



Approximation and sampling of multivariate probability distributions in the tensor train decomposition

Sergey Dolgov¹ · Karim Anaya-Izquierdo¹ · Colin Fox² · Robert Scheichl³

Received: 2 July 2019 / Accepted: 17 October 2019 / Published online: 2 November 2019
© The Author(s) 2019

Abstract

General multivariate distributions are notoriously expensive to sample from, particularly the high-dimensional posterior distributions in PDE-constrained inverse problems. This paper develops a sampler for arbitrary continuous multivariate distributions that is based on low-rank surrogates in the tensor train format, a methodology that has been exploited for many years for scalable, high-dimensional density function approximation in quantum physics and chemistry. We build upon recent developments of the cross approximation algorithms in linear algebra to construct a tensor train approximation to the target probability density function using a small number of function evaluations. For sufficiently smooth distributions, the storage required for accurate tensor train approximations is moderate, scaling linearly with dimension. In turn, the structure of the tensor train surrogate allows sampling by an efficient conditional distribution method since marginal distributions are computable with linear complexity in dimension. Expected values of non-smooth quantities of interest, with respect to the surrogate distribution, can be estimated using transformed independent uniformly-random seeds that provide Monte Carlo quadrature or transformed points from a quasi-Monte Carlo lattice to give more efficient quasi-Monte Carlo quadrature. Unbiased estimates may be calculated by correcting the transformed random seeds using a Metropolis–Hastings accept/reject step, while the quasi-Monte Carlo quadrature may be corrected either by a control-variate strategy or by importance weighting. We show that the error in the tensor train approximation propagates linearly into the Metropolis–Hastings rejection rate and the integrated autocorrelation time of the resulting Markov chain; thus, the integrated autocorrelation time may be made arbitrarily close to 1, implying that, asymptotic in sample size, the cost per effectively independent sample is one target density evaluation plus the cheap tensor train surrogate proposal that has linear cost with dimension. These methods are demonstrated in three computed examples: fitting failure time of shock absorbers; a PDE-constrained inverse diffusion problem; and sampling from the Rosenbrock distribution. The delayed rejection adaptive Metropolis (DRAM) algorithm is used as a benchmark. In all computed examples, the importance weight-corrected quasi-Monte Carlo quadrature performs best and is more efficient than DRAM by orders of magnitude across a wide range of approximation accuracies and sample sizes. Indeed, all the methods developed here significantly outperform DRAM in all computed examples.

Keywords Multivariate distributions · Surrogate models · Tensor decomposition · MCMC · Importance weights

✉ Sergey Dolgov
s.dolgov@bath.ac.uk

Karim Anaya-Izquierdo
K.Anaya-Izquierdo@bath.ac.uk

Colin Fox
colin.fox@otago.ac.nz

Robert Scheichl
r.scheichl@uni-heidelberg.de

¹ University of Bath, Bath, UK

² University of Otago, Dunedin, New Zealand

³ University of Heidelberg, Heidelberg, Germany

1 Introduction

We present an algorithm for efficient MCMC when the target distribution is a continuous multivariate distribution with known, tractable probability density function (PDF) $\pi(x)$ defined for x in a region in \mathbb{R}^d . Beyond a fixed function approximation phase that has cost that scales linearly with dimension d , independent draws from π cost (a fraction more than) one function evaluation per independent sample. We give a basic form of the algorithm that generates random samples from π and variants that allow efficient quadrature using

quasi-Monte Carlo constructions and/or importance weighting.

There are currently few general-purpose options for sampling from multivariate distributions with no special form, particularly if one is seeking a black box sampler that does not require case-specific tuning. We commonly encounter such distributions as the posterior distribution in a Bayesian analysis of a nonlinear inverse problem (Fox and Nicholls 1997; Fox et al. 2013; Dodwell et al. 2015) (also see the example in Sect. 5.3), or as the marginal posterior distribution over hyperparameters in a linear Gaussian inverse problem (Fox and Norton 2016), see also Norton et al. (2018). This work is motivated by the desire to compute inference in those examples, though the samplers and quadrature methods we present here are applicable to arbitrary continuous distributions, which could be non-Gaussian, or multimodal, and unnormalized; for example, see the example in Sect. 5.1. In target applications, the aim of sampling is often to implement Monte Carlo integration to compute summary statistics of the posterior distribution over an unobserved quantity of interest (QoI). For applications in inverse problems, the state variable is typically high-dimensional, thus requiring computation of high-dimensional quadratures (Stuart 2010), even when the QoI is low-dimensional.

Efficient black box samplers exist for some special classes of distributions. Most notable amongst multivariate distributions are multivariate normal (MVN) distributions, with fixed covariance or precision matrix, for which efficient, automatic sampling is available using stochastic variants of efficient algorithms for solving systems of equations in the covariance or precision matrix; methods based on *direct* solvers, using Cholesky factoring, can be found in Rue (2001) and Rue and Held (2005), while more recently samplers based on accelerated *iterative* solvers have been developed; see Fox and Parker (2017) and references therein. For non-Gaussian distributions, virtually all samplers are variants of Metropolis–Hastings (MH) MCMC with a random walk proposal, of which there are many variants (Brooks et al. 2011). These algorithms are geometrically convergent, at best, so they can be very slow for our target applications. Two black box versions are the delayed acceptance adaptive Metropolis (DRAM) (Haario et al. 2006) and the t-walk (Christen and Fox 2010). Both of these algorithms require multiple evaluations of the target PDF per effectively independent output sample, with that number growing roughly linearly or worse with dimension, even for simple distributions such as MVN [see Norton et al. (2018) for the cost of these algorithms].

