Symmetry exploits for Bayesian cubature methods

Toni Karvonen[1] · **Simo Särkkä**[1] · **Chris. J. Oates**[2]

Abstract

Bayesian cubature provides a flexible framework for numerical integration, in which a priori knowledge on the integrand can be encoded and exploited. This additional flexibility, compared to many classical cubature methods, comes at a computational cost which is cubic in the number of evaluations of the integrand. It has been recently observed that fully symmetric point sets can be exploited in order to reduce—in some cases substantially—the computational cost of the standard Bayesian cubature method. This work identifies several additional symmetry exploits within the Bayesian cubature framework. In particular, we go beyond earlier work in considering non-symmetric measures and, in addition to the standard Bayesian cubature method, present exploits for the Bayes–Sard cubature method and the multi-output Bayesian cubature method.

Keywords Probabilistic numerics · Numerical integration · Gaussian processes · Fully symmetric sets

1 Introduction

This paper considers the numerical approximation of an integral

$$I(f^{\dagger}) := \int_M f^{\dagger}(\boldsymbol{x}) \mathrm{d}\nu(\boldsymbol{x}),$$

where $(M, \mathcal{B}, \nu)$ is a Borel probability space with M any Borel measurable non-empty subset of $\mathbb{R}^m$ and $f^{\dagger} \colon M \to \mathbb{R}$ is a $\mathcal{B}$-measurable scalar-valued integrand (vector-valued integrands will be considered in Sect. 4). Additional assumptions will be made when necessary. Our interest is in the situation where the exact values of $f^{\dagger}$ cannot be deduced until the function itself is evaluated, and that the evaluations are associated with a substantial computational cost or a very large number of them is required. Such situations are typical in, for example, uncertainty quantification for chemical systems (Najm et al. 2009), fluid mechanical simulation (Xiu and Karniadakis 2003) and certain financial applications (Holtz 2011).

In the presence of a limited computational budget, it is natural to exploit any contextual information that may be available on the integrand. Classical cubatures, such as spline-based or Gaussian cubatures, are able to exploit abstract mathematical information, such as the number of continuous derivatives of the integrand (Davis and Rabinowitz 2007). However, in situations where more detailed or specific contextual information is available to the analyst, the use of generic classical cubatures can be sub-optimal.

The language of probabilities provides one mechanism in which contextual information about the integrand can be captured. Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. Then an analyst can elicit their prior information about the integrand $f^{\dagger}$ in the form of a stochastic process model

$$\omega \mapsto f(\cdot\,; \omega), \quad \omega \in \Omega, \tag{1}$$

wherein the function $\boldsymbol{x} \mapsto f(\boldsymbol{x}; \omega)$ is $\mathcal{B}$-measurable for each fixed $\omega \in \Omega$. Through the stochastic process, the analyst can encode both abstract mathematical information, such as the number of continuous derivatives of the integrand, and specific contextual information, such as the possibility of a trend or a periodic component. The process of elicitation is not discussed in this work (see Diaconis 1988 and Hennig et al. 2015); for our purposes the stochastic process in (1) is considered to be provided.

✉ Toni Karvonen
toni.karvonen@aalto.fi

Simo Särkkä
simo.sarkka@aalto.fi

Chris. J. Oates
chris.oates@ncl.ac.uk

[1] Department of Electrical Engineering and Automation, Aalto University, Espoo, Finland

[2] School of Mathematics, Statistics and Physics, Newcastle University, Newcastle upon Tyne, UK

In Bayesian cubature methods, due to Larkin (1972) and re-discovered by Diaconis (1988), O'Hagan (1991) and Minka (2000), the analyst first selects a point set $X = \{x_i\}_{i=1}^N \subset M$, $N \in \mathbb{N}$, on which the true integrand $f^\dagger$ is evaluated. Let this data be denoted $\mathcal{D} = \{(x_i, f^\dagger(x_i))\}_{i=1}^N$. Then the analyst conditions their stochastic process according to these data $\mathcal{D}$, to obtain a second stochastic process

$$\omega \mapsto f_N(\cdot\,; \omega).$$

The analyst reports the implied distribution over the value of the integral of interest; that is the law of the random variable

$$\omega \mapsto \int_M f_N(x; \omega) \mathrm{d}\nu(x).$$

This distribution can be computed in closed form under certain assumptions on the structure of the prior model. A sufficient condition is that the stochastic process is Gaussian, which (arguably) does not severely restrict the analyst in terms of what contextual information can be included (Rasmussen and Williams 2006). In addition, the probabilistic output of the method enables uncertainty quantification for the unknown true value of the integral (Larkin 1972; Cockayne et al. 2017; Briol et al. 2019). These appealing properties have led to Bayesian cubature methods being used in diverse areas such as from computer graphics (Marques et al. 2013), nonlinear filtering (Prüher et al. 2017) and applied Bayesian statistics (Osborne et al. 2012a).

The theoretical aspects of Bayesian cubature methods have now been widely studied. In particular, convergence of the posterior mean point estimator

$$\int_\Omega \int_M f_N(x; \omega) \mathrm{d}\nu(x) \mathrm{d}\mathbb{P}(\omega) \rightarrow \int_M f^\dagger(x) \mathrm{d}\nu(x) \qquad (2)$$

as $N \rightarrow \infty$ has been studied in both the well-specified (Bezhaev 1991; Sommariva and Vianello 2006; Briol et al. 2015; Ehler et al. 2019; Briol et al. 2019) and mis-specified (Kanagawa et al. 2016, 2019) regimes. Some relationships between the posterior mean estimator and classical cubature methods have been documented in Diaconis (1988), Särkkä et al. (2016) and Karvonen and Särkkä (2017). In Larkin (1974), O'Hagan (1991) and Karvonen et al. (2018) the *Bayes–Sard* framework was studied, where it was proposed to incorporate an explicit parametric component (O'Hagan 1978) into the prior model in order that contextual information, such as trends, can be properly encoded. The choice of point set X for Bayesian cubature has been studied in Briol et al. (2015), Bach (2017), Briol et al. (2017), Oettershagen (2017), Chen et al. (2018) and Pronzato and Zhigljavsky (2018). In addition, several extensions have been considered to address specific technical challenges posed by non-negative integrands (Chai and Garnett 2018),

model evidence integrals in a Bayesian context (Osborne et al. 2012a; Gunter et al. 2014), ratios (Osborne et al. 2012b), non-Gaussian prior models (Kennedy 1998; Prüher et al. 2017), measures that can be only be sampled (Oates et al. 2017), and vector-valued integrands (Xi et al. 2018).

Despite these recent successes, a significant drawback of Bayesian cubature methods is that the cost of computing the distributional output is typically cubic in N, the size of the point set. For integrals whose domain M is high-dimensional, the number N of points required can be exponential in $m = \dim(M)$. Thus the cubic cost associated with Bayesian cubature methods can render them impractical. In recent work, Karvonen and Särkkä (2018) noted that symmetric structure in the point set can be exploited to reduce the total computational cost. Indeed, in some cases the exponential dependence on m can be reduced to (approximately) linear. This is a similar effect to that achieved in the circulant embedding approach (Dietrich and Newsam 1997), or by the use of $\mathcal{H}$-matrices (Hackbusch 1999) and related approximations (Schäfer et al. 2017), though the approaches differ at a fundamental level. The aim of this paper is to present several related *symmetry exploits* that are specifically designed to reduce computational cost of Bayesian cubature methods.

Our principal contributions are following: *First*, the techniques developed in Karvonen and Särkkä (2018) are extended to the Bayes–Sard cubature method. This results in a computational method that is, essentially, of the complexity $\mathcal{O}(J^3 + JN)$, where J is the number of symmetric sets that constitute the full point set, instead of being cubic in N. In typical scenarios, there are at most a few hundred symmetric sets even though the total number of points can go up to millions. *Second*, we present an extension to the multi-output (i.e. vector-valued) Bayesian cubature method that is used to simultaneously integrate $D \in \mathbb{N}$ related integrals. In this case, the computational complexity is reduced from $\mathcal{O}(D^3 N^3)$ to $\mathcal{O}(D^3 J^3 + D J N)$. *Third*, a symmetric change of measure technique is proposed to avoid the (strong) assumption of symmetry on the measure ν that was required in Karvonen and Särkkä (2018). *Fourth*, the performance of our techniques is empirically explored. Throughout, our focus is not on the performance of these integration methods, which has been explored in earlier work, already cited. Rather, our focus is on how computation for these methods can be accelerated.

The remainder of the article is structured as follows: Sect. 2 covers the essential background for Bayesian cubature methods and introduces fully symmetric sets that are used in the symmetry exploits throughout the article. Sections 3 and 4 develop fully symmetric Bayes–Sard cubature and fully symmetric multi-output Bayesian cubature. Section 5 explains how the assumption that ν is symmetric can be relaxed. In Sect. 6, a detailed selection of empirical results

is presented. Finally, some concluding remarks and discussion are contained in Sect. 7.

2 Background

This section reviews the standard Bayesian cubature method, due to Larkin (1972), and explains how fully symmetric sets can be used to alleviate its computational cost, as proposed by Karvonen and Särkkä (2018).

2.1 Standard Bayesian cubature

In this section, we present explicit formulae for the Bayesian cubature method in the case where the prior model (1) is a Gaussian random field. To simplify the notation, Sects. 2 and 3 assume that the integrand has scalar output (i.e. $D = 1$); this is then extended to vector-valued output in Sect. 4.

To reduce the notational overhead, in what follows the $\omega \in \Omega$ argument is left implicit. Thus we consider $f(\boldsymbol{x})$ to be a scalar-valued random variable for each $\boldsymbol{x} \in M$. In particular, in this paper we focus on stochastic processes that are Gaussian, meaning that there exists a *mean function* $m : M \to \mathbb{R}$ and a symmetric positive-definite *covariance function* (or *kernel*) $k : M \times M \to \mathbb{R}$ such that $[f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)]^\mathsf{T} \in \mathbb{R}^N$ has the multivariate Gaussian distribution

$$
N\left(\begin{bmatrix} m(\boldsymbol{x}_1) \\ \vdots \\ m(\boldsymbol{x}_N) \end{bmatrix}, \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_1, \boldsymbol{x}_N) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_N, \boldsymbol{x}_1) & \cdots & k(\boldsymbol{x}_N, \boldsymbol{x}_N) \end{bmatrix}\right)
$$

for any $N \in \mathbb{N}$ and all point sets $\{\boldsymbol{x}_i\}_{i=1}^N \subset M$. We assume that $\int_M k(\boldsymbol{x}, \boldsymbol{x}) \, \mathrm{d}\nu(\boldsymbol{x}) < \infty$.

The conditional distribution f_N of this field, based on the data $\mathcal{D} = \{(\boldsymbol{x}_i, f^\dagger(\boldsymbol{x}_i)\}_{i=1}^N$ of function evaluations at the points $X = \{\boldsymbol{x}_i\}_{i=1}^N$, is also Gaussian, with mean and covariance functions

$$
m_N(\boldsymbol{x}) = m(\boldsymbol{x}) + \boldsymbol{k}_X(\boldsymbol{x})^\mathsf{T} \boldsymbol{K}_X^{-1}(\boldsymbol{f}_X^\dagger - \boldsymbol{m}_X), \tag{3}
$$

$$
k_N(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{k}_X(\boldsymbol{x})^\mathsf{T} \boldsymbol{K}_X^{-1} \boldsymbol{k}_X(\boldsymbol{x}'), \tag{4}
$$

where the vector $\boldsymbol{f}_X^\dagger \in \mathbb{R}^N$ contains evaluations of the integrand, $[\boldsymbol{f}_X^\dagger]_i = f^\dagger(\boldsymbol{x}_i)$, the vector $\boldsymbol{m}_X \in \mathbb{R}^N$ contains evaluations of the prior mean, $[\boldsymbol{m}_X]_i = m(\boldsymbol{x}_i)$, the vector $\boldsymbol{k}_X(\boldsymbol{x}) \in \mathbb{R}^N$ contains evaluations of the kernel, $[\boldsymbol{k}_X(\boldsymbol{x})]_i = k(\boldsymbol{x}, \boldsymbol{x}_i)$, and $\boldsymbol{K}_X = \boldsymbol{K}_{X,X} \in \mathbb{R}^{N \times N}$ is the *kernel matrix*, $[\boldsymbol{K}_X]_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. From the fact that linear functionals of Gaussian processes are Gaussian, we obtain that

$$
\int_M f_N(\boldsymbol{x}) \, \mathrm{d}\nu(\boldsymbol{x}) \sim N\left(\mu_N(f^\dagger), \sigma_N^2\right), \tag{5}
$$

with

$$
\mu_N(f^\dagger) = I(m) + \boldsymbol{k}_{\nu,X}^\mathsf{T} \boldsymbol{K}_X^{-1}(\boldsymbol{f}_X^\dagger - \boldsymbol{m}_X), \tag{6}
$$

$$
\sigma_N^2 = k_{\nu,\nu} - \boldsymbol{k}_{\nu,X}^\mathsf{T} \boldsymbol{K}_X^{-1} \boldsymbol{k}_{\nu,X}. \tag{7}
$$

Here $k_\nu(\boldsymbol{x}) := \int_M k(\boldsymbol{x}, \boldsymbol{x}') \, \mathrm{d}\nu(\boldsymbol{x}')$ is called the *kernel mean* function (Smola et al. 2007) and $\boldsymbol{k}_{\nu,X} \in \mathbb{R}^N$ is the column vector with $[\boldsymbol{k}_{\nu,X}]_i = k_\nu(\boldsymbol{x}_i)$, while $k_{\nu,\nu} := \int_M k_\nu(\boldsymbol{x}) \, \mathrm{d}\nu(\boldsymbol{x}) \geq 0$ is the variance of the integral itself under the prior model. The assumption $\int_M k(\boldsymbol{x}, \boldsymbol{x}) \, \mathrm{d}\nu(\boldsymbol{x}) < \infty$ guarantees that the kernel mean is finite. This method is known as the *standard Bayesian cubature*, with the implicit understanding that the model for the integrand should be carefully selected to ensure (5) is well-calibrated (Briol et al. 2019), meaning that the uncertainty assessment can be trusted. The need for careful calibration is in line with standard approaches to the Gaussian process regression task (Rasmussen and Williams 2006).

