0.x_1 x_2 \cdots)_b$, $\mathbf{x} = (x_1, x_2, \cdots)^T$ $k = \sum_{j\geq 0} k_j b^j = (\cdots k_1 k_0)_b$, $\mathbf{k} = (k_0, k_1, \cdots)^T$, and $\mathbf{k}^T\mathbf{x} = x_1 k_0 + x_2 k_1 + \cdots$ where the number of digits used in (3) are limited to the length required to represent x or k, i.e., $\max\left(\lceil -\log_b x\rceil, \lceil \log_b k\rceil\right)$. Multivariate Walsh functions are defined as the product of the one-dimensional Walsh functions,

$$\mathrm{wal}_{b,\boldsymbol{k}}(\boldsymbol{x}) := \prod_{\ell=1}^{d} \mathrm{wal}_{b,k_\ell}(x_\ell)$$

As shown in (3), for the case of $b = 2$, the Walsh functions only take the values in $\{1, -1\}$, i.e., $\mathrm{wal}_{b,\boldsymbol{k}} : [0,1)^d \to \{-1, 1\}$, $k \in \mathbb{N}_0^d$. Walsh functions form an orthonormal basis of the Hilbert space $L^2[0,1)^d$,

$$\int_{[0,1)^d} \mathrm{wal}_{b,\boldsymbol{l}}(\boldsymbol{x})\mathrm{wal}_{b,\boldsymbol{k}}(\boldsymbol{x})\mathrm{d}x = \delta_{\boldsymbol{l},\boldsymbol{k}}, \quad \forall \boldsymbol{l}, \boldsymbol{k} \in \mathbb{N}_0^d$$

Digital nets are designed to integrate certain Walsh functions without error. Thus our Bayesian cubature algorithm integrates linear combinations of certain Walsh functions without error. Functions that are well approximated by such linear combinations are then integrated with small errors.

In this research we use Sobol' nodes which are digital nets with base $b = 2$. So here afterwards base $b = 2$ is assumed. In this case, the Walsh function is simply

$$\mathrm{wal}_{2,\boldsymbol{k}}(\boldsymbol{x}) = (-1)^{\boldsymbol{k}^T\boldsymbol{x}}.$$

4.2 Walsh kernels

Consider the covariance kernels of the form,

$$C_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{t}) = K_{\boldsymbol{\theta}}(\boldsymbol{x} \ominus \boldsymbol{t}) \tag{4}$$

where $\ominus$ is bitwise subtraction. This is called a *digitally shift invariant kernel* because shifting both arguments of the covariance function by the same amount leaves the value unchanged. By a proper scaling of the function $K_{\boldsymbol{\theta}}$, it follows that assumption (**??**) is satisfied. The function $K_{\boldsymbol{\theta}}$ must be of the form that ensures that $C_{\boldsymbol{\theta}}$ is symmetric and positive definite, as assumed in (**??**). We drop the $\boldsymbol{\theta}$ sometimes to make the notation simpler. The Walsh kernels are of the form,

$$K_{\boldsymbol{\theta}}(\boldsymbol{x} \ominus \boldsymbol{t}) = \prod_{\ell=1}^{d} 1 + \eta_\ell \omega_r(x_\ell \ominus t_\ell), \quad \boldsymbol{\eta} = (\eta_1, \cdots, \eta_d), \quad \boldsymbol{\theta} = (r, \boldsymbol{\eta}) \tag{5}$$

where r is the kernel order, $\boldsymbol{\eta}$ is the kernel shape parameter, and

$$\omega_r(x) = \sum_{k=1}^{\infty} \frac{\mathrm{wal}_{2,k}(x)}{2^{2r\lfloor \log_2 k\rfloor}}.$$

Explicit expression is available for ω_r in the case of order $r = 1$ **?**,

$$\omega_1(x) = 6\left(\frac{1}{6} - 2^{\lfloor \log_2 x\rfloor - 1}\right). \tag{6}$$