Computational sampling from *univariate* distributions is effectively a solved problem due to developments of the adaptive rejection sampler (ARS) (Gilks and Wild 1992), such as independent doubly adaptive rejection Metropolis sampling (IA²RMS) (Martino et al. 2015). These algorithms approximate the univariate PDF using simple functions, with

the approximation improving (adaptively) as the algorithm progresses to achieve efficient sampling. The ARS, which is restricted to log-convex PDFs, builds a piecewise linear upper bound to the log of the PDF and hence bounds the PDF, to give an efficient proposal in a rejection sampler.¹ The IA²RMS has no restriction on the PDF and uses a sequence of simple function approximations to the PDF or log PDF, such as piecewise constant or piecewise linear approximations that converge in distribution to the PDF as the algorithm progresses. Sampling from these approximations is easy in this univariate case, whether approximating the PDF or log PDF, using the inverse cumulative transformation method (Devroye 1986; Johnson 1987; Hörmann et al. 2004), with samples providing independence proposals to a Metropolis–Hastings accept/reject step that ensures the correct equilibrium distribution. Distributional convergence of the approximation implies that, asymptotic in sample size, just one PDF evaluation is required per independent sample.²

The sampler developed here is inspired, to some extent, by IA²RMS, in that it uses function approximation methods to approximate the multivariate PDF in a way that then allows cheap simulation from the approximation. Specifically, we use an interpolation in tensor train (TT) representation that may be made arbitrarily accurate, with sampling via the *conditional distribution method* that is the multivariate extension of inverse cumulative transformation sampling for univariate distributions (Johnson 1987). The conditional distribution method requires computing integrals of the multivariate PDF $\pi(x_1, \dots, x_d)$, over subsets of variables x_k, \dots, x_d for $k = 2, \dots, d$, in order to obtain univariate marginal-conditional distributions. *Per se*, this problem is as difficult as the original quadrature. By using the TT decomposition (Oseledets 2011b), this integration can be performed efficiently, and each univariate marginal-conditional distribution can then be easily sampled using its inverse cumulative distribution function (CDF). Since the inverse cumulative transform is isoprobabilistic, the resulting samples are *exact* for the interpolated probability tensor, which is, however, an *approximation* to the original target PDF. We provide bounds on the sampling error based on the approximation errors of the TT decomposition and discretization, and thus are able to trade accuracy for compute time.

An accurate approximation to the PDF allows the almost exact samples to be used directly, while a less expensive approximation may be used to produce independence proposals for a MH accept/reject step that ‘corrects’ the distribution.

¹ Meyer et al. (2008) used piecewise quadratic approximations to the log PDF giving piecewise Gaussian approximated PDF.

² The MATLAB package for IA²RMS available at <http://a2rms.sourceforge.net/> is far more expensive than this minimal theoretical cost, besides not being robust.

The conditional distribution sampler may also be seeded with quasi-Monte Carlo points in the unit cube to implement quasi-Monte Carlo quadrature that is corrected by a multilevel MCMC scheme or by importance weighting. These variants are discussed in Sect. 4. We find that the combination of quasi-Monte Carlo seed points combined with importance-weighted quadrature gives the best performance in computed examples.

The attraction of approximating the PDF in TT format is that the computational cost of the construction, the storage requirements and the operations required for conditional distribution method sampling from the distributional approximation all scale *linearly* with dimension; see Sect. 3. In contrast, direct calculation or naïve representations lead to *exponential* cost for each of these tasks. This is a remarkable feature of the TT representation, and this is why the recent introduction of low-rank hierarchical tensor methods, such as TT (Oseledets and Tyrtyshnikov 2010; Oseledets 2011a, b, 2013), is a significant development in scientific computing for multidimensional problems.

Thus, the basic sampler we present here differs from IA^2RMS in two important aspects (beyond being able to handle multivariate distributions): we approximate the PDF and not the log PDF, and the sampler is not adaptive. The PDF is approximated because operations available on the TT representation that have cost that scales linearly with dimension include those required for performing the conditional distribution sampling (see Sect. 3), while it is not clear how to perform sampling when the log PDF is approximated in the multivariate case. Further, current methods for TT representation do not include convenient and cheap schemes for updating a TT representation using a single new evaluation. Hence, the algorithm we present consists of two steps: in a set-up phase, the TT approximation to $\pi(\cdot)$ is constructed; then, that fixed approximation is used to generate samples. Hence, unlike the univariate samplers mentioned above, the TT approximation and samplers presented here are restricted to distributions with bounded, known support. While it is simple to define coordinate transformations $\mathbb{R} \mapsto [0, 1]$ to represent a distribution on the (bounded) unit cube, efficient sampling still requires locating the appreciable support of the distribution; indeed, that is often a significant task when performing sampling. We do not consider such transformations here. Despite this restriction, the method advances sample-based inference in some problems of substantial interest, as shown in the computed examples in Sect. 5.