To understand when the Bayesian cubature output is meaningful, it is useful to write the posterior mean and variance (6) and (7) in terms of the *weight* vector

$$
\boldsymbol{w}_X := \boldsymbol{K}_X^{-1} \boldsymbol{k}_{\nu,X}. \tag{8}
$$

That is, we have $\mu_N(f^\dagger) = I(m) + \boldsymbol{w}_X^\mathsf{T}(\boldsymbol{f}_X^\dagger - \boldsymbol{m}_X)$ and $\sigma_N^2 = k_{\nu,\nu} - \boldsymbol{w}_X^\mathsf{T} \boldsymbol{k}_{\nu,X}$. Let $\mathcal{H}(k)$ be the Hilbert space reproduced by the kernel k (see Berlinet and Thomas-Agnan (2011) for background). It can then be verified that $\boldsymbol{w}_X$ solves a quadratic minimisation problem of approximating k_ν with a function from the finite-dimensional space spanned by $\{k(\cdot, \boldsymbol{x})\}_{\boldsymbol{x} \in X} \subset \mathcal{H}(k)$, namely:

$$
\boldsymbol{w}_X = \arg\min_{\boldsymbol{w} \in \mathbb{R}^N} \left\| k_\nu(\cdot) - \sum_{i=1}^N w_i k(\cdot, \boldsymbol{x}_i) \right\|_{\mathcal{H}(k)}.
$$

The minimum value of this norm is σ_N; see e.g. Oettershagen (2017, Ch. 3) and Bach et al. (2012). Equivalently, the weight vector can be obtained as the minimiser of the worst-case error

$$
\sup_{\|f^\dagger\|_{\mathcal{H}(k)} \leq 1} \left| \int_M f^\dagger(\boldsymbol{x}) \, \mathrm{d}\nu(\boldsymbol{x}) - \sum_{i=1}^N w_i f^\dagger(\boldsymbol{x}_i) \right|
$$

among all cubature rules with points X, with σ_N corresponding to the minimal worst-case error (Briol et al. 2019; Oettershagen 2017). Thus, in terms of uncertainty quantification, the posterior standard deviation σ_N can indeed be meaningfully related to the integration problem being solved.

The principal motivation for this work is the observation that both (6) and (7) involve the solution of an N-dimensional linear system defined by the matrix $\boldsymbol{K}_X$. In general this is a

dense matrix and, as such, in the absence of additional structure in the linear system (Karvonen and Särkkä 2018) or further approximations [(e.g. Lázaro-Gredilla et al. (2010), Hensman et al. (2018), Schäfer et al. (2017)], the computational complexity associated with the standard Bayesian cubature method is $\mathcal{O}(N^3)$. Moreover, it is often the case that $\boldsymbol{K}_X$ is ill-conditioned (Schaback 1995; Stein 2012). The exploitation of symmetric structure to circumvent the solution of a large and ill-conditioned linear system would render Bayesian cubature more practical, in the sense of computational efficiency and numerical robustness; this is the contribution of the present article.

2.2 Symmetry properties

Next we introduce fully symmetric sets and related symmetry concepts, before explaining in Sect. 2.3 how these can be exploited for computational simplification in the standard Bayesian cubature method. Note that, in what follows, no symmetry properties are needed for the integrand $f^\dagger$ itself.

2.2.1 Fully symmetric point sets

Given a vector $\boldsymbol{\lambda} \in \mathbb{R}^m$, the *fully symmetric set* $[\boldsymbol{\lambda}] \subset \mathbb{R}^m$ generated by this vector is defined as the point set consisting of all vectors that can be obtained from $\boldsymbol{\lambda}$ via coordinate permutations and sign changes. That is,

$$\boldsymbol{\lambda}t = [\lambda_1, \ldots, \lambda_d]$$
$$:= \bigcup_{q \in \Pi_m} \bigcup_{\boldsymbol{s} \in S_m} \left\{ (s_1 \lambda_{q_1}, \ldots, s_d \lambda_{q_d}) \right\} \subset \mathbb{R}^m,$$

where Π_m and S_m stand for the collections of all permutations of the first m positive integers and of all vectors of the form $\boldsymbol{s} = (s_1, \ldots, s_m)$ with each s_i either 1 or −1. Here $\boldsymbol{\lambda}$ is called a *generator vector* and its individual elements are called *generators*. Alternatively, we can write the fully symmetric set in terms of permutation and sign change matrices:

$$[\boldsymbol{\lambda}] = \bigcup_{\boldsymbol{P} \in \mathrm{Perm}_m^{\mathrm{SC}}} \boldsymbol{P} \boldsymbol{\lambda},$$

where $\mathrm{Perm}_m^{\mathrm{SC}}$ is the collection of $m \times m$ matrices having exactly one non-zero element on each row and column, this element being either 1 or −1. Some fully symmetric sets are displayed in Fig. 1. The cardinality of a fully symmetric set $[\boldsymbol{\lambda}]$, generated by a generator vector $\boldsymbol{\lambda}$ containing r_0 zero generators and l distinct non-zero generators with multiplicities $r_1, \ldots, r_l$, is

$$\#[\boldsymbol{\lambda}] = \frac{2^{m-r_0} d!}{r_0! \cdots r_l!}. \tag{9}$$

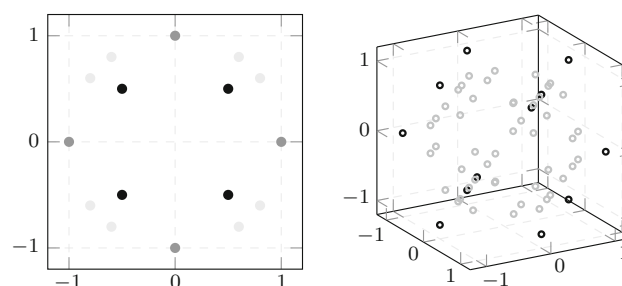**Fig. 1** Fully symmetric sets generated by the vectors [0.5, 0.5], [1, 0], and [0.6, 0.8] in $\mathbb{R}^2$ (left) and [1, 1, 0] and [0.2, 0.6, 0.8] in $\mathbb{R}^3$ (right)

Table 1 Sizes of fully symmetric sets generated by the generator vector $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_l, 0, \ldots, 0)$ having $l \leq m$ distinct non-zero elements $\lambda_1, \ldots, \lambda_l$ [see (9)]

	Dimension (m)					
	2	3	4	5	6	7
$l = 1$	4	6	8	10	12	14
$l = 2$	8	24	48	80	120	168
$l = 3$	–	48	192	480	960	1680
$l = 4$	–	–	384	1920	5760	13,440
$l = 5$	–	–	–	3840	23,040	80,640
$l = 6$	–	–	–	–	46,080	322,560
$l = 7$	–	–	–	–	–	645,120

See Table 1 for a number of examples in low dimensions.

For $\boldsymbol{\lambda} \in \mathbb{R}^m$ having non-negative elements, we occasionally need the concept of a *non-negative* fully symmetric set

$$[\boldsymbol{\lambda}]^+ := \bigcup_{\boldsymbol{P} \in \mathrm{Perm}_m} \boldsymbol{P} \boldsymbol{\lambda} \subset \mathbb{R}_+^m,$$

where $\mathrm{Perm}_m \subset \mathrm{Perm}_m^{\mathrm{SC}}$ is the collection of $m \times m$ permutation matrices.

2.2.2 Fully symmetric domains, kernels, and measures

At this point, we introduce several related definitions; these enable us later to state precisely which symmetry assumptions are being exploited.

Domains It will be assumed in the sequel that $M \subset \mathbb{R}^m$ is a *fully symmetric domain*, meaning that every fully symmetric set generated by a vector from M is contained in M: $[\boldsymbol{\lambda}] \subset M$ whenever $\boldsymbol{\lambda} \in M$. Equivalently, $M = \boldsymbol{P}M = \{\boldsymbol{P}\boldsymbol{x} : , \boldsymbol{x} \in M\}$ for any $\boldsymbol{P} \in \mathrm{Perm}_m^{\mathrm{SC}}$. Most popular domains, such as the whole of $\mathbb{R}^m$, hypercubes of the form $[-a, a]^m$ (from which, e.g. the unit hypercube can be obtained by simple translation and scaling), balls and spheres, are fully symmetric.

Kernels A kernel $k \colon M \times M \to \mathbb{R}$ defined on a fully symmetric domain M is said to be a *fully symmetric kernel* if $k(\boldsymbol{P}\boldsymbol{x}, \boldsymbol{P}\boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}')$ for any $\boldsymbol{P} \in \mathrm{Perm}_m^{\mathrm{SC}}$. Basic examples of fully symmetric kernels include isotropic kernels and products and sums of isotropic one-dimensional kernels.

Measures A measure ν on a fully symmetric domain M is a *fully symmetric measure* if it is invariant under fully symmetric pushforwards: $\boldsymbol{P}_*(\nu) = \nu$ for any $\boldsymbol{P} \in \mathrm{Perm}_m^{\mathrm{SC}}$. If ν admits a Lebesgue density p_ν, this condition is equivalent to $p_\nu(\boldsymbol{x}) = p_\nu(\boldsymbol{P}\boldsymbol{x})$ for any $\boldsymbol{P} \in \mathrm{Perm}_m^{\mathrm{SC}}$. Note that this is a narrow class of measures and a relaxation of this assumption is discussed in Sect. 5.

2.2.3 Fully symmetric cubature rules

The linear functional $\mu(f^\dagger) = \sum_{i=1}^N w_i f^\dagger(\boldsymbol{x}_i)$ is said to be *fully symmetric cubature rule* if its point set can be written as a union of a number $J \in \mathbb{N}$ of fully symmetric sets $[\boldsymbol{\lambda}^1], \dots, [\boldsymbol{\lambda}^J]$ and all points in each $[\boldsymbol{\lambda}^j]$ are assigned an equal weight. That is, a fully symmetric cubature rule is of the form

$$\mu(f^\dagger) = \sum_{j=1}^J w_j^{\mathrm{FS}} \sum_{\boldsymbol{x} \in [\boldsymbol{\lambda}^j]} f^\dagger(\boldsymbol{x})$$

for some weights $\boldsymbol{w}^{\mathrm{FS}} \in \mathbb{R}^J$ and generator vectors $\boldsymbol{\lambda}^1, \dots, \boldsymbol{\lambda}^J \in M$. Because this structure typically greatly simplifies design of the weights, many classical polynomial-based cubature rules are fully symmetric (McNamee and Stenger 1967; Genz 1986; Genz and Keister 1996; Lu and Darmofal 2004), including certain sparse grids (Novak and Ritter 1999; Novak et al. 1999)

2.3 Fully symmetric Bayesian cubature

The central aim of this article is to derive generalisations for the Bayes–Sard and multi-output Bayesian cubatures of the following result from Karvonen and Särkkä (2018), originally developed only for the standard Bayesian cubature method.

Theorem 1 *Consider the standard Bayesian cubature method based on a domain M, measure ν, and kernel k that are each fully symmetric and fix the mean function to be $m \equiv 0$. Suppose that the point set is a union of J fully symmetric sets: $X = \bigcup_{j=1}^J [\boldsymbol{\lambda}^j]$ for some distinct generator vectors $\Lambda = \{\boldsymbol{\lambda}^1, \dots, \boldsymbol{\lambda}^J\} \subset M$. Then the output of the standard Bayesian cubature method can be expressed in the fully symmetric form*

$$\mu_N(f^\dagger) = \sum_{j=1}^J w_{\Lambda, j} \sum_{\boldsymbol{x} \in [\boldsymbol{\lambda}^j]} f^\dagger(\boldsymbol{x}),$$

$$\sigma_N^2 = k_{\nu, \nu} - \sum_{j=1}^J w_{\Lambda, j} k_\nu(\boldsymbol{\lambda}^j) \#[\boldsymbol{\lambda}^j].$$

The weights $\boldsymbol{w}_\Lambda \in \mathbb{R}^J$ are the solution to the linear system $\boldsymbol{S}\boldsymbol{w}_\Lambda = \boldsymbol{k}_{\nu, \Lambda}$ of J equations, where

$$[\boldsymbol{S}]_{ij} = \sum_{\boldsymbol{x} \in [\boldsymbol{\lambda}^j]} k(\boldsymbol{\lambda}^i, \boldsymbol{x}) \quad \textit{and} \quad [\boldsymbol{k}_{\nu, \Lambda}]_j = k_\nu(\boldsymbol{\lambda}^j).$$

Theorem 1 demonstrates the principal idea; that one can exploit symmetry to reduce the number of kernel evaluations needed in the standard Bayesian cubature method from N^2 to NJ and decrease the number of equations in the linear system that needs to be solved from N to J. Since J is typically considerably smaller than $N = \sum_{j=1}^J \#[\boldsymbol{\lambda}^j]$, using fully symmetric sets results in a substantial reduction in computational cost. Numerical examples in Karvonen and Särkkä (2018) showed that sets containing up to tens of millions of points become feasible in the standard Bayesian cubature method when symmetry exploits are used. The aim of this paper is to generalise these techniques to the important cases of Bayes–Sard cubature (Sect. 3) and multi-output Bayesian cubature (Sect. 4).