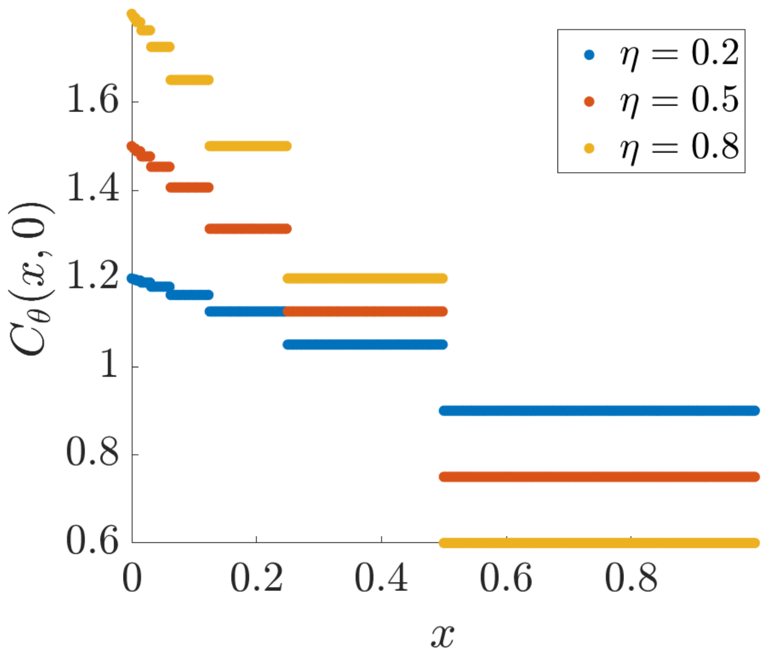

Fig. 2: Walsh kernel of order $r = 1$ in dimension $d = 1$. This figure can be reproduced using `plot_walsh_kernel.m`.

The Figure 2 shows the Walsh kernel (5) of order $r = 1$ in the interval $[0, 1)$. Unlike the shift-invariant kernels used with lattice nodes, low order Walsh kernels are discontinuous and are only piecewise constant. Smaller η_ℓ implies lesser variation in the amplitude of the kernel. Also, the Walsh kernels are digitally shift invariant but not periodic.

5 Eigenvectors

We show the eigenvectors V in (**??**) of the Gram matrix formed by the covariance kernel (5) and Sobol' nets are the columns of the Walsh-Hadamard matrix. First we introduce the necessary concepts.

5.1 Walsh transform

The Walsh-Hadamard transform (WHT) is a generalized class of discrete Fourier transform (DFT) and is much simpler to compute than the DFT. The WHT matrices are comprised of only ± 1 values, so the computation usually involves only ordinary additions and subtractions. Hence, the WHT is also sometimes called the integer transform. In comparison, the DFT that was used with lattice nodes, uses complex exponential functions and the computation involves complex, non-integer multiplications.

The WHT involves multiplications by $2^m \times 2^m$ Walsh-Hadamard matrices, which is constructed recursively, starting with $\mathsf{H}^{(0)} = 1$,

$$\begin{aligned}
\mathsf{H}^{(1)} &= \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \\
\mathsf{H}^{(2)} &= \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}, \\
&\vdots \\
\mathsf{H}^{(m)} &= \begin{pmatrix} \mathsf{H}^{(m-1)} & \mathsf{H}^{(m-1)} \\ \mathsf{H}^{(m-1)} & -\mathsf{H}^{(m-1)} \end{pmatrix} = \underbrace{\mathsf{H}^{(1)} \bigotimes \cdots \bigotimes \mathsf{H}^{(1)}}_{m \text{ times}} = \mathsf{H}^{(1)} \bigotimes \mathsf{H}^{(m-1)}
\end{aligned} \tag{7}$$

where $\bigotimes$ is Kronecker product. Alternatively for base $b = 2$, these matrices can be directly obtained by,

$$\mathsf{H}^{(m)} = \left((-1)^{(\boldsymbol{\imath}^T \boldsymbol{\jmath})} \right)_{i,j=0}^{2^m-1},$$

where the notation $\boldsymbol{\imath}^T \boldsymbol{\jmath}$ indicates the bitwise dot product.