Approximation of the multivariate target distribution can be recommended for the following two cases: first, the quantity of interest may be very poorly representable in the TT format, and hence, direct tensor product integration of the QoI, as suggested in Eigel et al. (2018), is not possible. The most remarkable example is the indicator function, which occurs in the computation of the probability of an event. If

the jump of the indicator function is not aligned to the coordinate axes, the cost of its TT approximation might grow exponentially in the number of variables. Then, Monte Carlo quadrature becomes the only possibility, with the quadrature error depending on the particular distribution of the samples. When the target density function admits a TT approximation with a modest storage, the cumulative transform method can produce optimally distributed samples at a low cost. Secondly, even when a fast growth of the TT storage prevents accurate computation of the density function, the TT surrogate distributed samples can still be used as proposals in the MH algorithm or with importance weighting. Even a crude approximation to the PDF with 10% error can produce the acceptance rate of 90% and the integrated autocorrelation time of 1.2, which is close enough to the best possible practical MCMC. The relationship between approximation error and acceptance rate is formalized in Sect. 4.2.

The paper is structured as follows: in Sect. 2, we review the conditional sampling method used to sample from the multivariate TT-interpolated approximation. Some background on the TT decomposition is presented in Sect. 3. A metropolized algorithm that uses the TT surrogate for sampling from the target distribution is presented in Sect. 4, as well as methods for unbiased quadrature that utilize a two-level algorithm, importance weighting and quasi-Monte Carlo seed points. Several numerical examples are presented in Sect. 5: Sect. 5.1 shows posterior estimation of a shock absorber failure probability; Sect. 5.2 demonstrates efficient sampling when the Rosenbrock function is the log target density, which is a synthetic ‘banana-shaped’ PDF that presents difficulties to random walk MCMC samplers; and Sect. 5.3 demonstrates posterior inference in a classical inverse problem in subsurface flow. In each of the numerical examples, scaling for the TT-based sampling and quadrature is shown, in comparison with DRAM (Haario et al. 2006), as well as (in Sect. 5.3) with direct quasi-Monte Carlo quadrature.

2 Conditional distribution sampling method

The *conditional distribution method* (Devroye 1986; Johnson 1987; Hörmann et al. 2004) reduces the task of generating a d -dimensional random vector into a sequence of d univariate generation tasks.

Let (X_1, \dots, X_d) be a continuous random vector with a probability density function $\pi(x_1, \dots, x_d)$. To simplify the presentation, we assume in this section that π is normalized. The density function can be written as a product of conditional densities,

$$\pi(x_1, \dots, x_d) = \pi_1(x_1)\pi_2(x_2|x_1) \cdots \pi_d(x_d|x_1, \dots, x_{d-1}),$$

where $\pi_k(x_k|x_1, \dots, x_{k-1})$ is a conditional density given by

$$\pi_k(x_k|x_1, \dots, x_{k-1}) = \frac{p_k(x_1, \dots, x_k)}{p_{k-1}(x_1, \dots, x_{k-1})}, \quad (1)$$

in terms of the marginal densities,

$$p_k = \int \pi(x_1, \dots, x_{k-1}, x_k, x_{k+1}, \dots, x_d) dx_{k+1} \cdots dx_d, \quad (2)$$

where $k = 1, \dots, d$. To simplify the notation we set $p_0 = 1$. The conditional distribution method then generates $(x_1, \dots, x_d) \sim \pi$ by sampling from each of the univariate conditional densities in turn:

for $k = 1, 2, \dots, d$ **do**

 Generate $x_k \sim \pi_k(x_k|x_1, \dots, x_{k-1})$.

end for

This follows by straightforward manipulation of the definitions of marginal and conditional distributions.

To generate the univariate samples in the algorithm above, we use the inverse cumulative transformation method. Thus, our algorithm coincides with the inverse Rosenblatt transformation (Rosenblatt 1952) from the d -dimensional unit cube to the state space of π . The standard conditional distribution method uses independent samples distributed uniformly in the unit cube as seeds for the transformation to produce independent draws from π . This generalizes the inverse cumulative transformation method for univariate distributions. Later, we will also use quasi-random points to implement quasi-Monte Carlo quadrature for evaluating expectations with respect to π .

When the analytic inverse of each univariate cumulative distribution function is not available, a straightforward numerical procedure is to discretize the univariate density on a grid, with approximate sampling carried out using a polynomial interpolation. In that case, the normalization, i.e. the denominator in (1), is not necessary as normalization of the numerical approximation is evaluated, allowing sampling from an unnormalized marginal density (2), directly.

The main difficulty with the conditional distribution method for multivariate random generation is obtaining all necessary marginal densities, which requires the high-dimensional integral over $x_{k+1} \dots x_d$ in (2). In general, this calculation can be extremely costly. Even a simple discretization of the argument of the marginal densities (2), or the conditional–marginal densities (1), leads to exponential cost with dimension.

To overcome this cost, we *precompute* an approximation of $\pi(x_1, \dots, x_d)$ in a compressed representation that allows fast computation of integrals in (2) and subsequent sampling from the conditionals in (1). In the next sections, we introduce

the TT decomposition and the related TT-cross algorithm (Oseledets and Tyrtshnikov 2010) for building a TT approximation to π . Moreover, we show that the separated form of the TT representation allows an efficient integration in (2), with cost that scales *linearly* with dimension.