Remark 1 If $\#[\boldsymbol{\lambda}^1] = \cdots = \#[\boldsymbol{\lambda}^J]$, the condition number of the matrix $\boldsymbol{S}$ in Theorem 1 cannot exceed that of $\boldsymbol{K}_X$ (similar results are available for the matrices in Theorems 2 and 3). This scenario occurs in, for instance, the numerical example of Sect. 6.3. To verify the claim, observe that by Lemma 4 $\boldsymbol{S}\boldsymbol{v} = \alpha \boldsymbol{v}$ implies that the block vector

$$\boldsymbol{v}' = \begin{bmatrix} v_1 \mathbf{1}_{\#[\boldsymbol{\lambda}^1]} \\ \vdots \\ v_J \mathbf{1}_{\#[\boldsymbol{\lambda}^J]} \end{bmatrix}$$

satisfies $\boldsymbol{K}_X \boldsymbol{v}' = \alpha \boldsymbol{v}'$. Consequently, the spectrum of $\boldsymbol{S}$ is a subset of that of $\boldsymbol{K}_X$. Furthermore, when $\#[\boldsymbol{\lambda}^1] = \cdots = \#[\boldsymbol{\lambda}^J]$, the matrix $\boldsymbol{S}$ is symmetric; therefore its condition number is the ratio of the largest and smallest eigenvalues. It follows that the condition number of $\boldsymbol{S}$ must be smaller or equal to that of $\boldsymbol{K}_X$.

3 Fully symmetric Bayes–Sard cubature

In this section, we first review the Bayes–Sard cubature method from Karvonen et al. (2018) and then derive a generalisation of Theorem 1 for this method.

3.1 Bayes–Sard cubature

In the standard Bayesian cubature method, the mean function m must be a priori specified. This requirement is relaxed in Bayes–Sard cubature (Karvonen et al. 2018), where a hierarchical approach is taken instead. Specifically, in Bayes–Sard cubature the prior mean function is given the parametric form

$$m_{\boldsymbol{\theta}}(\boldsymbol{x}) = \theta_1 \phi_1(\boldsymbol{x}) + \cdots + \theta_Q \phi_Q(\boldsymbol{x})$$
$$= \boldsymbol{\theta}^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{x}),$$

where $\boldsymbol{\phi}(\boldsymbol{x}) \in \mathbb{R}^Q$ has entries $[\boldsymbol{\phi}(\boldsymbol{x})]_i = \phi_i(\boldsymbol{x})$ and the parameter vector $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_Q) \in \mathbb{R}^Q$ represents coefficients in a pre-defined basis consisting of functions $\phi_i : M \to \mathbb{R}$, $i = 1, \ldots, Q$, that are assumed ν-integrable and that span a finite-dimensional linear function space $\pi := \mathrm{span}(\phi_1, \ldots, \phi_Q)$. That is, $m_{\boldsymbol{\theta}} \in \pi$ for any $\boldsymbol{\theta} \in \mathbb{R}^Q$. Then, for a positive-definite $\boldsymbol{\Sigma} \in \mathbb{R}^{Q \times Q}$, a Gaussian hyperprior distribution

$$\boldsymbol{\theta} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$$

is specified. The conditional distribution f_N of this field, based as before on data $\mathcal{D}$, is again Gaussian. In particular, when $\boldsymbol{\Sigma}^{-1} \to \mathbf{0}$ (meaning that the prior on $\boldsymbol{\theta}$ becomes improper, or weakly informative[1]) and assuming that $Q \leq N$, the posterior mean and variance take the forms

$$m_N(\boldsymbol{x}) = \boldsymbol{\alpha}^{\mathsf{T}} \boldsymbol{k}_X(\boldsymbol{x}) + \boldsymbol{\beta}^{\mathsf{T}} \boldsymbol{\phi}(\boldsymbol{x}), \tag{10}$$
$$k_N(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{k}_X(\boldsymbol{x})^{\mathsf{T}} \boldsymbol{K}_X^{-1} \boldsymbol{k}_X(\boldsymbol{x}')$$
$$+ [\boldsymbol{\Phi}_X^{\mathsf{T}} \boldsymbol{K}_X^{-1} \boldsymbol{k}_X(\boldsymbol{x}) - \boldsymbol{\phi}(\boldsymbol{x})]^{\mathsf{T}} [\boldsymbol{\Phi}_X^{\mathsf{T}} \boldsymbol{K}_X^{-1} \boldsymbol{\Phi}_X]^{-1}$$
$$\times [\boldsymbol{\Phi}_X^{\mathsf{T}} \boldsymbol{K}_X^{-1} \boldsymbol{k}_X(\boldsymbol{x}') - \boldsymbol{\phi}(\boldsymbol{x}')], \tag{11}$$

where $\boldsymbol{\Phi}_X \in \mathbb{R}^{N \times Q}$ with entries $[\boldsymbol{\Phi}_X]_{i,j} = \phi_j(\boldsymbol{x}_i)$ is called the *Vandermonde matrix* and the vectors $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are defined via the linear system

$$\begin{bmatrix} \boldsymbol{K}_X & \boldsymbol{\Phi}_X \\ \boldsymbol{\Phi}_X^{\mathsf{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_X^{\dagger} \\ \mathbf{0} \end{bmatrix}. \tag{12}$$

For there to exist a unique solution to (12), the Vandermonde matrix has to be of full rank. This technical condition, equivalent to the zero function being the only element of π vanishing on X, is known as π-*unisolvency* of the point set X. Throughout the article, we assume this is the case; see Wendland (2005, Section 2.2) or Karvonen et al. (2018, Supplement B) for more information and examples of unisolvent point sets.

The output of the *Bayes–Sard cubature* method is the posterior marginal distribution of the integral, namely

$$\int_M f_N(\boldsymbol{x}) \, \mathrm{d}\nu(\boldsymbol{x}) \sim N\big(\mu_N(f^{\dagger}), \sigma_N^2\big). \tag{13}$$

The mean and variance, obtained by integrating (10) and (11), are

$$\mu_N(f^{\dagger}) = (\boldsymbol{w}_X^k)^{\mathsf{T}} \boldsymbol{f}_X^{\dagger},$$
$$\sigma_N^2 = k_{\nu,\nu} - \boldsymbol{k}_{\nu,X}^{\mathsf{T}} \boldsymbol{K}_X^{-1} \boldsymbol{k}_{\nu,X}$$
$$+ (\boldsymbol{w}_X^{\pi})^{\mathsf{T}} \big(\boldsymbol{\Phi}_X^{\mathsf{T}} \boldsymbol{K}_X^{-1} \boldsymbol{k}_{\nu,X} - \boldsymbol{\phi}_{\nu}\big),$$

where $\boldsymbol{\phi}_{\nu} \in \mathbb{R}^Q$ has the entries $[\boldsymbol{\phi}_{\nu}]_i = \int_M \phi_i(\boldsymbol{x}) \mathrm{d}\nu(\boldsymbol{x})$ and the weight vectors $\boldsymbol{w}_X^k \in \mathbb{R}^N$ and $\boldsymbol{w}_X^{\pi} \in \mathbb{R}^Q$ are the solution to the linear system

$$\begin{bmatrix} \boldsymbol{K}_X & \boldsymbol{\Phi}_X \\ \boldsymbol{\Phi}_X^{\mathsf{T}} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_X^k \\ \boldsymbol{w}_X^{\pi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{k}_{\nu,X} \\ \boldsymbol{\phi}_{\nu} \end{bmatrix}. \tag{14}$$

The Bayes–Sard weights $\boldsymbol{w}_X^k$, like the standard Bayesian cubature weights, have a worst-case interpretation:

$$\boldsymbol{w}_X^k = \underset{\boldsymbol{w} \in \mathbb{R}^N}{\arg\min} \; \sup_{\|f^{\dagger}\|_{\mathcal{H}(k)} \leq 1} \left| \int_M f^{\dagger} \, \mathrm{d}\nu - \sum_{i=1}^N w_i f^{\dagger}(\boldsymbol{x}_i) \right|$$

subject to the linear constraints $\sum_{i=1}^N w_i \phi_j(\boldsymbol{x}_i) = I(\phi_j)$ for $j = 1, \ldots, Q$ (DeVore et al. 2018).

The Bayes–Sard method has some important theoretical and practical advantages over the standard Bayesian cubature method, which motivate us to study it in detail:

– The posterior mean $\mu_N(f^{\dagger})$ is exactly equal to the integral $I(f^{\dagger})$ if $f^{\dagger} \in \pi$. In particular, if π contains a non-zero constant function then $\sum_{i=1}^N w_{X,i}^k = 1$ so that the cubature rule is *normalised* (however, non-negativity of the weights is not guaranteed[2]). This can improve the stability of the method in high-dimensional settings Karvonen et al. (2018). In general, if π is the set of polynomials up to a certain order q, then the posterior mean is recognised as a cubature rule of algebraic degree q (Cools 1997, Definition 3.1).
– Given any cubature rule $\mu(f^{\dagger}) = \sum_{i=1}^N w_i f^{\dagger}(\boldsymbol{x}_i)$ for specified $w_i \in \mathbb{R}$ and $\boldsymbol{x}_i \in M$, and given any covariance function k, one can find an N-dimensional function space π such that $\mu_N = \mu$. Furthermore, the posterior standard deviation σ_N coincides with the worst-case error of the

---

[1] See Larkin (1974) and O'Hagan (1991) for slightly different earlier formulations where an improper prior is placed "directly" on $\boldsymbol{\theta}$.

[2] It is possible to employ a positivity constraint (Ehler et al. 2019), but in that case there is no convenient closed-form expression for the weights and the Bayesian interpretation is sacrificed.

cubature rule μ in the Hilbert space induced by k (Karvonen et al. 2018, Section 2.4). This demonstrates that any cubature rule can be interpreted as the posterior mean under an infinitude of prior models, providing a bridge between classical and Bayesian cubature methods.

The dimension of the linear system in (14) is $N + Q$. Thus the computational cost associated with the Bayes–Sard method is strictly greater than that of standard Bayesian cubature; at least $\mathcal{O}(N^3)$ in general. It is therefore of considerable practical interest to ask whether symmetry exploits can also be developed for the Bayes–Sard method.

3.2 A symmetry exploit for Bayes–Sard cubature

In this section, we present a novel result that enables fully symmetric sets to be exploited in the Bayes–Sard cubature method. In what follows we only consider a function space π spanned by even monomials exhibiting symmetries.[3] In practice, we do not believe this to be a significant restriction since polynomials typically serve as a good and functional default and, in fact, one retains considerable freedom in selecting the polynomials, not being restricted to, for example, spaces of all polynomials of at most a given degree.

Let $\pi_\alpha \subset \mathbb{N}_0^m$ denote a finite collection of multi-indices that in turn define the function space π:

$$\pi = \text{span}\{x^\alpha \, : \, \alpha \in \pi_\alpha\}.$$

Here x^α denotes the monomial $x_1^{\alpha_1} \times \cdots \times x_m^{\alpha_m}$. Define the index set

$$\mathbb{E}_0^m := \{\alpha \in \mathbb{N}_0^m \, : \, \alpha_i \text{ is even for every } i = 1, \ldots, m\}.$$

Our development will require that π_α is a union of $J_\alpha \in \mathbb{N}$ non-negative fully symmetric sets in $\mathbb{E}_0^m$. That is, $\alpha \in \pi_\alpha$ implies $P\alpha \in \pi_\alpha$ for any permutation matrix $P \in \text{Perm}_m$ and there exist distinct $\alpha^1, \ldots, \alpha^{J_\alpha} \in \mathbb{E}_0^m$ such that

$$\pi_\alpha = \bigcup_{j=1}^{J_\alpha} [\alpha^j]^+.$$

To prove a Bayes–Sard analogue of Theorem 1, we need four simple lemmas:

Lemma 1 *Suppose that M and ν are each fully symmetric. If $\alpha \in \mathbb{E}_0^m$ then $I(x^\alpha) = I(x^{P\alpha})$ for any $P \in \text{Perm}_m$.*

Proof First, observe that $(P^{-1}x)^\alpha = x^{P\alpha}$. By the change of variables formula of pushforwards and the assumption

$$P_*^{-1}(\nu) = \nu,$$

$$I(x^\alpha) = \int_M x^\alpha \, d\nu(x)$$
$$= \int_M x^\alpha \, dP_*^{-1}(\nu)(x)$$
$$= \int_M (P^{-1}x)^\alpha \, d\nu(x)$$
$$= \int_M x^{P\alpha} \, d\nu(x)$$
$$= I(x^{P\alpha})$$

for any $\alpha \in \mathbb{N}_0^m$. $\quad\square$

Lemma 2 *Suppose that M, ν, and k are each fully symmetric and let $\lambda \in M$. Then $k_\nu(x) = k_\nu(\lambda)$ for every $x \in [\lambda]$.*

Proof The proof is essentially identical to that of Lemma 1. $\quad\square$

Lemma 3 *Let $\lambda \in \mathbb{R}^m$ and $\alpha \in \mathbb{E}_0^m$. Then*

$$\sum_{\beta \in [\alpha]^+} x^\beta = \sum_{\beta \in [\alpha]^+} \lambda^\beta \qquad \text{for any} \quad x \in [\lambda], \qquad (15)$$

$$\sum_{x \in [\lambda]} x^\beta = \sum_{x \in [\lambda]} x^\alpha \qquad \text{for any} \quad \beta \in [\alpha]^+. \qquad (16)$$

Proof For any $\alpha \in \mathbb{E}_0^m$, $x \in [\lambda]$, and $P \in \text{Perm}_m^{\text{SC}}$,

$$\sum_{\beta \in [\alpha]^+} x^\beta = \sum_{\beta \in [\alpha]^+} (P^{-1}Px)^\beta = \sum_{\beta \in [\alpha]^+} (Px)^{P^+\beta},$$

where $P^+ \in \text{Perm}_m$ has the elements $[P^+]_{ij} = |[P]_{ij}|$ and the second equality follows from the fact that every element of β is even. Because $[P^+\alpha]^+ = [\alpha]^+$, it follows that $\sum_{\beta \in [\alpha]^+} x^\beta = \sum_{\beta \in [\alpha]^+} (Px)^\beta$. That is,

$$\sum_{\beta \in [\alpha]^+} x^\beta = \sum_{\beta \in [\alpha]^+} \lambda^\beta$$

since $\lambda = Px$ for some $P \in \text{Perm}_m^{\text{SC}}$. Consider then the "transpose" sum $\sum_{x \in [\lambda]} x^\beta$ for $\beta \in [\alpha]^+$. Similar arguments as above establish that

$$\sum_{x \in [\lambda]} x^\beta = \sum_{x \in [\lambda]} (Px)^{P\beta} = \sum_{x \in [\lambda]} x^{P\beta}$$

for any $P \in \text{Perm}_m$. Consequently,

$$\sum_{x \in [\lambda]} x^\beta = \sum_{x \in [\lambda]} x^\alpha$$

for every $\beta \in [\alpha]^+$. $\quad\square$

---

[3] Odd monomials come for "free"; see Remark 2.