5.2 Eigenvectors of C are columns of Walsh-Hadamard matrix

The Gram matrix $\mathsf{C}_{\boldsymbol{\theta}}$ formed by Walsh kernels and Sobol' nodes have a special structure called block-Toeplitz, which can be used to construct the fast Bayesian transform. A Toeplitz matrix is a diagonal-constant matrix in which each descending diagonal from left to right is constant. A block Toeplitz matrix is a special block matrix, which contains blocks that are repeated down the diagonals of the matrix. We prove that the eigenvectors of $\mathsf{C}_{\boldsymbol{\theta}}$ are columns of a Walsh-Hadamard matrix in two theorems.

Theorem 2 *Let $(\boldsymbol{x}_i)_{i=0}^{n-1}$ be digitally shifted Sobol' nodes and K be any function, then the Gram matrix,*
$\mathsf{C}_{\boldsymbol{\theta}} = \big(C(\boldsymbol{x}_i, \boldsymbol{x}_j)\big)_{i,j=0}^{n-1} = \big(K(\boldsymbol{x}_i \ominus \boldsymbol{x}_j)\big)_{i,j=0}^{n-1},$

$$\text{where} \qquad n = 2^m, \quad C(\boldsymbol{x}, \boldsymbol{t}) = K(\boldsymbol{x} \ominus \boldsymbol{t}), \quad \boldsymbol{x}, \boldsymbol{t} \in [0,1)^d,$$

is a 2×2 block-Toeplitz matrix and all the sub-blocks and their sub-sub-blocks, etc. are also 2×2 block-Toeplitz.

Proof We prove this theorem by induction. Let $\mathsf{C}_{\boldsymbol{\theta}}^{(m)}$ denote the Gram matrix of size $2^m \times 2^m$. The relation between sub-block matrices can be deciphered using the properties of digital nets. To help with the proof of block-Toeplitz structure, consider the digital net properties (1), (2), and notations,

$$\begin{aligned}
\mathsf{K}^{(m)} &:= \big(K(\boldsymbol{z}_i \ominus \boldsymbol{z}_j)\big)_{i,j=0}^{2^m-1} = \big(K(\boldsymbol{z}_{i \ominus j})\big)_{i,j=0}^{2^m-1}, \quad m = 1, 2, \cdots, \\
\mathsf{K}^{(m,q)} &:= \big(K(\boldsymbol{z}_{i \ominus j + q2^m})\big)_{i,j=0}^{2^m-1}, \quad q = 0, 1, \cdots.
\end{aligned}$$

These two notations are related by $\mathsf{K}^{(m)} = \mathsf{K}^{(m,0)}$. Please note that $\mathsf{C}_{\boldsymbol{\theta}}^{(m)} = \mathsf{K}^{(m,0)}$. We will prove $\mathsf{K}^{(m,q)}$ is a 2×2 block-toeplitz matrix for all $m \in \mathbb{N}, q \in \mathbb{N}$.

As the first step, we verify the property holds for $m = 1$,

$$\mathsf{K}^{(1,q)} = \begin{pmatrix} K(\boldsymbol{z}_{0\ominus 0+q2^1}) & K(\boldsymbol{z}_{1\ominus 0+q2^1}) \\ K(\boldsymbol{z}_{0\ominus 1+q2^1}) & K(\boldsymbol{z}_{1\ominus 1+q2^1}) \end{pmatrix} = \begin{pmatrix} K(\boldsymbol{z}_{2q}) & K(\boldsymbol{z}_{1+2q}) \\ K(\boldsymbol{z}_{1+2q}) & K(\boldsymbol{z}_{2q}) \end{pmatrix}, \quad \text{by (1)}$$

has diagonal elements repeated. Thus by definition, it is a 2×2 block-Toeplitz.