3 TT approximation of the target distribution

Tensor decompositions trace back to the low-rank skeleton decompositions of matrices, which can in turn be computed by the singular value decomposition (SVD). Any matrix $P \in \mathbb{R}^{n \times m}$ (e.g. a bivariate discrete distribution) admits a SVD $P = U \Sigma V^T$, where U, V are orthonormal matrices of singular vectors and Σ is a diagonal matrix of non-negative singular values. If the matrix is of *low rank*, $r := \text{rank } P < \min(m, n)$, the bottom right corner of Σ is zero, so we can *truncate* the SVD to $U_r \Sigma_r V_r^T$, where U_r, V_r contain only the first r columns and Σ_r contains only the principal $r \times r$ submatrix. However, we can also *approximate* the given matrix P by a truncated decomposition of lower rank; the Eckart–Young theorem (Golub and Van Loan 2013) ensures the optimality of the rank- r SVD approximation among all possible rank- r approximations. Naturally, U_r and V_r contain only $(n+m)r$ elements in contrast to nm elements in P . This process can be extended to build low-rank decompositions of multivariate distributions, which we will describe next.

3.1 Interpolated TT decomposition

Throughout the paper, we approximate the target PDF by an interpolated TT decomposition (Oseledets 2011b),

$$\begin{aligned} \pi(x_1, \dots, x_d) &\approx \tilde{\pi}(x_1, \dots, x_d) \\ &= \sum_{\alpha_0, \dots, \alpha_d=1}^{r_0, \dots, r_d} \pi_{\alpha_0, \alpha_1}^{(1)}(x_1) \pi_{\alpha_1, \alpha_2}^{(2)}(x_2) \cdots \pi_{\alpha_{d-1}, \alpha_d}^{(d)}(x_d), \end{aligned} \quad (3)$$

that is a sum of products of the univariate functions $\pi_{\alpha_{k-1}, \alpha_k}^{(k)}(x_k)$, $k = 1, 2, \dots, d$ indexed by $\alpha_k = 1, \dots, r_k$. The r_k , $k = 0, \dots, d$, are called *TT ranks*, with $r_0 = r_d = 1$ (because π is scalar-valued), but r_1, \dots, r_{d-1} can be larger. The efficiency of this representation relies on the TT ranks being bounded by some (smallish) number r , as discussed later.

The TT decomposition natively represents a tensor, or d -dimensional array of values. The function approximation (3) is obtained by first approximating the tensor that results from discretizing the PDF $\pi(x_1, \dots, x_d)$ by collocation on a tensor product of univariate grids. Let $x_k^{i_k} \in \mathbb{R}$, with $i_k = 1, \dots, n_k$ and $x_k^1 < \dots < x_k^{n_k}$, define independent univariate grids in each variable, and let $\hat{\pi}(i_1, i_2, \dots, i_d) = \pi(x_1^{i_1}, x_2^{i_2}, \dots, x_d^{i_d})$. The TT representation is

$$\hat{\pi}(i_1, i_2, \dots, i_d) = \sum_{\alpha_0, \dots, \alpha_d=1}^{r_0, \dots, r_d} \hat{\pi}_{\alpha_0, \alpha_1}^{(1)}(i_1) \hat{\pi}_{\alpha_1, \alpha_2}^{(2)}(i_2) \cdots \hat{\pi}_{\alpha_{d-1}, \alpha_d}^{(d)}(i_d) \tag{4}$$

with *TT blocks* $\hat{\pi}^{(k)}$. Each TT block is a collection of $r_{k-1}r_k$ vectors of length n_k , i.e. $\hat{\pi}^{(k)}(i_k) = \pi^{(k)}(x_k^{i_k})$ is a three-dimensional tensor of size $r_{k-1} \times n_k \times r_k$. If we assume that all $n_k \leq n$ and $r_k \leq r$ for some uniform bounds $n, r \in \mathbb{N}$, the storage cost of (4) can be estimated by dnr^2 which is linear in the number of variables. In contrast, the number of elements in the *tensor* of nodal values $\hat{\pi}(i_1, \dots, i_d)$ grows exponentially in d and quickly becomes prohibitively large with increasing d .

The continuous approximation of π (3) is given by a piecewise polynomial interpolation of nodal values or TT blocks. For example, in the linear case we have

$$\pi^{(k)} = \frac{x_k - x_k^{i_k}}{x_k^{i_k+1} - x_k^{i_k}} \cdot \hat{\pi}^{(k)}(i_k + 1) + \frac{x_k^{i_k+1} - x_k}{x_k^{i_k+1} - x_k^{i_k}} \cdot \hat{\pi}^{(k)}(i_k),$$

for $x_k^{i_k} \leq x_k \leq x_k^{i_k+1}$, which induces the corresponding multilinear approximation $\tilde{\pi}^{(k)}$ of π in (3).

If the individual terms $\pi_{\alpha_{k-1}, \alpha_k}^{(k)}(x_k)$ are normalized PDFs, the TT approximation in (3) may be viewed as a mixture distribution. However, the TT decomposition can be more general and may also include negative terms. Moreover, at some locations where $\pi(x)$ is close to zero the whole approximation $\tilde{\pi}(x)$ may take (small) negative values. This will be circumvented by explicitly taking absolute values in the conditional distribution sampling method; see Sec. 4.1.