Lemma 4 *Let* $\lambda, \lambda' \in \mathbb{R}^m$ *and suppose that the kernel k is fully symmetric. Then*

$$\sum_{x' \in [\lambda']} k(x, x') = \sum_{x' \in [\lambda']} k(\lambda, x') \quad \text{for any} \quad x \in [\lambda].$$

Proof For any $x \in [\lambda]$ there is $P_x \in \text{Perm}_m^{\text{SC}}$ such that $x = P_x \lambda$. Therefore

$$
\begin{aligned}
\sum_{x' \in [\lambda']} k(x, x') &= \sum_{x' \in [\lambda']} k(P_x \lambda, x') \\
&= \sum_{x' \in [\lambda']} k(P_x^{-1} P_x \lambda, P_x^{-1} x') \\
&= \sum_{x' \in [\lambda']} k(\lambda, P_x^{-1} x') \\
&= \sum_{x' \in [P_x^{-1} \lambda']} k(\lambda, x'),
\end{aligned}
$$

and the claim follows from the fact that $[P\lambda'] = [\lambda']$ for any $P \in \text{Perm}_m^{\text{SC}}$. $\qquad\square$

We are now ready to prove the main result of this section. Theorem 2 establishes sufficient conditions for the Bayes–Sard cubature rule to be fully symmetric and, in that case, provides an explicit simplification of its output (13).

Theorem 2 *Consider the Bayes–Sard cubature method based on a domain M, measure v, and kernel k that are each fully symmetric. Suppose that*

$$\pi = \text{span}\{x^{\alpha} : \alpha \in \pi_{\alpha}\} \quad \text{and} \quad \pi_{\alpha} = \bigcup_{j=1}^{J_{\alpha}} [\alpha^j]^+$$

for a collection $\mathcal{A} = \{\alpha^1, \ldots, \alpha^{J_{\alpha}}\} \subset \mathbb{E}_0^m$ of distinct even multi-indices and that X is a union of J distinct fully symmetric sets: $X = \bigcup_{j=1}^J [\lambda^j]$ for a collection $\Lambda = \{\lambda^1, \ldots, \lambda^J\} \subset M$ of distinct generator vectors. Then the output of the Bayes–Sard cubature method can be expressed in the fully symmetric form

$$\mu_N(f^{\dagger}) = \sum_{j=1}^J w_{\Lambda,j}^k \sum_{x \in [\lambda^j]} f^{\dagger}(x),$$

$$\sigma_N^2 = k_{v,v} - \sum_{j=1}^J w_{\Lambda,j}^{\sigma} k_v(\lambda^j) n_j^{\Lambda}$$

$$+ \sum_{j=1}^{J_{\alpha}} w_{\mathcal{A},j}^{\pi} n_j^{\mathcal{A}} \left[\sum_{i=1}^J w_{\Lambda,i}^{\sigma} \sum_{x \in [\lambda^i]} x^{\alpha^j} - I(x^{\alpha^j}) \right],$$

where $n_j^{\Lambda} = \#[\lambda^j]$, $n_j^{\mathcal{A}} = \#[\alpha^j]^+$, and $w_{\Lambda}^{\sigma} \in \mathbb{R}^J$ are the weights w_{Λ} in Theorem 1. The weights $w_{\Lambda}^k \in \mathbb{R}^J$ and $w_{\mathcal{A}}^{\pi} \in \mathbb{R}^{J_{\alpha}}$ form the solution to the linear system

$$\begin{bmatrix} S & A \\ B & 0 \end{bmatrix} \begin{bmatrix} w_{\Lambda}^k \\ w_{\mathcal{A}}^{\pi} \end{bmatrix} = \begin{bmatrix} k_{v,\Lambda} \\ \phi_{v,\mathcal{A}} \end{bmatrix} \tag{17}$$

of $J + J_{\alpha}$ equations, where $[k_{v,\Lambda}]_j = k_v(\lambda^j)$, $[\phi_{v,\mathcal{A}}]_j = I(x^{\alpha^j})$, $[S]_{ij} = \sum_{x \in [\lambda^j]} k(\lambda^i, x)$, $[A]_{ij} = \sum_{\beta \in [\alpha^j]^+} (\lambda^i)^{\beta}$, and $[B]_{ij} = \sum_{x \in [\lambda^j]} x^{\alpha^i}$.

Proof The linear system (17) is equivalent to

$$\sum_{j=1}^J w_{\Lambda,j}^k S_{ij} + \sum_{j=1}^{J_{\alpha}} w_{\mathcal{A},j}^{\pi} A_{ij} = k_v(\lambda^i) \text{ for } i \in \{1, \ldots, J\}$$

and

$$\sum_{j=1}^J w_{\Lambda,j}^k B_{ij} = I(x^{\alpha^i}) \quad \text{for} \quad i \in \{1, \ldots, J_{\alpha}\}.$$

These two groups of equations are equivalent, respectively, to the N equations (Lemmas 2 and 4 and (15))

$$\sum_{j=1}^J w_{\Lambda,j}^k \sum_{x' \in [\lambda^j]} k(x, x') + \sum_{j=1}^{J_{\alpha}} w_{\mathcal{A},j}^{\pi} \sum_{\beta \in [\alpha^j]^+} x^{\alpha} = k_v(x)$$

for $i \in \{1, \ldots, J\}$, $x \in [\lambda^i]$, and to the Q equations (Lemma 1 and (16))

$$\sum_{j=1}^J w_{\Lambda,j}^k \sum_{x' \in [\lambda^j]} (x')^{\alpha} = I(x^{\alpha})$$

for $\alpha \in \pi_{\alpha}$. From these two equations, we recognise that

$$w_X^k = \begin{bmatrix} w_{\Lambda,1}^k \mathbf{1}_{n_1^{\Lambda}} \\ \vdots \\ w_{\Lambda,J}^k \mathbf{1}_{n_J^{\Lambda}} \end{bmatrix} \quad \text{and} \quad w_X^{\pi} = \begin{bmatrix} w_{\mathcal{A},1}^k \mathbf{1}_{n_1^{\mathcal{A}}} \\ \vdots \\ w_{\mathcal{A},J_{\alpha}}^k \mathbf{1}_{n_{J_{\alpha}}^{\mathcal{A}}} \end{bmatrix}$$

solve the full Bayes–Sard weight system

$$\begin{bmatrix} K_X & \Phi_X \\ \Phi_X^{\mathsf{T}} & 0 \end{bmatrix} \begin{bmatrix} w_X^k \\ w_X^{\pi} \end{bmatrix} = \begin{bmatrix} k_{v,X} \\ \phi_v \end{bmatrix}.$$

The expression for the Bayes–Sard variance σ_N^2 can be obtained by first recognising that the unique elements of $K_X^{-1} k_{v,X}$ are precisely the weights w_{Λ} in Theorem 1, here denoted w_{Λ}^{σ}. Then we compute

$$k_{v,X}^{\mathsf{T}} K_X^{-1} k_{v,X} = \sum_{j=1}^{J} w_{\Lambda,j}^{\sigma} k_v(\lambda^j) n_j^{\Lambda}$$

and

$$(\boldsymbol{w}_X^{\pi})^{\mathsf{T}} \big(\boldsymbol{\Phi}_X^{\mathsf{T}} K_X^{-1} k_{v,X} - \boldsymbol{\phi}_v \big)$$

$$= (\boldsymbol{w}_X^{\pi})^{\mathsf{T}} \begin{bmatrix} \left(\sum_{j=1}^{J} w_{\Lambda_j}^{\sigma} \sum_{\boldsymbol{x} \in [\lambda]^j} \boldsymbol{x}^{\boldsymbol{\alpha}^1} - I\left(\boldsymbol{x}^{\boldsymbol{\alpha}^1}\right) \right) \mathbf{1}_{n_1^{\mathcal{A}}} \\ \vdots \\ \left(\sum_{j=1}^{J} w_{\Lambda_j}^{\sigma} \sum_{\boldsymbol{x} \in [\lambda]^j} \boldsymbol{x}^{\boldsymbol{\alpha}^{J_\alpha}} - I\left(\boldsymbol{x}^{\boldsymbol{\alpha}^{J_\alpha}}\right) \right) \mathbf{1}_{n_{J_\alpha}^{\mathcal{A}}} \end{bmatrix}$$

that, when expanded, yields the result. $\quad\square$

Remark 2 The polynomial space π could be appended with fully symmetric collections of odd polynomials (i.e. by using additional basis functions $\boldsymbol{x}^{\boldsymbol{\beta}}$, $\boldsymbol{\beta} \in [\boldsymbol{\alpha}]^+$ for $\boldsymbol{\alpha} \notin \mathbb{E}_0^m$). However, by doing this one gains nothing since the weights in $\boldsymbol{w}_{\mathcal{A}}^{\pi}$ corresponding to these basis functions turn out to be zero. This is quite easy to see from the easily proven facts that $\sum_{\boldsymbol{x} \in [\lambda]} \boldsymbol{x}^{\boldsymbol{\beta}} = 0$ and $I(\boldsymbol{x}^{\boldsymbol{\beta}}) = 0$ whenever $\boldsymbol{\beta} \notin \mathbb{E}_0^m$.

Just like Theorem 1 for the standard Bayesian cubature, Theorem 2 reduces the number of kernel and basis function evaluations from roughly $N^2 + Q^2$ to $NJ + NJ_\alpha$ and the size of the linear system that needs to be solved from $N + Q$ to $J + J_\alpha$. Typically, this translates to a significant computational speed-up; see Sect. 6.2 for a numerical example involving point sets of up to $N = 179,400$. Such results could not realistically be obtained by direct solution of the original linear system (14).

4 Fully symmetric multi-output Bayesian cubature

In this section, we review the multi-output Bayesian cubature method recently proposed by Xi et al. (2018) and show how to exploit fully symmetric sets in reducing computational complexity of this method.

4.1 Multi-output Bayesian cubature

One often needs to integrate a number of related integrands, $f_1^{\dagger}, \dots, f_D^{\dagger} \colon M \to \mathbb{R}$. It is of course trivial to treat these as a set of D independent integrals and apply either the standard Bayesian or Bayes–Sard cubature method to approximate each integral. However, in many cases the relationship between the integrands can be explicitly modelled and leveraged.

Such a setting can be handled by modelling a single vector-valued function $\boldsymbol{f}^{\dagger} := (f_1^{\dagger}, \dots, f_D^{\dagger}) \colon M \to \mathbb{R}^D$ as a vector-valued Gaussian field; full details can be found in Álvarez et al. (2012). In this case, the data $\mathcal{D}$ consist of evaluations

$$\boldsymbol{f}_{d,X_d}^{\dagger} = \big[f_d^{\dagger}(\boldsymbol{x}_{d1}), \dots, f_d^{\dagger}(\boldsymbol{x}_{dN}) \big]^{\mathsf{T}} \in \mathbb{R}^N$$

at points $X_d = \{\boldsymbol{x}_{d1}, \dots, \boldsymbol{x}_{dN}\} \subset M$ for each $d = 1, \dots, D$. In this section, we denote $X = \{X_d\}_{d=1}^{D}$. The assumption that each integrand is evaluated at N points is made only for notational simplicity; all results can be easily modified to accommodate different numbers of points for each integrand. Evaluations of each integrand are concatenated into the vector

$$\boldsymbol{f}_X^{\dagger} = \big[f_1^{\dagger}(\boldsymbol{x}_{11}), \dots, f_1^{\dagger}(\boldsymbol{x}_{1N}), \dots, \\ f_D^{\dagger}(\boldsymbol{x}_{D1}), \dots, f_D^{\dagger}(\boldsymbol{x}_{DN}) \big]^{\mathsf{T}} \in \mathbb{R}^{DN}.$$

In multi-output Bayesian cubature, the integrand is modelled as a vector-valued Gaussian field $\boldsymbol{f} \in \mathbb{R}^D$ characterised by vector-valued mean function $\boldsymbol{m} \colon M \to \mathbb{R}^D$ and matrix-valued covariance function $\boldsymbol{k} \colon M \times M \to \mathbb{R}^{D \times D}$. For notational simplicity, the prior mean function is fixed at $\boldsymbol{m} \equiv \boldsymbol{0}$. The conditional distribution $\boldsymbol{f}_N$ of this field, based on the data $\mathcal{D} = (X, \boldsymbol{f}_X^{\dagger})$, is also Gaussian with mean and covariance functions

$$\boldsymbol{m}_N(\boldsymbol{x}) = \boldsymbol{k}_X(\boldsymbol{x})^{\mathsf{T}} K_X^{-1} \boldsymbol{f}_X^{\dagger},$$
$$\boldsymbol{k}_N(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{k}_X(\boldsymbol{x})^{\mathsf{T}} K_X^{-1} \boldsymbol{k}_X(\boldsymbol{x}).$$