Now assume that $\mathsf{K}^{(m,q)}$ is block-Toeplitz. We need to prove $\mathsf{K}^{(m+1,q)}$ is also a 2×2 block-Toeplitz. Let $n = 2^m$,

$$\mathsf{K}^{(m+1)} = \begin{pmatrix} K(\boldsymbol{z}_{0\ominus 0}) & \dots & K(\boldsymbol{z}_{0\ominus n-1}) & K(\boldsymbol{z}_{0\ominus n}) & \dots & K(\boldsymbol{z}_{0\ominus 2n-1}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\boldsymbol{z}_{n-1\ominus 0}) & \dots & K(\boldsymbol{z}_{n-1\ominus n-1}) & K(\boldsymbol{z}_{n-1\ominus n}) & \dots & K(\boldsymbol{z}_{n-1\ominus 2n-1}) \\ K(\boldsymbol{z}_{n\ominus 0}) & \dots & K(\boldsymbol{z}_{n\ominus n-1}) & K(\boldsymbol{z}_{n\ominus n}) & \dots & K(\boldsymbol{z}_{n\ominus 2n-1}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ K(\boldsymbol{z}_{2n-1\ominus 0}) & \dots & K(\boldsymbol{z}_{2n-1\ominus n-1}) & K(\boldsymbol{z}_{2n-1\ominus n}) & \dots & K(\boldsymbol{z}_{2n-1\ominus 2n-1}) \end{pmatrix}$$

$$= \begin{pmatrix} \begin{pmatrix} K(\boldsymbol{z}_0) & \dots & K(\boldsymbol{z}_{n-1}) \\ \vdots & \vdots & \vdots \\ K(\boldsymbol{z}_{n-1}) & \dots & K(\boldsymbol{z}_0) \end{pmatrix} & \begin{pmatrix} K(\boldsymbol{z}_n) & \dots & K(\boldsymbol{z}_{2n-1}) \\ \vdots & \vdots & \vdots \\ K(\boldsymbol{z}_{2n-1}) & \dots & K(\boldsymbol{z}_n) \end{pmatrix} \\ \begin{pmatrix} K(\boldsymbol{z}_n) & \dots & K(\boldsymbol{z}_{2n-1}) \\ \vdots & \vdots & \vdots \\ K(\boldsymbol{z}_{2n-1}) & \dots & K(\boldsymbol{z}_n) \end{pmatrix} & \begin{pmatrix} K(\boldsymbol{z}_0) & \dots & K(\boldsymbol{z}_{n-1}) \\ \vdots & \vdots & \vdots \\ K(\boldsymbol{z}_{n-1}) & \dots & K(\boldsymbol{z}_0) \end{pmatrix} \end{pmatrix}$$

$$= \begin{pmatrix} \mathsf{K}^{(m)} & \mathsf{K}^{(m,1)} \\ \mathsf{K}^{(m,1)} & \mathsf{K}^{(m)} \end{pmatrix}$$

is a 2×2 block-Toeplitz, where we used the properties (1), (2) and facts $2n-1\ominus n = n-1$, $2n-1\ominus n-1 = n$, and $n\ominus n-1 = 2n-1$. Thus $\mathsf{K}^{(m+1)}$ is a 2×2 block-Toeplitz. Similarly

$$\mathsf{K}^{(m+1,q)} = \begin{pmatrix} \mathsf{K}^{(m,q)} & \mathsf{K}^{(m,q+1)} \\ \mathsf{K}^{(m,q+1)} & \mathsf{K}^{(m,q)} \end{pmatrix}$$

is a 2×2 block-Toeplitz. Thus $\mathsf{C}_{\boldsymbol{\theta}}^{(m)}$ of size $2^m \times 2^m$, for $m \in \mathbb{N}$, is a 2×2 block-Toeplitz and every block and it's sub-blocks of size 2^p, $p \in \mathbb{N}$, $p \leq m$ are also 2×2 block-Toeplitz.