The interpolated TT approximation to π in (3) required several choices. First a coordinate system must be chosen, then an ordering of coordinates, then a rectangular region that contains the (appreciable) support of the PDF and then univariate grids for each coordinate within the rectangular region. Each of these choices affects the TT ranks and hence the efficiency of the TT representation in terms of storage size versus accuracy of the approximation that is also chosen; see later. In this sense, the sampler that we develop is not ‘black box’. However, as we demonstrate in the computed examples, an unsophisticated choice for each of these steps already leads to a computational method for sampling and evaluating expectations that is substantially more efficient than existing MCMC algorithms. Smart choices for each of these steps could lead to further improvements.

The rationale behind the independent discretization of all variables is the rapid convergence of tensor product Gaussian quadrature rules. If $\pi(x)$ is analytic with respect to all variables, the error of the Gaussian quadrature converges exponentially in n . A straightforward summation of n^d quadrature terms would imply a cost of $\mathcal{O}(|\log \varepsilon|^d)$ for

accuracy ε . In contrast, the TT ranks often depend logarithmically on ε under the same assumptions on $\pi(x)$ (Tyrtshnikov 2003; Khoromskij 2006; Schneider and Uschmajew 2013), leading to $\mathcal{O}(d|\log \varepsilon|^3)$ cost of the TT integration, since the integration of the TT decomposition factorizes into one-dimensional integrals over the TT blocks. This can also be significantly cheaper than the $\mathcal{O}(\varepsilon^{-2})$ cost of Monte Carlo quadrature.

In general, it is difficult to deduce sharp bounds for the TT ranks. Empirically, low ranks occur in the situation of ‘weakly’ dependent variables. For example, if x_1, \dots, x_d correspond to independent random quantities, the PDF factorizes into a single product of univariate densities, which corresponds to the simplest case, $r = 1$ in (3). Thus, a numerical algorithm that can robustly reveal the ranks is indispensable.

3.2 TT-cross approximation

A quasi-optimal approximation of $\hat{\pi}$ for a given TT rank, in the Frobenius norm, is available via the truncated singular value decomposition (SVD) (Oseledets 2011b). However, the SVD requires storage of the full tensor which is not affordable in many dimensions. A practical method needs to be able to compute the representation (3) using *only a few* evaluations of π . A workhorse algorithm of this kind is the alternating TT-cross method (Oseledets and Tyrtshnikov 2010). That builds on the skeleton decomposition of a matrix (Goreinov et al. 1997). It represents an $n \times m$ matrix P of rank r as the *cross* (in MATLAB-like notation)

$$P = P(:, \mathcal{J})P(\mathcal{I}, \mathcal{J})^{-1}P(\mathcal{I}, :) \tag{5}$$

of r columns and rows, where \mathcal{I} and \mathcal{J} are two index sets of cardinality r such that $P(\mathcal{I}, \mathcal{J})$ (the intersection matrix) is non-singular. If $r \ll n, m$, this decomposition requires computing only $(n + m - r)r \ll nm$ elements of the original matrix. The SVD may be used for choosing the cross (5), though with greater cost, as noted above.

The TT-cross approximation may now be constructed by reducing the sequence of *unfolding* matrices $\hat{\pi}_k = [\hat{\pi}(i_1, \dots, i_k; i_{k+1}, \dots, i_d)]$ that have the first k indices grouped together to index rows and the remaining indices grouped to index columns. We begin with $\hat{\pi}_1$.

We start with a set $\mathcal{I}_{>1} = \{(i_2^{\alpha_1}, \dots, i_d^{\alpha_1})\}_{\alpha_1=1}^{r_1}$ of r_1 $(d - 1)$ -tuples such that $\hat{\pi}(\cdot, \mathcal{I}_{>1})$ forms a ‘good’ basis for the rows of $\hat{\pi}_1$ (in the i_1 variable) and choose a set $\mathcal{I}_{<2} = \{i_1^{\alpha_1}\}_{\alpha_1=1}^{r_1}$ of r_1 row indices such that the *volume* (the modulus of the determinant) of the $r_1 \times r_1$ submatrix $\hat{\pi}(\mathcal{I}_{<2}, \mathcal{I}_{>1})$ is *maximized*. This can be achieved in $\mathcal{O}(nr_1^2)$ operations using the *maxvol* algorithm (Goreinov et al. 2010). The first discrete TT block $\hat{\pi}^{(1)}$ is then assembled from the rectangular $n \times r_1$ matrix $\hat{\pi}(\cdot, \mathcal{I}_{>1})\hat{\pi}(\mathcal{I}_{<2}, \mathcal{I}_{>1})^{-1}$, and the

reduced tensor $[\hat{\pi}_{>1}(\alpha_1, i_2, \dots, i_d)] = [\hat{\pi}(i_1^{\alpha_1}, i_2, \dots, i_d)]$ is passed on to the next step of the TT-cross. In a practical algorithm, to ensure numerical stability all these operations are actually carried out using QR decompositions of the matrices (Oseledets and Tyrtshnikov 2010).