Here, in contrast to 3 and 4, all objects are of extended dimensions:

$$\boldsymbol{k}_X(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{k}_{X_1}(\boldsymbol{x}) \\ \vdots \\ \boldsymbol{k}_{X_D}(\boldsymbol{x}) \end{bmatrix} \in \mathbb{R}^{DN \times D},$$

$$K_X = \begin{bmatrix} K_{X_1,X_1}^{11} & \cdots & K_{X_1,X_D}^{1D} \\ \vdots & \ddots & \vdots \\ K_{X_D,X_1}^{D1} & \cdots & K_{X_D,X_D}^{DD} \end{bmatrix} \in \mathbb{R}^{DN \times DN},$$

where $\boldsymbol{k}_{X_d}(\boldsymbol{x})$ and K_{X_d,X_q}^{dq} are the $N \times D$ and $N \times N$ matrices

$$\boldsymbol{k}_{X_d}(\boldsymbol{x}) = \begin{bmatrix} [\boldsymbol{k}(\boldsymbol{x}_{d1}, \boldsymbol{x})]_{11} & \cdots & [\boldsymbol{k}(\boldsymbol{x}_{d1}, \boldsymbol{x})]_{1D} \\ \vdots & \ddots & \vdots \\ [\boldsymbol{k}(\boldsymbol{x}_{dN}, \boldsymbol{x})]_{11} & \cdots & [\boldsymbol{k}(\boldsymbol{x}_{dN}, \boldsymbol{x})]_{DD} \end{bmatrix},$$

$$K_{X_d,X_q}^{dq} = \begin{bmatrix} [\boldsymbol{k}(\boldsymbol{x}_{d1}, \boldsymbol{x}_{q1})]_{dq} & \cdots & [\boldsymbol{k}(\boldsymbol{x}_{d1}, \boldsymbol{x}_{qN})]_{dq} \\ \vdots & \ddots & \vdots \\ [\boldsymbol{k}(\boldsymbol{x}_{dN}, \boldsymbol{x}_{q1})]_{dq} & \cdots & [\boldsymbol{k}(\boldsymbol{x}_{dN}, \boldsymbol{x}_{qN})]_{dq} \end{bmatrix}.$$

The output of the *multi-output* (or *vector-valued*) Bayesian cubature method is a D-dimensional Gaussian random vector:

$$\int_M \boldsymbol{f}_N(\boldsymbol{x}) \mathrm{d}\nu(\boldsymbol{x}) \sim N\big(\boldsymbol{\mu}_N(\boldsymbol{f}^{\dagger}), \boldsymbol{\Sigma}_N \big)$$

with

$$\boldsymbol{\mu}_N(\boldsymbol{f}^\dagger) = \boldsymbol{k}_{\nu,X}^\mathsf{T} \boldsymbol{K}_X^{-1} \boldsymbol{f}_X^\dagger, \tag{18}$$

$$\boldsymbol{\Sigma}_N = \boldsymbol{k}_{\nu,\nu} - \boldsymbol{k}_{\nu,X}^\mathsf{T} \boldsymbol{K}_X^{-1} \boldsymbol{k}_{\nu,X}, \tag{19}$$

where $\boldsymbol{k}_{\nu,X} = \int_M \boldsymbol{k}_X(\boldsymbol{x}) \, \mathrm{d}\nu(\boldsymbol{x}) \in \mathbb{R}^{DN \times D}$ and $\boldsymbol{k}_{\nu,\nu} = \int_M \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x}') \, \mathrm{d}\nu(\boldsymbol{x}) \, \mathrm{d}\nu(\boldsymbol{x}') \in \mathbb{R}^{D \times D}$. Equivalently, the posterior mean and variance can be written in terms of the weights

$$\boldsymbol{W}_X = \boldsymbol{K}_X^{-1} \boldsymbol{k}_{\nu,X} = [\boldsymbol{W}_1^\mathsf{T} \cdots \boldsymbol{W}_D^\mathsf{T}]^\mathsf{T} \in \mathbb{R}^{DN \times D}, \tag{20}$$

where $\boldsymbol{W}_d \in \mathbb{R}^{N \times D}$. For example, mean of the dth integral then takes the form

$$\mu_N(f_d^\dagger) = [\boldsymbol{\mu}_N(\boldsymbol{f}^\dagger)]_d = \sum_{q=1}^D \sum_{i=1}^N [\boldsymbol{W}_q]_{id} f_q^\dagger(\boldsymbol{x}_{qi}). \tag{21}$$

If the dth integrand is modelled as independent of all the other integrands, the posterior mean (21) reduces to the standard Bayesian cubature posterior mean (6).

4.2 Separable kernels

The structure of matrices appearing in the multi-output Bayesian cubature equations can be simplified when the multi-output kernel is *separable*. This means that there is a positive-definite $\boldsymbol{B} \in \mathbb{R}^{D \times D}$ such that

$$\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{B} c(\boldsymbol{x}, \boldsymbol{x}') \tag{22}$$

for some positive-definite kernel $c \colon M \times M \to \mathbb{R}$. The matrices $\boldsymbol{K}_X$ and $\boldsymbol{k}_{\nu,X}$ now assume the simplified forms

$$\boldsymbol{k}_{\nu,X} = \begin{bmatrix} B_{11} \boldsymbol{c}_{\nu,X_1} & \cdots & B_{1D} \boldsymbol{c}_{\nu,X_D} \\ \vdots & \ddots & \vdots \\ B_{D1} \boldsymbol{c}_{\nu,X_1} & \cdots & B_{DD} \boldsymbol{c}_{\nu,X_D} \end{bmatrix},$$

$$\boldsymbol{K}_X = \begin{bmatrix} B_{11} \boldsymbol{C}_{X_1,X_1} & \cdots & B_{1D} \boldsymbol{C}_{X_1,X_D} \\ \vdots & \ddots & \vdots \\ B_{D1} \boldsymbol{C}_{X_D,X_1} & \cdots & B_{DD} \boldsymbol{C}_{X_D,X_D} \end{bmatrix},$$

where $[\boldsymbol{c}_{\nu,X_d}]_i = c_\nu(\boldsymbol{x}_{di})$ and $[\boldsymbol{C}_{X_d,X_q}]_{ij} = c(\boldsymbol{x}_{di}, \boldsymbol{x}_{qj})$. However, even with the simplified structure afforded by the use of separable kernels, the implementation of multi-output Bayesian cubature remains computationally challenging, calling for some $(DN)^2$ kernel evaluations and solution to a linear system of dimension DN. This is problematic if a large number of integrands is to be handled simultaneously. The next section demonstrates how fully symmetric points sets can be exploited to reduce this cost.

Remark 3 Note that using the same point set X' for each integrand yields immediate computational simplification, since in this case the above matrices can be written as Kronecker products:

$$\boldsymbol{k}_{\nu,X} = \boldsymbol{B} \otimes \boldsymbol{c}_{\nu,X'} \quad \text{and} \quad \boldsymbol{K}_X = \boldsymbol{B} \otimes \boldsymbol{C}_{X',X'}.$$

However, this case is of little practical interest because, by the properties of the Kronecker product,

$$\boldsymbol{W}_X = \boldsymbol{I}_D \otimes \boldsymbol{w}_{X'},$$

where $\boldsymbol{w}_{X'} \in \mathbb{R}^N$ are the standard Bayesian cubature weights (8) for the covariance function c and points X' (Xi et al. 2018, Supplements B and C.1). That is, the integral estimates $\boldsymbol{\mu}_N(\boldsymbol{f}^\dagger)$ reduce to those given by the standard Bayesian cubature method applied independently to each integral.

4.3 A symmetry exploit for multi-output Bayesian cubature

Our main result in this section is a second generalisation of Theorem 1, in this case for the multi-output Bayesian cubature method.

Theorem 3 *Consider the multi-output Bayesian cubature method based on a separable matrix-valued kernel $\boldsymbol{k}$. Let the domain M, measure ν, and uni-output kernel c each be fully symmetric and fix the mean function to be $\boldsymbol{m} \equiv \boldsymbol{0}$. Suppose that each X_d is a union of J fully symmetric sets: $X_d = \bigcup_{j=1}^J [\boldsymbol{\lambda}^{dj}]$ for some $\Lambda_d = \{\boldsymbol{\lambda}^{d1}, \ldots, \boldsymbol{\lambda}^{dJ}\} \subset M$ such that $n_j^\Lambda = \#[\boldsymbol{\lambda}^{dj}]$ does not depend on d and, consequently, $\#X_d = N$ for each $d = 1, \ldots, D$. Then the output of the multi-output Bayesian cubature method can be expressed in the fully symmetric form*

$$[\boldsymbol{\mu}_N(\boldsymbol{f}^\dagger)]_d = \sum_{q=1}^D \sum_{j=1}^J [\boldsymbol{W}_{\Lambda,q}]_{jd} \sum_{\boldsymbol{x} \in [\boldsymbol{\lambda}^{qj}]} f_q^\dagger(\boldsymbol{x}), \tag{23}$$

$$\boldsymbol{\Sigma}_N = \boldsymbol{B} c_{\nu,\nu} - \sum_{d=1}^D \sum_{j=1}^J \boldsymbol{P}_{dj} \boldsymbol{B}_j^{\mathrm{diag}}, \tag{24}$$

where $\boldsymbol{B}_j^{\mathrm{diag}}$ is the diagonal $D \times D$ matrix formed out of the jth row of $\boldsymbol{B}$ and

$$\boldsymbol{P}_{dj} = \begin{bmatrix} [\boldsymbol{W}_{\Lambda,d}]_{j1} c_\nu(\boldsymbol{\lambda}^{1j}) n_j^\Lambda & \cdots & [\boldsymbol{W}_{\Lambda,d}]_{j1} c_\nu(\boldsymbol{\lambda}^{Dj}) n_j^\Lambda \\ \vdots & \ddots & \vdots \\ [\boldsymbol{W}_{\Lambda,d}]_{jD} c_\nu(\boldsymbol{\lambda}^{1j}) n_j^\Lambda & \cdots & [\boldsymbol{W}_{\Lambda,d}]_{jD} c_\nu(\boldsymbol{\lambda}^{Dj}) n_j^\Lambda \end{bmatrix}.$$

The weight matrix

$$\boldsymbol{W}_\Lambda = [\boldsymbol{W}_{\Lambda,1}^T \cdots \boldsymbol{W}_{\Lambda,D}^T]^T \in \mathbb{R}^{DJ \times D}, \quad \boldsymbol{W}_{\Lambda,d} \in \mathbb{R}^{J \times D},$$

is the solution to the linear system $SW_\Lambda = k_{\nu,\Lambda}$, where

$$S = \begin{bmatrix} B_{11}S_{11} & \cdots & B_{1D}S_{1D} \\ \vdots & \ddots & \vdots \\ B_{D1}S_{D1} & \cdots & B_{DD}S_{DD} \end{bmatrix} \in \mathbb{R}^{DJ \times DJ},$$

$$[S_{dq}]_{ij} = \sum_{x \in [\lambda^{qj}]} c(\lambda^{di}, x),$$

$$k_{\nu,\lambda} = \begin{bmatrix} B_{11}c_{\nu,\Lambda_1} & \cdots & B_{1D}c_{\nu,\Lambda_D} \\ \vdots & \ddots & \vdots \\ B_{D1}c_{\nu,\Lambda_1} & \cdots & B_{DD}c_{\nu,\Lambda_D} \end{bmatrix} \in \mathbb{R}^{DJ \times D},$$

$$[c_{\nu,\Lambda_d}]_j = c_\nu(\lambda^{dj}).$$

Proof The matrix equation $SW_\Lambda = k_{\nu,\Lambda}$ corresponds to the $D^2 J$ equations

$$\sum_{q=1}^{D} B_{dq} \sum_{i=1}^{J} [S_{dq}]_{ji} [W_{\Lambda,q}]_{id'} = B_{dd'} c_\nu(\lambda^{d'j})$$

for $(d, d', j) \in \{1, \ldots, d\}^2 \times \{1, \ldots, J\}$. In turn, through Lemmas 2 and 4, these are equivalent to

$$\sum_{q=1}^{D} B_{dq} \sum_{i=1}^{J} [W_{\Lambda,q}]_{id'} \sum_{x \in [\lambda^{qi}]} c(x_{dj'}, x) = B_{dd'} c_\nu(x_{d'j'})$$

for $(d, d') \in \{1, \ldots, D\}^2$ and $j' = 1, \ldots, n_j^\Lambda$, $j = 1, \ldots, J$. There are a total of $D^2 \sum_{j=1}^{J} n_j^\Lambda = D^2 N$ of these equations. The weights

$$W_d = \begin{bmatrix} [W_{\Lambda,d}]_{11}\mathbf{1}_{n_1^\Lambda} & \cdots & [W_{\Lambda,d}]_{1D}\mathbf{1}_{n_1^\Lambda} \\ \vdots & \ddots & \vdots \\ [W_{\Lambda,d}]_{J1}\mathbf{1}_{n_J^\Lambda} & \cdots & [W_{\Lambda,d}]_{JD}\mathbf{1}_{n_J^\Lambda} \end{bmatrix}$$

in (20) are then seen to solve the full matrix equation $K_X W = k_{\nu,X}$. The expressions for the posterior mean and variance follow from straightforward manipulation of (18) and (19). $\square$

The computational complexity of forming the fully symmetric weight matrix W_Λ is dominated by the DJN kernel evaluations needed to form S and the inversion of this $DJ \times DJ$ matrix. Due to J often being orders of magnitude smaller than N, these tasks remain feasible even for a very large total number of points DN. For example, in Sect. 6.3 the result of Theorem 3 is applied to facilitate the simultaneous computation of up to $D = 50$ integrals arising in a global illumination problem, each integrand being evaluated at up to $N = 288$ points. Such results can barely be obtained by direct solution of the original linear system in (20).

5 Symmetric change of measure

The results presented in this article, and those originally described in Karvonen and Särkkä (2018), rely on the assumption that the measure ν is fully symmetric (see Sect. 2.2.2). This is a strong restriction; most measures are not fully symmetric. However, this assumption can be avoided in a relatively straightforward manner, which is now described.