Theorem 3 *The Walsh-Hadamard matrix $\mathsf{H}^{(m)}$ factorizes $\mathsf{C}_{\boldsymbol{\theta}}^{(m)}$, so that the columns of Walsh-Hadamard matrix are the eigenvectors of $\mathsf{C}_{\boldsymbol{\theta}}^{(m)}$, i.e.,*

$$\mathsf{H}^{(m)}\mathsf{C}_{\boldsymbol{\theta}}^{(m)} = \mathsf{\Lambda}^{(m)}\mathsf{H}^{(m)}, \quad m \in \mathbb{N}.$$

Proof Again, we use the proof-by-induction technique to show that the Walsh-Hadamard matrix factorizes $\mathsf{K}^{(m,q)}$. We can easily see the Hadamard matrix $\mathsf{H}^{(1)}$ diagonalizes $\mathsf{K}^{(1,q)}$,

$$\begin{aligned} \mathsf{H}^{(1)}\mathsf{K}^{(1,q)} &= \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} K(\boldsymbol{z}_{0+q2^1}) & K(\boldsymbol{z}_{1+q2^1}) \\ K(\boldsymbol{z}_{1+q2^1}) & K(\boldsymbol{z}_{0+q2^1}) \end{pmatrix}, \quad \text{by Theorem 2} \\ &= \begin{pmatrix} K(\boldsymbol{z}_{2q}) + K(\boldsymbol{z}_{2q+1}) & K(\boldsymbol{z}_{2q}) + K(\boldsymbol{z}_{2q+1}) \\ K(\boldsymbol{z}_{2q}) - K(\boldsymbol{z}_{2q+1}) & K(\boldsymbol{z}_{2q+1}) - K(\boldsymbol{z}_{2q}) \end{pmatrix} \\ &= \begin{pmatrix} K(\boldsymbol{z}_{2q}) + K(\boldsymbol{z}_{2q+1}) & 0 \\ 0 & K(\boldsymbol{z}_{2q}) - K(\boldsymbol{z}_{2q+1}) \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \end{aligned}$$

$$= \mathsf{\Lambda}^{(1,q)}\mathsf{H}^{(1)},$$

where $\mathsf{\Lambda}^{(1,q)}$ is a diagonal matrix, thus $\mathsf{H}^{(1)}$ factorizes $\mathsf{K}^{(1,q)}$.

Now assume $\mathsf{H}^{(m)}$ factorizes $\mathsf{K}^{(m,q)}$, so $\mathsf{H}^{(m)}\mathsf{K}^{(m,q)} = \mathsf{\Lambda}^{(m,q)}\mathsf{H}^{(m)}$ where $\mathsf{\Lambda}^{(m,q)}$ is diagonal. We need to prove $\mathsf{H}^{(m+1)}$ factorizes $\mathsf{K}^{(m+1,q)}$,

$$\begin{aligned}
\mathsf{H}^{(m+1)}\mathsf{K}^{(m+1,q)} &= \begin{pmatrix} \mathsf{H}^{(m)} & \mathsf{H}^{(m)} \\ \mathsf{H}^{(m)} & -\mathsf{H}^{(m)} \end{pmatrix} \begin{pmatrix} \mathsf{K}^{(m,q)} & \mathsf{K}^{(m,q+1)} \\ \mathsf{K}^{(m,q+1)} & \mathsf{K}^{(m,q)} \end{pmatrix}, \quad \text{by Theorem 2} \\
&= \begin{pmatrix} \mathsf{H}^{(m)}(\mathsf{K}^{(m,q)} + \mathsf{K}^{(m,q+1)}) & \mathsf{H}^{(m)}(\mathsf{K}^{(m,q)} + \mathsf{K}^{(m,q+1)}) \\ \mathsf{H}^{(m)}(\mathsf{K}^{(m,q)} - \mathsf{K}^{(m,q+1)}) & \mathsf{H}^{(m)}(\mathsf{K}^{(m,q+1)} - \mathsf{K}^{(m,q)}) \end{pmatrix} \\
&= \begin{pmatrix} (\mathsf{\Lambda}^{(m,q)} + \mathsf{\Lambda}^{(m,q+1)})\mathsf{H}^{(m)} & (\mathsf{\Lambda}^{(m,q)} + \mathsf{\Lambda}^{(m,q+1)})\mathsf{H}^{(m)} \\ (\mathsf{\Lambda}^{(m,q)} - \mathsf{\Lambda}^{(m,q+1)})\mathsf{H}^{(m)} & (\mathsf{\Lambda}^{(m,q+1)} - \mathsf{\Lambda}^{(m,q)})\mathsf{H}^{(m)} \end{pmatrix} \\
&= \begin{pmatrix} \mathsf{\Lambda}^{(m,q)} + \mathsf{\Lambda}^{(m,q+1)} & 0 \\ 0 & \mathsf{\Lambda}^{(m,q)} - \mathsf{\Lambda}^{(m,q+1)} \end{pmatrix} \begin{pmatrix} \mathsf{H}^{(m)} & \mathsf{H}^{(m)} \\ \mathsf{H}^{(m)} & -\mathsf{H}^{(m)} \end{pmatrix} \\
&= \mathsf{\Lambda}^{(m+1,q)}\mathsf{H}^{(m+1)}.
\end{aligned}$$