In the k th step, we assume that we are given the reduction $\hat{\pi}_{>k-1}(\alpha_{k-1}, i_k, \dots, i_d)$ from the previous step, as well as two sets $\mathcal{I}_{<k} = \{(i_1^{\alpha_{k-1}}, \dots, i_{k-1}^{\alpha_{k-1}})\}_{\alpha_{k-1}=1}^{r_{k-1}}$ and $\mathcal{I}_{>k} = \{(i_{k+1}^{\alpha_k}, \dots, i_d^{\alpha_k})\}_{\alpha_k=1}^{r_k}$ containing, resp., r_{k-1} ($k-1$)-tuples and r_k ($d-k$)-tuples. The unfolding tensor $[\hat{\pi}_{>k-1}(\alpha_{k-1}, i_k; \mathcal{I}_{>k})]$ can then be seen as a $r_{k-1}n \times r_k$ rectangular matrix and the *maxvol* algorithm can be applied again to produce a set of row positions $\{\alpha_{k-1}^{\alpha_k}, i_k^{\alpha_k}\}_{\alpha_k=1}^{r_k}$, which upon replacing $\alpha_{k-1}^{\alpha_k}$ with the corresponding indices from $\mathcal{I}_{<k}$ leads to the next index set $\mathcal{I}_{<k+1} = \{(i_1^{\alpha_k}, \dots, i_k^{\alpha_k})\}_{\alpha_k=1}^{r_k}$. The induction is completed by taking $\hat{\pi}^{(d)} = \hat{\pi}_{>d-1}$.

This process can be also organized in the form of a binary tree, which gives rise to the so-called hierarchical Tucker cross algorithm (Ballani and Grasedyck 2015). In total, we need $\mathcal{O}(dnr^2)$ evaluations of π and $\mathcal{O}(dnr^3)$ additional operations for the computation of the maximum volume matrices.

The choice of the univariate grids, $x_1^1 < \dots < x_k^{n_k}$, and of the initial index sets $\mathcal{I}_{>k}$ can be crucial. In this paper, we found that a uniform grid in each coordinate was sufficient, with even relatively coarse grids resulting in efficient sampling algorithms; see the numerical examples for details. Given any easy to sample reference distribution (e.g. uniform or Gaussian), it seems reasonable to initialize $\mathcal{I}_{>k}$ with independent realizations of that distribution. (We could also expand the *grids* with reference samples, though we did not do that.) If the target function π admits an *exact* TT decomposition with TT ranks not greater than r_1, \dots, r_{d-1} , and all unfolding matrices have ranks not smaller than the TT ranks of π , the cross iteration outlined above reconstructs $\hat{\pi}$ *exactly* (Oseledets and Tyrtshnikov 2010). This is still a rare exception though, since most functions have infinite exact TT ranks, even if they can be *approximated* by a TT decomposition with a small error and low ranks. Nevertheless, the cross iteration, initialized with slightly *overestimated* values r_1, \dots, r_{d-1} , can deliver a good approximation, if a function is regular enough (Ballani and Grasedyck 2015; Dolgov and Scheichl 2019).

This might be not the case for *localized* probability density functions. For example, for a heavy-tailed function $(1 + x_1^2 + \dots + x_d^2)^{-1/2}$ one might try to produce $\mathcal{I}_{>k}$ from a uniform distribution in a cube $[0, a]^d$ with a sufficiently large a . However, since this function is localized in an exponentially small volume $[0, \varepsilon]^d$, uniform index sets deliver a poor TT decomposition, worse for larger a and d .

Algorithm 1 TT-cross algorithm for TT approximation of π .

Input: Initial index sets $\mathcal{I}_{>k}$, rank increasing parameter $\rho \geq 0$, relative stopping tolerance $\delta > 0$ and/or maximum number of iterations iter_{\max} .

Output: TT blocks of an approximation $\tilde{\pi}(x) \approx \pi(x)$.

```

1: while iter < itermax and  $\|\tilde{\pi}_{\text{iter}} - \tilde{\pi}_{\text{iter}-1}\| > \delta \|\tilde{\pi}_{\text{iter}}\|$  do
2:   for  $k = 1, 2, \dots, d$  do ▷ Forward iteration
3:     (Optionally) prepare enrichment set  $\mathcal{I}_{>k}^{\text{aux}}$ .
4:     Compute  $r_{k-1}n \times r_k$  unfolding  $\hat{\pi}(\mathcal{I}_{<k}, i_k; \mathcal{I}_{>k})$ .
5:     Compute  $\mathcal{I}_{<k+1}$  by maxvol alg. and truncate.
6:   end for
7:   for  $k = d, d-1, \dots, 1$  do ▷ Backward iteration
8:     (Optionally) prepare enrichment set  $\mathcal{I}_{<k}^{\text{aux}}$ .
9:     Compute  $r_{k-1} \times nr_k$  unfolding  $\hat{\pi}(\mathcal{I}_{<k}; i_k, \mathcal{I}_{>k})$ .
10:    Compute  $\mathcal{I}_{>k-1}$  by maxvol alg. and truncate.
11:  end for
12: end while