Suppose that M is a fully symmetric domain and that ν is an arbitrary measure, admitting a density p_ν, against which the function $f^\dagger \colon M \to \mathbb{R}$ is to be integrated. Further suppose that there is a fully symmetric measure ν_* on M such that ν is absolutely continuous with respect to ν_*, and therefore admits a density p_{ν_*} such that the Radon–Nikodym derivative $d\nu/d\nu_* = p_\nu(x)/p_{\nu_*}(x)$ is well-defined. Then the integral of interest can be re-written as an integral with respect to the fully symmetric measure ν_*:

$$\int_M f^\dagger(x) \, d\nu(x) = \int_M f_*^\dagger(x) \, d\nu_*(x),$$

$$f_*^\dagger(x) := f^\dagger(x) \frac{p_\nu(x)}{p_{\nu_*}(x)}.$$

Note that the existence of the second integral follows from the Radon–Nikodym property and the monotone convergence theorem. Thus, the assumption of a fully symmetric measure ν in the statement of Theorems 1, 2, and 3 is not overly restrictive. This *symmetric change of measure* technique is demonstrated on a numerical example in Sect. 6.4.

Remark 4 Note that the situation here is unlike standard importance sampling (Robert and Casella 2013, Section 3.3), in that the importance distribution ν_* is required to be fully symmetric. As such, it seems not obvious how to mathematically characterise an "optimal" choice of ν_*. Indeed, any notion of optimality ought also depend on the cubature method that will be used. Nevertheless, obvious constructions (e.g. the choice of ν_* as an isotropic centred Gaussian for ν sub-Gaussian and $M = \mathbb{R}^m$) can work rather well.

6 Results

In this section, we assess the performance of the fully symmetric Bayes–Sard and fully symmetric multi-output Bayesian cubature methods based on computational simplifications provided in Theorems 2 and 3. MATLAB code for all examples is provided at https://github.com/tskarvone/bc-symmetry-exploits.

6.1 Selection of fully symmetric sets

The choice of generator vectors $\Lambda = \{\boldsymbol{\lambda}^1, \dots, \boldsymbol{\lambda}^J\}$ for a fully symmetric point set is practically important and has not yet been discussed. In principle one may wish to select Λ in order to minimise a criterion, such as the posterior standard deviation σ_N. However, it appears that such optimal Λ are mathematically intractable in general. Moreover, numerical optimisation methods cannot be naively applied to approximate the optimal Λ, since in high dimensions a sparsity structure in the generator vectors $\boldsymbol{\lambda}^j$ is required to prevent creation of massive point sets $[\boldsymbol{\lambda}^j]$. Thus, although we cannot provide definitive guidelines on how to select the generators in the setting of this article, there are some useful heuristics that have guided us in the examples to follow and those presented in Karvonen and Särkkä (2018, Section 5):

- In low dimensions, say $m \leq 4$, it is feasible to use (quasi) Monte Carlo samples as generators, as each fully symmetric set will contain at most 384 points (see Table 1). However, a large number of fully symmetric sets may be needed to ensure sufficient coverage of the space. This approach can work, as in Sect. 6.3, but is occasionally prone to failure (Karvonen and Särkkä 2018, Section 5.3).
- In higher dimensions (or when a more robust design is desired), we recommend selecting a tried-and-tested fully symmetric point set, such as a sparse grid (Holtz 2011, Chapter 4). This can then be further modified if required, since fully symmetric sets can be added or removed at will. In very high dimensions, this can amount to using effectively low-dimensional generator vectors of the forms $(x_1, 0, \dots, 0)$, $(x_2, 0, \dots, 0)$, $(x_1, x_2, 0, \dots, 0)$ and so on, for points x_i that come from some classical one-dimensional integration rule, such as Gauss–Hermite or Clenshaw–Curtis.

These principles guided our choice of fully symmetric point sets in the sequel.

6.2 Zero coupon bonds

This example involves a model for zero coupon bonds that has been used to assess accuracy and robustness of the Bayes–Sard cubature and fully symmetric Bayesian cubature methods in Karvonen and Särkkä (2018) and Karvonen et al. (2018).

6.2.1 Integration problem

The integral of interest, arising from Euler–Maruyama discretisation of the Vasicek model, is

$$
\begin{aligned}
P(0, T) &:= \mathbb{E}\left[\exp\left(-\Delta t \sum_{i=0}^{T-1} r_{t_i}\right)\right] \\
&= \exp(-\Delta t r_{t_0})\mathbb{E}\left[\exp\left(-\Delta t \sum_{i=1}^{T-1} r_{t_i}\right)\right],
\end{aligned}
\tag{25}
$$

where r_{t_i} are particular Gaussian random variables and Δt and r_{t_0} are parameters of the integrand. The dimension $m = T - 1$ of the integrand can be freely selected and the integral admits a convenient closed-form solution; see Holtz (Holtz 2011, Section 6.1) or Karvonen and Särkkä (2018, Section 5.5) for a more complete description of this benchmark integral.

6.2.2 Setting

The accuracy of the standard Bayesian cubature and Bayes–Sard cubature methods was compared, for computing the integral (25) in a setting identical to that of Karvonen and Särkkä (2018, Section 5.5). In particular, the same parameter values and point set (a sparse grid based on a certain Gauss–Hermite sequence with the origin removed), were used. The kernel was the Gaussian kernel with length-scale $\ell > 0$:

$$
k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{x}'\|^2}{2\ell^2}\right).
\tag{26}
$$

Accuracy of the two cubature methods was assessed for the heuristic length-scale choices $\ell = m$ and $\ell = \sqrt{m}$. The linear space π in the Bayes–Sard method, defined by the collection π_α of multi-indices, was taken to be $\pi_\alpha = \{\boldsymbol{\alpha} : |\alpha| \leq r\}$ for either $r = 1$ (linear) or $r = 2$ (quadratic) polynomials. The dimension T ranged between 20 and 300. Since the number of points in a sparse grid depends on the dimension, the maximal N used was $179,400$. Theorems 1 and 2 facilitated the computation, respectively, of the standard Bayesian cubature and Bayes–Sard cubature method. Note that, in the results that are presented next, even though N increases, no convergence (or necessarily monotonicity of the error) is to be expected because the integration problem becomes more difficult as T is increased.

6.2.3 Results

The results are depicted in Fig. 2. We observe that Bayes–Sard method is much less sensitive to the length-scale choice compared to the standard Bayesian cubature method. For instance, the selection $\ell = \sqrt{m}$ has Bayes–Sard outperform the standard Bayesian cubature by roughly three orders of magnitude. It is also clear that, in this particular problem, the addition of more polynomial basis functions can significantly improve the integral estimates.
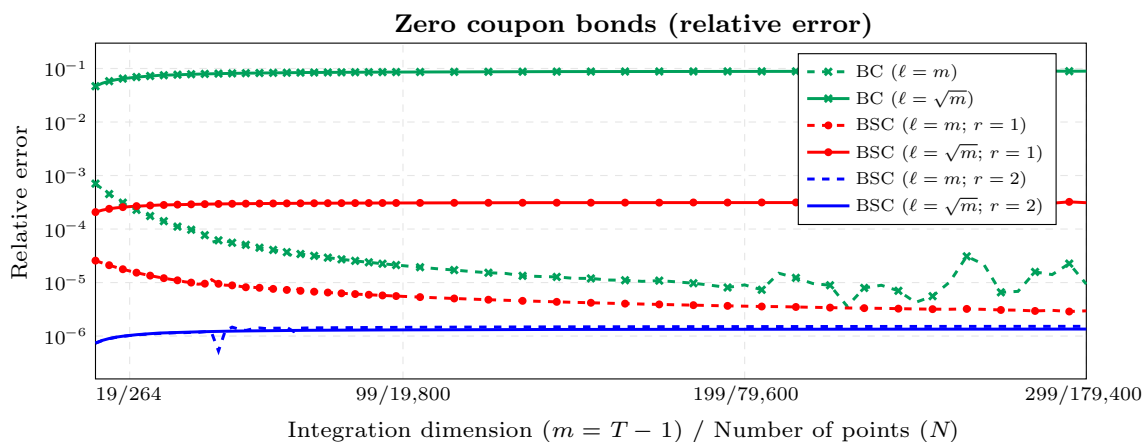
Fig. 2 Numerical computation of the integral (25) using fully symmetric Bayesian cubature (BC) and Bayes–Sard cubature (BSC) for different choices of the length-scale ℓ and polynomial degree r of the parametric function space π used in BSC

It was not possible to obtain results at this scale in the earlier work of Karvonen et al. (2018), where the largest value of N considered was 5,000. In contrast, our result in Theorem 2 enabled point sets of size up to $N = 179,400$ to be used. The computational time required to produce the results for the Bayes–Sard cubature in the most demanding case, $T = 300$ and $m = 2$, was on the order of 2.5 min on a standard laptop computer. However, this can be mostly attributed to a suboptimal algorithm for generating the sparse grid. Indeed, after the points had been obtained it took roughly one second to compute the Bayes–Sard weights.

6.3 Global illumination integrals

Next we considered the multi-output Bayesian cubature method, together with the symmetry exploit developed in Sect. 4.1, to compute a collection of closely related integrals arising in a global illumination context. This is a popular application of Bayesian cubature methods; see Brouillat et al. (2009), Marques et al. (2013, 2015), Briol et al. (2019) and Xi et al. (2018) for existing work. In particular, multi-output Bayesian cubature was applied to the problem that we consider below in Xi et al. (2018), where $D = 5$ integrals were simultaneously computed. Through computational simplifications obtained by using fully symmetric sets, in what follows we simultaneously compute up to $D = 50$ integrals, a tenfold improvement.

6.3.1 Integration problem

Global illumination is concerned with the rendering of glossy objects in a virtual environment (Dutre et al. 2006). The integration problem studied here is to compute the *outgoing radiance* $L_0(\omega_o)$ in the direction ω_o, for different values of the observation angle ω_o. In practical terms, this represents

the amount of light travelling from the object to an observer at an observation angle ω_o. The need for simultaneous computation for different ω_o can arise when the observation angle is rapidly changing, for example as the player moves in a video game context. The outgoing radiance is given by the integral

$$L_0(\omega_0) = L_e(\omega_0) + \int_{\mathbb{S}^2} L_i(\omega_i)\rho(\omega_i, \omega_o)[\omega_i^\mathsf{T} n]_+ \, d\nu(\omega_i)$$

with respect to the uniform (i.e. Riemannian) measure ν on the unit sphere

$$\mathbb{S}^2 = \left\{ x \in \mathbb{R}^3 : \|x\| = 1 \right\} \subset \mathbb{R}^3.$$

Here $L_e(\omega_o)$ is the amount of light emitted by the object itself, essentially a constant, while $L_i(\omega_i)$ is the amount of light being reflected from the object, originating from angle $\omega_i \in \mathbb{S}^2$. That reflection is impossible from a reflexive angle is captured by the term $[\omega_i^\mathsf{T} n]_+ := \max\{0, \omega_i^\mathsf{T} n\}$ with n the unit normal to the object. That light is reflected less efficiently at larger incidence angles is captured by a *bidirectional reflectance distribution* function

$$\rho(\omega_i, \omega_o) = \frac{1}{2\pi} \exp\left(\omega_i^\mathsf{T}\omega_o - 1\right).$$

Evaluation of $L_i(\omega_i)$ involves a call to an *environment map* [in this case, a picture of a lake in California; see Briol et al. (2019)], which is associated with a computational communication cost. The illumination integral must be computed for each of the red, green, and blue (RGB) colour channels; we treat the integration problems corresponding to different colour channels as statistically independent.
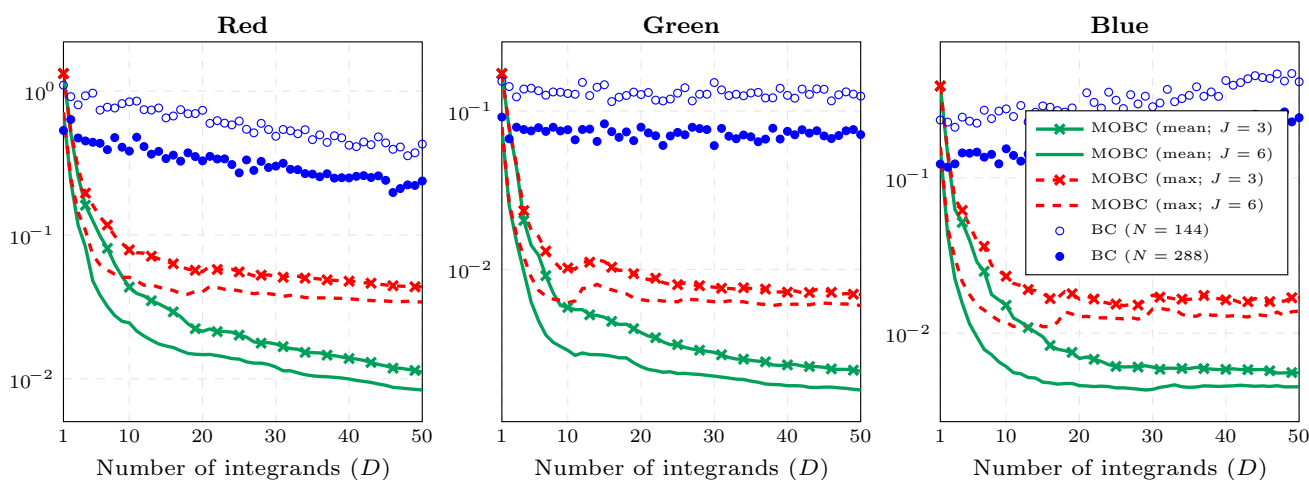
Fig. 3 The mean [green; Eq. (28)] and maximal [red; Eq. (29)] relative integration errors obtained when simultaneously approximating D global illumination integrals (27) using Bayesian cubature (BC) with random points and fully symmetric multi-output Bayesian cubature (MOBC). Here $J = 3$ and $J = 6$ random generator vectors were used to produce a fully symmetric point set of size $N = 144$ (for $J = 3$) and $N = 288$ (for $J = 6$). The displayed results have been averaged over 100 independent realisations of the point sets. (Color figure online)

6.3.2 Setting

The performance of the standard Bayesian cubature and multi-output Bayesian cubature methods was assessed on a collection of D related integrals, where D was varied up to a maximum of $D_{\max} = 50$. The integrands were indexed by observation angles $\boldsymbol{\omega}_o^d$ with a fixed azimuth and elevation ranging uniformly on the interval $[\frac{\pi}{4} - \frac{\pi}{24}, \frac{\pi}{4} + \frac{\pi}{24}]$:

$$\boldsymbol{\omega}_o^d := \left(0, \frac{\pi}{4} - \frac{\pi}{24}\left[1 - 2\left(\frac{d-1}{D_{\max} - 1}\right)\right]\right).$$

To formulate the problem in the multi-output framework, we define the associated integrands

$$f_d^\dagger = L_i(\boldsymbol{\omega}_i)\rho(\boldsymbol{\omega}_i, \boldsymbol{\omega}_o^d)[\boldsymbol{\omega}_i^\mathsf{T}\boldsymbol{n}]_+$$

for $d = 1, \ldots, D_{\max}$. The aim is then to compute the integrals

$$I(f_d^\dagger) := \int_{\mathbb{S}^2} f_d^\dagger(\boldsymbol{\omega}_i) \, \mathrm{d}\nu(\boldsymbol{\omega}_i). \tag{27}$$

In our experiments, a separable vector-valued covariance function was used, defined as in (22) with

$$c(\boldsymbol{x}, \boldsymbol{x}') = \frac{8}{3} - \|\boldsymbol{x} - \boldsymbol{x}'\|, \qquad [\boldsymbol{B}]_{dq} = \exp\left((\boldsymbol{\omega}_o^d)^\mathsf{T}\boldsymbol{\omega}_o^q - 1\right).$$

This prior structure is identical to that used in Briol et al. (2019) and Xi et al. (2018) and corresponds to assuming that the integrand belongs to a Sobolev space of smoothness $\frac{3}{2}$. The kernel c has tractable kernel means: $c_\nu(\boldsymbol{x}) = \frac{4}{3}$ for every $\boldsymbol{x} \in \mathbb{S}^2$ and $c_{\nu,\nu} = \frac{4}{3}$.