Thus, $\mathsf{H}^{(m+1)}$ factorizes $\mathsf{K}^{(m+1,q)}$ to a diagonal matrix $\mathsf{\Lambda}^{(m+1,q)}$. This implies $\mathsf{H}^{(p)}$ factorizes $\mathsf{C}_{\boldsymbol{\theta}}^{(p)}$ for $p \in \mathbb{N}$. Please recall $\mathsf{C}_{\boldsymbol{\theta}}^{(p)} = \mathsf{K}^{(p,0)}$. Here we used the fact that both H and K are symmetric positive definite.

5.3 Fast Bayesian transform

We can easily show that the Walsh-Hadamard matrices satisfy the assumptions of fast Bayesian transform (**??**). As shown in Section 5.2 the columns of $\mathsf{H}^{(m)}$ are the eigenvectors. Since the Gram matrix C is symmetric, the columns/rows of Walsh-Hadamard matrices are mutually orthogonal. Thus the Gram matrix can be written as

$$\mathsf{C}^{(m)} = \frac{1}{n}\mathsf{H}^{(m)}\mathsf{\Lambda}^{(m)}\mathsf{H}^{(m)}, \quad \text{where} \quad \mathsf{H}^{(m)} = \underbrace{\mathsf{H}^{(1)} \bigotimes \cdots \bigotimes \mathsf{H}^{(1)}}_{m \text{ times}}. \qquad (8)$$

Assumption (**??**) follows automatically by the fact that Walsh-Hadamard matrices can be constructed analytically. Assumption (**??**) can also be verified as the first row/column are one vectors. Finally, assumption (**??**) is satisfied due to the fact that fast Walsh transform can be computed in $\mathcal{O}(n \log n)$ operations using fast Walsh-Hadamard transform. Thus the Walsh-Hadamard transform is a fast Bayesian transform, $\mathsf{V} := \mathsf{H}$, as per (**??**).

We have implemented a fast adaptive Bayesian cubature algorithm using the kernel (5) with $r = 1$ and Sobol' points **?** in MATLAB as part of the Guaranteed Adaptive Integration Library (GAIL) **?** as `cubBayesNet_g`. The Sobol' points used in this algorithm are generated using MATLAB's builtin function `sobolset` and scrambled using MATLAB function `scramble` **?**. The fast Walsh-Hadamard transform (8) is computed using MATLAB's builtin function `fwht` with *hadamard* ordering.