```

In this situation, it is crucial to use fine grids and refine the sets $\mathcal{I}_{<k}, \mathcal{I}_{>k}$ by conducting *several* TT-cross iterations, going back and forth over the TT blocks and optimizing the sets by the *maxvol* algorithm. For example, after computing $\hat{\pi}^{(d)} = \hat{\pi}_{>d-1}$, we ‘reverse’ the algorithm and consider the unfolding matrices with indices $\{(i_1^{\alpha_{d-1}}, \dots, i_{d-1}^{\alpha_{d-1}})\}_{\alpha_{d-1}=1}^{r_{d-1}} = \mathcal{I}_{<d}$. Applying the *maxvol* algorithm to the *columns* of a $r_{d-1} \times n$ matrix $\hat{\pi}^{(d)}$, we obtain a *refined* set of points $\mathcal{I}_{>d-1} = \{i_d^{\alpha_{d-1}}\}_{\alpha_{d-1}=1}^{r_{d-1}}$. The recursion continues from $k = d$ to $k = 1$, optimizing the right sets $\mathcal{I}_{>k}$, while taking the left sets $\mathcal{I}_{<k}$ from the previous (forward) iteration. After several iterations, both $\mathcal{I}_{<k}$ and $\mathcal{I}_{>k}$ will be optimized to the particular target function, even if the initial index sets gave a poor approximation.

This adaptation of points goes hand in hand with the *adaptation of ranks*. If the initial ranks r_1, \dots, r_{d-1} were too large for the desired accuracy, they can be reduced. However, we can also *increase* the ranks by computing the unfolding matrix $[\hat{\pi}(\mathcal{I}_{<k}, i_k; i_{k+1}^{\alpha_k}, \dots, i_d^{\alpha_k})]$ on some *enriched* index set $\{(i_{k+1}^{\alpha_k}, \dots, i_d^{\alpha_k})\}_{\alpha_k=1}^{r_k+\rho}$, by augmenting the original index set $\mathcal{I}_{>k}$ with an *auxiliary* set $\mathcal{I}_{>k}^{\text{aux}}$ and increasing the k th TT rank from r_k to $r_k + \rho$. The auxiliary set can be chosen at random (Oseledets 2011a) or using a surrogate for the error (Dolgov and Savostyanov 2014). The pseudocode of the entire TT-cross method is listed in Algorithm 1. For uniformity, we let $\mathcal{I}_{<1} = \mathcal{I}_{>d} = \emptyset$.

Systematically using the enrichment scheme, we can even employ a different approach moving away from truncating ranks. Instead, we start with a low-rank initial guess and increase the ranks until the desired accuracy is met. We have found that this approach is often more accurate in numerical experiments. The relative cost of the two approaches depends on the application.

4 Sampling algorithms based on TT Surrogates

4.1 Conditional distribution sampling (TT-CD)

One of the main contributions of this paper is to show that conditional distribution method is feasible, and efficient, once a PDF has been put into TT format. This section presents those calculations.

First, we describe the computation of the marginal PDFs p_k , defined in (2), given π in a TT format (3). Note that integrals over the variable x_p appear in all conditionals (2) with $k < p$. The TT format allows to compute the $r_{k-1} \times 1$ vector P_k required for evaluating the marginal PDF p_{k-1} by the following algorithm.

- 1: Initialize $P_{d+1} = 1$
- 2: **for** $k = d, d - 1, \dots, 2$ **do**
- 3: $(P_k)_{\alpha_{k-1}} = \sum_{\alpha_k=1}^{r_k} \left(\int_{\mathbb{R}} \pi_{\alpha_{k-1}, \alpha_k}^{(k)}(x_k) dx_k \right) (P_{k+1})_{\alpha_k}$
- 4: **end for**

Since $\pi^{(k)}(x_k) \in \mathbb{R}^{r_{k-1} \times r_k}$ for each fixed x_k , the integral $\int \pi^{(k)}(x_k) dx_k$ is a $r_{k-1} \times r_k$ matrix, where α_{k-1} is the row index and α_k is the column index. Hence, we can write Line 3 as the matrix–vector product,

$$P_k = \left(\int_{\mathbb{R}} \pi^{(k)}(x_k) dx_k \right) P_{k+1}.$$

Assuming n quadrature points for each x_k , and the uniform rank bound $r_k \leq r$, the asymptotic complexity of this algorithm is $\mathcal{O}(dnr^2)$.

The first marginal PDF is approximated by $p_1^*(x_1) = |\pi^{(1)}(x_1)P_2|$. We take the absolute value because the TT approximation $\tilde{\pi}$ (and hence, $\pi^{(1)}(x_1)P_2$) may be negative at some locations. In the k th step of the sampling procedure, the marginal PDF also requires the first $k - 1$ TT blocks, restricted to the components of the sample that are already determined,³

$$p_k^*(x_k) = \left| \pi^{(1)}(x_1) \cdots \pi^{(k-1)}(x_{k-1}) \pi^{(k)}(x_k) P_{k+1} \right|.$$

However, since the loop goes sequentially from $k = 1$ to $k = d$, the sampled TT blocks can be accumulated in the same fashion as the integrals P_k . Again, we take the absolute value to ensure positivity. The overall method for drawing N samples is written in Algorithm 2. Note that if $\tilde{\pi}$ is negative at any points, the actual density π^* at x^ℓ , which is the product of marginal PDFs computed in each step, may slightly differ from $\tilde{\pi}$.

³ Here again, we treat $\pi^{(k)}(x_k)$ as a $r_{k-1} \times r_k$ matrix, such that the product is valid.