In order to exploit Theorems 1 and 3, we need to restrict to fully symmetric point sets on $\mathbb{S}^2$. To obtain such sets, we followed the method proposed in Karvonen and Särkkä (2018, Section 5.3). That is, we draw, for each $d = 1, \ldots, D$, either $J = 3$ or $J = 6$ independent generator vectors from the uniform distribution ν on $\mathbb{S}^2$ and use these to generate distinct fully symmetric point sets $X_1, \ldots, X_{D_{\max}} \subset \mathbb{S}^2$. Equation (9) implies that $N = 3 \times 48 = 144$ or $N = 6 \times 48 = 288$.[4] This approach to generation of a point set was selected for its simplicity, our main focus being on the multi-output framework and a large number of integrals D. Alternative point sets on $\mathbb{S}^2$ are numerous, such as rotated adaptations of numerically computed approximations to the optimal quasi Monte Carlo designs developed in Brauchart et al. (2014).

6.3.3 Results

The results are depicted in Fig. 3 in terms of the relative integration error for each RGB colour channel. For $D = 1, \ldots, D_{\max}$, define the vector-valued functions

$$\boldsymbol{f}^{\dagger,D} = (f_1^\dagger, \ldots, f_D^\dagger) \colon \mathbb{S}^2 \to \mathbb{R}^D.$$

The figure shows the improvement in integration accuracy when D increases and more integrands are considered simultaneously. Displayed are the mean

$$\frac{1}{D}\sum_{d=1}^{D} \frac{\left|I(f_d^\dagger) - [\boldsymbol{\mu}_N(\boldsymbol{f}^{\dagger,D})]_d\right|}{I(f_d^\dagger)} \tag{28}$$

---

[4] The elements of each random generator vector are almost surely non-zero and distinct.
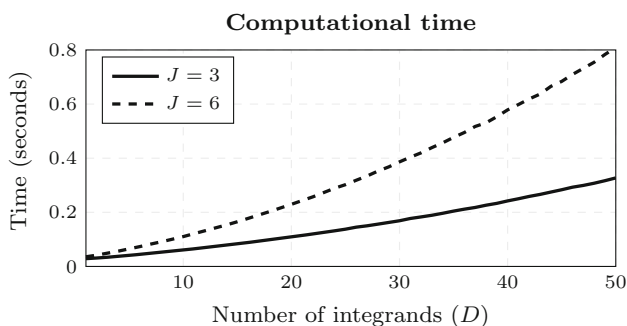
Fig. 4 Average computational time (over 100 independent runs), integrand evaluations included, for computation of the fully symmetric multi-output Bayesian cubature estimates for one colour channel. The algorithm was implemented in MATLAB and run on a desktop computer with an Intel Xeon 3.40 GHz processor and 15 GB of RAM

and maximal

$$\max_{d=1,\dots,D} \frac{\left| I(f_d^\dagger) - [\boldsymbol{\mu}_N(\boldsymbol{f}^{\dagger,D})]_d \right|}{I(f_d^\dagger)} \tag{29}$$

relative errors for $D = 1, \dots, D_{\max}$. For comparison, the figure also contains results for the standard Bayesian cubature method, applied separately to each of the uni-output integrands $f_d^\dagger$. Each of the reference integrals $I(f_d^\dagger)$ was computed using brute force Monte Carlo, with 10 million points used.

In accordance with Xi et al. (2018), we observed that the multi-output Bayesian cubature method is superior to the standard one already when $D = 5$. The performance gain of the multi-output method keeps increasing when more integrands are added but is ultimately bounded. This is reasonable since integrands for wildly different $\boldsymbol{\omega}_o^d$ can convey little information about each other. For the smallest values of D the multi-output method is less accurate than the standard Bayesian cubature method. This can be explained by potential non-uniform covering of the unit sphere when the total number DJ of fully symmetric sets is low (e.g. when some of the generator vectors happen to cluster, the fully symmetric sets they generated do not greatly differ, so that less information is obtained on the integrand). For instance, the standard deviation over the 100 runs in the relative error of fully symmetric Bayesian cubature for the first integral (i.e. the case $D = 1$ in Fig. 3) was 0.34 ($J = 3$) or 0.17 ($J = 6$) while that of the standard Bayesian cubature with random points was only 0.19 ($N = 144$) or 0.11 ($N = 288$). See also Karvonen and Särkkä (2018, Figure 5.1).

Computational times remained reasonable throughout this experiment; see Fig. 4. For example, without symmetry exploits, the case $D = 50$ and $J = 6$ would require $(DN)^2 = 207,360,000$ kernel evaluations and inversion of a 14,400-dimensional matrix while Theorem 3 reduces these numbers,

respectively, to $DNJ = 86,400$ and $DJ = 300$. From Fig. 4, it is seen that this computation took only 0.8 s. This suggests that with more carefully selected fully symmetric point sets it may be possible to realise the desire expressed in Xi et al. (2018, Section 4) of simultaneous computation of up to thousands of related integrals.

6.4 Symmetric change of measure illustration

The purpose of this final experiment is to briefly illustrate the symmetric change of measure technique, proposed in Sect. 5. To limit scope, we consider applying this technique in conjunction with the fully symmetric standard Bayesian cubature method (i.e. Theorem 1).

6.4.1 Integration problem

Let $\boldsymbol{\mu}_f \in \mathbb{R}^6$ and $\boldsymbol{\Sigma}_f \in \mathbb{R}^{6\times 6}$ be a vector and a positive-definite matrix. Consider integration over $\mathbb{R}^6$ of the function

$$f^\dagger(\boldsymbol{x}) = \exp\left(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_f)^\mathsf{T}\boldsymbol{\Sigma}_f^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_f)\right) \tag{30}$$

with respect to a Gaussian mixture distribution, ν, to be specified. Integrals of this form can be easily computed in closed form. For this illustration, we took

$$\boldsymbol{\mu}_f = \begin{bmatrix} \frac{1}{5} \\ \frac{1}{5} + \frac{3}{50} \\ \vdots \\ \frac{1}{2} - \frac{3}{50} \\ \frac{1}{2} \end{bmatrix}, \quad \boldsymbol{\Sigma}_f = \begin{bmatrix} \left(\frac{4}{5}\right)^2 \boldsymbol{I}_3 & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & \left(\frac{11}{10}\right)^2 \boldsymbol{I}_3 \end{bmatrix}.$$

6.4.2 Setting

For these experiments ν was taken to be a uniform mixture of eight Gaussian distributions $N(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, $i = 1, \dots, 8$, with their mean vectors drawn independently from the standard normal distributions and detrended so that $\sum_{i=1}^8 \boldsymbol{\mu}_i = \boldsymbol{0}$. The covariance matrices of each Gaussian component were independent and normalised draws from the Wishart distribution $W_6(\boldsymbol{I}_6, d+2(q-1))$, $q \in \mathbb{N}$. The resulting ν is almost surely not fully symmetric and therefore Theorem 1 cannot be applied. Different values of q correspond to different degrees of symmetricity of ν: for small values of q covariance matrices $\boldsymbol{\Sigma}_i$ are likely to be nearly singular, while as $q \to \infty$ they become diagonal. Accordingly, we experimented with $q \in \{1, \dots, 8\}$. For each q, the proposal distribution ν_* was a zero-mean Gaussian with diagonal covariance $\sigma^2 \boldsymbol{I}_6$ for σ^2 set to the mean of the diagonal elements of the $\boldsymbol{\Sigma}_i$. For Bayesian cubature, we used the Gaussian kernel (26) with a length-scale $\ell = 0.8$ and the Gauss–Hermite sparse
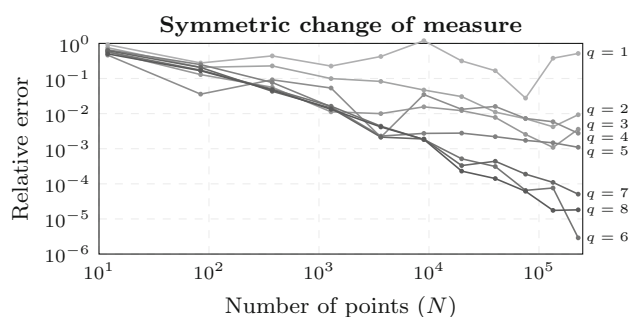
Fig. 5 Relative error in numerical integration of the function (30) using the fully symmetric Bayesian cubature method based on a symmetric change of measure. Here q is used to index symmetricity of ν and thus more challenging ν correspond to small q

grid (Karvonen and Särkkä 2018, Section 4.2) with the midpoint removed. Note that the resulting point sets are not nested for different N.

6.4.3 Results

The results are depicted in Fig. 5 for one fairly representative run. Note how larger values of q correspond to improved integration accuracy. It appears that for reasonably symmetric constituent distributions the proposed method works well; when the covariance matrices are nearly singular we have observed that this simple procedure can seriously fail. This is analogous to scenarios where standard importance sampling can be expected to fare well (Robert and Casella 2013). Thus, based on this example at least, the symmetric change of measure technique appears to be a promising strategy to generalise the results in Theorems 1, 2 and 3. The largest point sets considered contained $J = 168$ fully symmetric sets, which correspond to a point set of size $N = 227,304$.

7 Discussion

There is increasing interest in the use of Bayesian methods for numerical integration (Briol et al. 2019). Bayesian cubature methods are attractive due to analytic and theoretical tractability of the underlying Gaussian model. However, these method are also associated with a computational cost that is cubic in the number of points, N, and moreover the linear systems that must be inverted are typically ill-conditioned.

The symmetry exploits developed in this work circumvent the need for large linear systems to be solved in Bayesian cubature methods. In particular, we presented novel results for Bayes–Sard cubature (Karvonen et al. 2018) and multi-output Bayesian cubature (Xi et al. 2018) that make it possible to apply these methods even for extremely large datasets or when there are many function to be integrated.

In conjunction with the inherent robustness of the Bayes–Sard cubature method (Karvonen et al. 2018), this results in a highly reliable probabilistic integration method that can be applied even to integrals that are relatively high-dimensional.

Three extensions of this work are highlighted: *First*, the combination of multi-output and Bayes–Sard methods appears to be a natural extension and we expect that symmetry properties can similarly be exploited for this method. This could lead to promising procedures for integration of collections of closely related high-dimensional functions appearing in, for example, financial applications (Holtz 2011). Similarly, our exploits should extend to the Student's t based Bayesian cubatures proposed in Prüher et al. (2017). *Second*, the investigation of optimality criteria for the symmetric change of measure technique in Sect. 5 remains to be explored. *Third*, although we focussed solely on computational aspects, the important statistical question of how to ensure Bayesian cubature methods produce output that is well-calibrated remains to some extent unresolved.[5] As discussed in Karvonen and Särkkä (2018), it appears that symmetry exploits do not easily lend themselves to selection of kernel parameters, for instance via cross-validation or maximisation of marginal likelihood.[6] A potential, though somewhat heuristic, way to proceed might be to exploit the concentration of measure phenomenon (Ledoux 2001) or low effective dimensionality of the integrand (Wang and Sloan 2005) in order to identify a suitable data subset on which kernel parameters can be calibrated more easily or a priori.

---

[5] Though, see related work Jagadeeswaran and Hickernell (2019) on this point.

[6] An exception is for kernel amplitude parameters, which can be analytically marginalised as in Proposition 2 of Briol et al. (2019).