5.4 Iterative Computation of Walsh Transform

In every iteration of our algorithm, we double the number of function values. Using the technique described here, we have to only compute the Walsh transform for the newly added function values. Similar to the lattice points, Sobol' points are extensible by definition. This property is used in our algorithm to improve the integration accuracy till the required error tolerance is met. Sobol' nodes can be combined with Hadamard matrices as demonstrated here for iterative computation. Let $\widetilde{\boldsymbol{y}} = \mathsf{H}^{(m+1)}\boldsymbol{y}$ for some arbitrary $\boldsymbol{y} \in \mathbb{R}^{2n}$, $n = 2^m$. Define,

$$\boldsymbol{y} = \begin{pmatrix} [1.1]y_1 \\ \vdots \\ y_{2n} \end{pmatrix}, \quad \boldsymbol{y}^{(1)} = \begin{pmatrix} [1.1]y_1 \\ \vdots \\ y_n \end{pmatrix}, \quad \boldsymbol{y}^{(2)} = \begin{pmatrix} [1.1]y_{n+1} \\ \vdots \\ y_{2n} \end{pmatrix},$$

$$\widetilde{\boldsymbol{y}}^{(1)} = \mathsf{H}^{(m)}\boldsymbol{y}^{(1)} = \begin{pmatrix} [1.0]\widetilde{y}_1^{(1)} \\ \widetilde{y}_2^{(1)} \\ \vdots \\ \widetilde{y}_n^{(1)} \end{pmatrix}, \quad \widetilde{\boldsymbol{y}}^{(2)} = \mathsf{H}^{(m)}\boldsymbol{y}^{(2)} = \begin{pmatrix} [1.0]\widetilde{y}_1^{(2)} \\ \widetilde{y}_2^{(2)} \\ \vdots \\ \widetilde{y}_n^{(2)} \end{pmatrix}.$$

Then,

$$\begin{aligned} \widetilde{\boldsymbol{y}} &= \mathsf{H}^{(m+1)}\boldsymbol{y} \\ &= \begin{pmatrix} \mathsf{H}^{(m)} & \mathsf{H}^{(m)} \\ \mathsf{H}^{(m)} & -\mathsf{H}^{(m)} \end{pmatrix} \begin{pmatrix} \boldsymbol{y}^{(1)} \\ \boldsymbol{y}^{(2)} \end{pmatrix}, \qquad \text{by (7)} \\ &= \begin{pmatrix} \mathsf{H}^{(m)}\boldsymbol{y}^{(1)} + \mathsf{H}^{(m)}\boldsymbol{y}^{(2)} \\ \mathsf{H}^{(m)}\boldsymbol{y}^{(1)} - \mathsf{H}^{(m)}\boldsymbol{y}^{(2)} \end{pmatrix} \\ &= \begin{pmatrix} \widetilde{\boldsymbol{y}}^{(1)} + \widetilde{\boldsymbol{y}}^{(2)} \\ \widetilde{\boldsymbol{y}}^{(1)} - \widetilde{\boldsymbol{y}}^{(2)} \end{pmatrix} =: \widetilde{\boldsymbol{y}} \quad . \end{aligned}$$

As before with the lattice nodes, the computational cost to compute $\mathsf{V}^{(m+1)H}\boldsymbol{y}$ is twice the cost of computing $\mathsf{V}^{(m)H}\boldsymbol{y}^{(1)}$ plus $2n$ additions, where $n = 2^m$. An inductive argument shows that for any $m \in \mathbb{N}$, $\mathsf{V}^{(m)H}\boldsymbol{y}$ requires only $\mathcal{O}(n \log n)$ operations. Usually the multiplications in $\mathsf{V}^{(m)H}\boldsymbol{y}^{(1)}$ are multiplications by -1 which are simply accomplished using sign change or negation, requiring no multiplications at all.

6 Higher Order Nets

Higher order digital nets are an extension of (t, m, d)-nets, introduced in **?**. They can be used to numerically integrate smoother functions which are not necessarily periodic, but have square integrable mixed partial derivatives of order α, at a rate of $\mathcal{O}(n^{-\alpha})$ multiplied by a power of a $\log n$ factor using rules corresponding to the modified (t, m, d)-nets. We want to emphasize that quasi-Monte Carlo rules based on these point sets can achieve convergence rates faster than $\mathcal{O}(n^{-1})$. Higher order digital nets are constructed using matrix-vector multiplications over finite fields.

One could develop matching digitally shift invariant kernels to formulate the fast Bayesian cubature. Bayesian cubatures using higher order digital nets are a topic for future research.