References

Álvarez, M., Rosasco, L., Lawrence, N.: Kernels for vector-valued functions: a review. Found. Trends Mach. Learn. **4**(3), 195–266 (2012). https://doi.org/10.1561/2200000036

Bach, F., Lacoste-Julien, S., Obozinski, G.: On the equivalence between herding and conditional gradient algorithms. In: Proceedings of the 29th International Conference on Machine Learning, pp. 1355–1362 (2012). https://icml.cc/2012/papers/683.pdf. Accessed Sept 3 2019

Bach, F.: On the equivalence between kernel quadrature rules and random feature expansions. J. Mach. Learn. Res. **18**(21), 1–38 (2017)

Berlinet, A., Thomas-Agnan, C.: Reproducing Kernel Hilbert Spaces in Probability and Statistics. Springer, New York (2011)

Bezhaev, A.Yu..: Cubature formulae on scattered meshes. Sov. J. Numer. Anal. Math. Model. **6**(2), 95–106 (1991). https://doi.org/10.1515/rnam.1991.6.2.95

Brauchart, J.S., Saff, E.B., Sloan, I.H., Womersley, R.S.: QMC designs: optimal order quasi Monte Carlo integration schemes on the sphere. Math. Comput. **83**(290), 2821–2851 (2014). https://doi.org/10.1090/S0025-5718-2014-02839-1

Briol, F.-X., Oates, C.J., Cockayne, J., Chen, W.Y., Girolami, M.: On the sampling problem for kernel quadrature. In: Proceedings of the 34th International Conference on Machine Learning, pp. 586–595 (2017). http://proceedings.mlr.press/v70/briol17a.html. Accessed Sept 3 2019

Briol, F.-X., Oates, C.J., Girolami, M., Osborne, M.A., Sejdinovic, D.: Probabilistic integration: a role in statistical computation? Stat. Sci. **34**(1), 1–22 (2019)

Briol, F.-X., Oates, C.J., Girolami, M., Osborne, M.A.: Frank-Wolfe Bayesian quadrature: probabilistic integration with theoretical guarantees. In: Advances in Neural Information Processing Systems, vol. 28, pp. 1162–1170 (2015). https://papers.nips.cc/paper/5749-frank-wolfe-bayesian-quadrature-probabilistic-integration-with-theoretical-guarantees. Accessed Sept 3 2019

Brouillat, J., Bouville, C., Loos, B., Hansen, C., Bouatouch, K.: A Bayesian Monte Carlo approach to global illumination. Comput. Graph. Forum **28**(8), 2315–2329 (2009). https://doi.org/10.1111/j.1467-8659.2009.01537.x

Chai, H., Garnett, R.: An improved Bayesian framework for quadrature of constrained integrands (2018). arXiv:1802.04782

Chen, W., Mackey, L., Gorham, J., Briol, F.-X., Oates, C.J.: Stein points. In: Proceedings of the 35th International Conference on Machine Learning (2018). http://proceedings.mlr.press/v80/chen18f. Accessed Sept 3 2019

Cockayne, J., Oates, C.J., Sullivan, T., Girolami, M.: Bayesian probabilistic numerical methods (2017). arXiv:1702.03673

Cools, R.: Constructing cubature formulae: the science behind the art. Acta Numer. **6**, 1–54 (1997). https://doi.org/10.1017/S0962492900002701

Davis, P.J., Rabinowitz, P.: Methods of Numerical Integration. Courier Corporation, North Chelmsford (2007)

DeVore, R., Foucart, S., Petrova, G., Wojtaszczyk, P.: Computing a quantity of interest from observational data. Constr. Approx. (2018). https://doi.org/10.1007/s00365-018-9433-7

Diaconis, P.: Bayesian numerical analysis. In: Gupta, S.S., Berger, J.O. (eds.) Statistical Decision Theory and Related Topics IV, vol. 1, pp. 163–175. Springer, New York (1988). https://doi.org/10.1007/978-1-4613-8768-8_20

Dietrich, C.R., Newsam, G.N.: Fast and exact simulation of stationary Gaussian processes through circulant embedding of the covariance matrix. SIAM J. Sci. Comput. **18**(4), 1088–1107 (1997). https://doi.org/10.1137/s1064827592240555

Dutre, P., Bekaert, P., Bala, K.: Advanced Global Illumination. AK Peters/CRC Press, Boca Raton (2006). https://doi.org/10.1201/9781315365473

Ehler, M., Graef, M., Oates, C.J.: Optimal Monte Carlo integration on closed manifolds. Stat. Comput. (2019). https://doi.org/10.1007/s11222-019-09894-w

Genz, A.: Fully symmetric interpolatory rules for multiple integrals. SIAM J. Numer. Anal. **23**(6), 1273–1283 (1986). https://doi.org/10.1137/0723086

Genz, A., Keister, B.D.: Fully symmetric interpolatory rules for multiple integrals over infinite regions with Gaussian weight. J. Comput. Appl. Math. **71**(2), 299–309 (1996). https://doi.org/10.1016/0377-0427(95)00232-4

Gunter, T., Osborne, M.A., Garnett, R., Hennig, P., Roberts, S.J.: Sampling for inference in probabilistic models with fast Bayesian quadrature. In: Advances in Neural Information Processing Systems, vol. 27, pp. 2789–2797 (2014). https://papers.nips.cc/paper/5483-sampling-for-inference-in-probabilistic-models-with-fast-bayesian-quadrature. Accessed Sept 3 2019

Hackbusch, W.: A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: introduction to $\mathcal{H}$-matrices. Computing **62**(2), 89–108 (1999). https://doi.org/10.1007/s006070050015

Hennig, P., Osborne, M.A., Girolami, M.: Probabilistic numerics and uncertainty in computations. Proc. R. Soc. Lond. A Math. Phys. Eng. Sci. (2015). https://doi.org/10.1098/rspa.2015.0142

Hensman, J., Durrande, N., Solin, A.: Variational Fourier features for Gaussian processes. J. Mach. Learn. Res. **11**(151), 1–52 (2018)

Holtz, M.: Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance. Number 77 in Lecture Notes in Computational Science and Engineering. Springer, New York (2011). https://doi.org/10.1007/978-3-642-16004-2

Jagadeeswaran, R., Hickernell, F.J.: Fast automatic Bayesian cubature using lattice sampling. Stat. Comput. (2019). https://doi.org/10.1007/s11222-019-09895-9. (to appear)

Kanagawa, M., Sriperumbudur, B. K., Fukumizu, K.: Convergence guarantees for kernel-based quadrature rules in misspecified settings. In: Advances in Neural Information Processing Systems, vol. 29, pp. 3288–3296 (2016). arXiv:1605.07254

Kanagawa, M., Sriperumbudur, B.K., Fukumizu, K.: Convergence analysis of deterministic kernel-based quadrature rules in misspecified settings. Found. Comput. Math. (2019). https://doi.org/10.1007/s10208-018-09407-7

Karvonen, T., Särkkä, S., Oates, C.J.: A Bayes–Sard cubature method. In: Advances in Neural Information Processing Systems, vol. 31, pp. 5882–5893 (2018). https://papers.nips.cc/paper/7829-a-bayes-sard-cubature-method. Accessed Sept 3 2019

Karvonen, T., Särkkä, S.: Classical quadrature rules via Gaussian processes. In: 27th IEEE International Workshop on Machine Learning for Signal Processing (2017). https://doi.org/10.1109/mlsp.2017.8168195

Karvonen, T., Särkkä, S.: Fully symmetric kernel quadrature. SIAM J. Sci. Comput. **40**(2), A697–A720 (2018). https://doi.org/10.1137/17m1121779

Kennedy, M.: Bayesian quadrature with non-normal approximating functions. Stat. Comput. **8**(4), 365–375 (1998). https://doi.org/10.1023/A:1008832824006

Larkin, F.M.: Probabilistic error estimates in spline interpolation and quadrature. In: Information Processing 74 (Proceedings of IFIP Congress, Stockholm, 1974), vol. 74, pp. 605–609. North-Holland (1974)

Larkin, F.M.: Gaussian measure in Hilbert space and applications in numerical analysis. Rocky Mt. J. Math. **2**(3), 379–421 (1972). https://doi.org/10.1216/rmj-1972-2-3-379

Lázaro-Gredilla, M., Quiñonero-Candela, J., Rasmussen, C.E., Figueiras-Vidal, A.R.: Sparse spectrum Gaussian process regression. J. Mach. Learn. Res. **11**, 1865–1881 (2010)

Ledoux, M.: The Concentration of Measure Phenomenon. Number 89 in Mathematical Surveys and Monographs. American Mathematical Society, Providence (2001). https://doi.org/10.1090/surv/089

Lu, J., Darmofal, D.L.: Higher-dimensional integration with Gaussian weight for applications in probabilistic design. SIAM J. Sci. Comput. **26**(2), 613–624 (2004). https://doi.org/10.1137/s1064827503426863

Marques, R., Bouville, C., Ribardière, M., Santos, L.P., Bouatouch, K.: A spherical Gaussian framework for Bayesian Monte Carlo rendering of glossy surfaces. IEEE Trans. Vis. Comput. Graph. **19**(10), 1619–1932 (2013). https://doi.org/10.1109/tvcg.2013.79

Marques, R., Bouville, C., Santos, L.P., Bouatouch, K.: Efficient Quadrature Rules for Illumination Integrals: From Quasi Monte Carlo to Bayesian Monte Carlo. Synthesis Lectures on Computer Graphics and Animation. Morgan & Claypool Publishers, San Rafael (2015). https://doi.org/10.2200/s00649ed1v01y201505cgr019

McNamee, J., Stenger, F.: Construction of fully symmetric numerical integration formulas. Numer. Math. **10**(4), 327–344 (1967). https://doi.org/10.1007/BF02162032

Minka, T.: Deriving quadrature rules from Gaussian processes. Technical report, Microsoft Research, Statistics Department, Carnegie Mellon University (2000). https://www.microsoft.com/en-us/research/publication/deriving-quadrature-rules-gaussian-processes/. Accessed Sept 3 2019

Najm, H.N., Debusschere, B.J., Marzouk, Y.M., Widmer, S., Le Maître, O.: Uncertainty quantification in chemical systems. Int. J. Numer. Methods Eng. **80**(6–7), 789–814 (2009). https://doi.org/10.1002/nme.2551

Novak, E., Ritter, K.: Simple cubature formulas with high polynomial exactness. Constr. Approx. **15**(4), 499–522 (1999). https://doi.org/10.1007/s003659900119

Novak, E., Ritter, K., Schmitt, R., Steinbauer, A.: On an interpolatory method for high dimensional integration. J. Comput. Appl. Math. **112**(1–2), 215–228 (1999). https://doi.org/10.1016/s0377-0427(99)00222-8

Oates, C.J., Niederer, S., Lee, A., Briol, F.-X., Girolami, M.: Probabilistic models for integration error in the assessment of functional cardiac models. In: Advances in Neural Information Processing Systems, vol. 30, pp. 109–117 (2017). http://papers.nips.cc/paper/6616-probabilistic-models-for-integration-error-in-the-assessment-of-functional-cardiac-models. Accessed Sept 3 2019

Oettershagen, J.: Construction of Optimal Cubature Algorithms with Applications to Econometrics and Uncertainty Quantification. Ph.D. thesis, Institut für Numerische Simulation, Universität Bonn (2017)

O'Hagan, A.: Curve fitting and optimal design for prediction. J. R. Stat. Soc. Ser. B (Methodol.) **40**(1), 1–42 (1978). https://doi.org/10.1111/j.2517-6161.1978.tb01643.x

O'Hagan, A.: Bayes–Hermite quadrature. J. Stat. Plan. Inference **29**(3), 245–260 (1991). https://doi.org/10.1016/0378-3758(91)90002-v

Osborne, M., Garnett, R., Ghahramani, Z., Duvenaud, D.K., Roberts, S.J., Rasmussen, C.E.: Active learning of model evidence using Bayesian quadrature. In: Advances in Neural Information Processing Systems, vol. 25, pp. 46–54 (2012a). https://papers.nips.cc/paper/4657-active-learning-of-model-evidence-using-bayesian-quadrature. Accessed Sept 3 2019

Osborne, M., Garnett, R., Roberts, S., Hart, C., Aigrain, S., Gibson, N.: Bayesian quadrature for ratios. In: Artificial Intelligence and Statistics, pp. 832–840 (2012b). http://proceedings.mlr.press/v22/osborne12/osborne12.pdf. Accessed Sept 3 2019

Pronzato, L., Zhigljavsky, A.: Bayesian quadrature and energy minimization for space-filling design (2018). arXiv:1808.10722

Prüher, J., Tronarp, F., Karvonen, T., Särkkä, S., Straka, O.: Student-t process quadratures for filtering of non-linear systems with heavy-tailed noise. In: 20th International Conference on Information Fusion (2017). https://doi.org/10.23919/icif.2017.8009742

Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)

Robert, C., Casella, G.: Monte Carlo Statistical Methods. Springer, New York (2013)

Särkkä, S., Hartikainen, J., Svensson, L., Sandblom, F.: On the relation between Gaussian process quadratures and sigma-point methods. J. Adv. Inf. Fusion **11**(1), 31–46 (2016). arXiv:1504.05994

Schaback, R.: Error estimates and condition numbers for radial basis function interpolation. Adv. Comput. Math. **3**(3), 251–264 (1995). https://doi.org/10.1007/bf02432002

Schäfer, F., Sullivan, T.J., Owhadi, H.: Compression, inversion, and approximate PCA of dense kernel matrices at near-linear computational complexity (2017). arXiv:1706.02205

Smola, A., Gretton, A., Song, L., Schölkopf, B.: A Hilbert space embedding for distributions. In: International Conference on Algorithmic Learning Theory, pp. 13–31. Springer (2007). https://doi.org/10.1007/978-3-540-75225-7_5

Sommariva, A., Vianello, M.: Numerical cubature on scattered data by radial basis functions. Computing **76**(3–4), 295–310 (2006). https://doi.org/10.1007/s00607-005-0142-2

Stein, M.L.: Interpolation of Spatial Data: Some Theory for Kriging. Springer, New York (2012)

Wang, X., Sloan, I.H.: Why are high-dimensional finance problems often of low effective dimension? SIAM J. Sci. Comput. **27**(1), 159–183 (2005). https://doi.org/10.1137/s1064827503429429

Wendland, H.: Scattered Data Approximation. Number 28 in Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge (2005)

Xi, X., Briol, F.-X., Girolami, M.: Bayesian quadrature for multiple related integrals. In: Proceedings of the 35th International Conference on Machine Learning (2018) **(to appear)**. arXiv:1801.04153

Xiu, D., Karniadakis, G.E.: Modeling uncertainty in flow simulations via generalized polynomial chaos. J. Comput. Phys. **187**(1), 137–167 (2003). https://doi.org/10.1016/s0021-9991(03)00092-5