Algorithm 2 CD sampling from a TT decomposition of a PDF

- Input:** TT blocks $\pi^{(1)}, \dots, \pi^{(d)}$ of the approximation $\tilde{\pi}$, uniformly distributed seeds $\{(q_1^\ell, \dots, q_d^\ell)\}_{\ell=1}^N \sim \mathcal{U}(0, 1)^d$.
- Output:** π^* -distributed samples $\{(x_1^\ell, \dots, x_d^\ell)\}_{\ell=1}^N$, as well as PDF values $\pi^*(x^\ell) = p_1^*(x_1^\ell) \cdots p_d^*(x_d^\ell)$.
- 1: Initialize $P_{d+1} = 1$.
 - 2: **for** $k = d, d - 1, \dots, 2$ **do**
 - 3: Compute $P_k = \int_{\mathbb{R}} \pi^{(k)}(x_k) dx_k \cdot P_{k+1}$.
 - 4: **end for**
 - 5: Initialize $\Phi_1 = 1 \in \mathbb{R}^N$.
 - 6: **for** $k = 1, 2, \dots, d$ **do**
 - 7: Prepare deterministic part $\Psi_k(x_k) = \pi^{(k)}(x_k) P_{k+1}$.
 - 8: **for** $\ell = 1, \dots, N$ **do**
 - 9: Compute marginal PDF $p_k^*(x_k) = |\Phi_k(\ell, \cdot) \Psi_k(x_k)|$,
 - 10: marginal CDF $C_k(x_k) = \int_{-\infty}^{x_k} p_k^*(y_k) dy_k / \int p_k^* dx_k$.
 - 11: Sample x_k component, $x_k^\ell = C_k^{-1}(q_k^\ell)$.
 - 12: Compute $\Phi_{k+1}(\ell, \cdot) = \Phi_k(\ell, \cdot) \pi^{(k)}(x_k^\ell)$.
 - 13: **end for**
 - 14: **end for**

The error induced by taking the absolute values in Line 9 of Algorithm 2 is of the order of the TT approximation error. The approximate marginal probability $\tilde{p}_k(x_k) = \Phi_k(\ell, \cdot) \Psi_k(x_k)$ is produced from $\tilde{\pi}(x)$ by integration; hence, if $\pi(x) - \tilde{\pi}(x) = \mathcal{O}(\varepsilon)$ due to the TT approximation, we also have $|\tilde{p}_k(x_k) - p_k(x_k)| \leq C\varepsilon \|p_k\|_\infty$ for the marginals, for some $C > 0$ independent of ε , where $\|p_k\|_\infty := \text{ess sup}_{\xi_k} p_k(\xi_k)$. Then, for all x_k that satisfy $p_k(x_k) \geq C\varepsilon \|p_k\|_\infty$, we have

$$\tilde{p}_k(x_k) \geq C\varepsilon \|p_k\|_\infty - |p_k(x_k) - \tilde{p}_k(x_k)| \geq 0.$$

Hence, \tilde{p}_k can only be negative where p_k is small and we have $-C\varepsilon \|p_k\|_\infty \leq \tilde{p}_k(x_k) \leq 0$. The error in taking the modulus in Line 9 of Algorithm 2 can then be estimated as follows:

$$|p_k^*(x_k) - \tilde{p}_k(x_k)| \leq \begin{cases} 2C\varepsilon \|p_k\|_\infty, & \text{for } \tilde{p}_k(x_k) < 0, \\ 0, & \text{otherwise.} \end{cases}$$

The sample-independent prefactor of the marginal PDF in Line 7 requires $\mathcal{O}(dnr^2)$ operations. The marginal PDF in Line 9 can then be computed with $\mathcal{O}(dNnr)$ cost. The cost of the CDF computation in Line 10 depends on the quadrature scheme used. For a piecewise spline approximation or for the barycentric Gauss formula, the cost for both C_k and C_k^{-1} is $\mathcal{O}(dNn)$. The complexity of computing the conditional PDF values Φ_{k+1} depends on how $\tilde{\pi}$ is interpolated onto x_k^ℓ . Global Lagrange interpolation requires $\mathcal{O}(nr^2)$ cost per sample, whereas local interpolation is independent of n , requiring only $\mathcal{O}(r^2)$ operations. In our numerical experiments, we have found piecewise linear interpolation on a uniform grid to be sufficient. In summary, the total complexity is

$$\mathcal{O}(dr(nr + N(n + r))). \tag{6}$$

4.2 Metropolis–Hastings correction (TT-MH)

For the TT-CD sampling procedure in Algorithm 2 to be fast, the TT ranks r should be as small as possible. Since the joint PDF is typically a complicated multivariate function, its TT ranks may grow fast with the increasing accuracy. On the other hand, low accuracy is typically sufficient if we ‘correct’ the distribution using the Metropolis–Hastings (MH) algorithm to ensure that the samples are distributed according to the target distribution π . Thus, we first propose to use a coarse TT approximation together with TT-CD sampling as independence proposals in a MH algorithm.⁴

When the current state is x and the new proposal is x' , the next state is determined by the stochastic iteration that first computes the Metropolis–Hastings ratio

$$h(x, x') = \frac{\pi(x') \pi^*(x)}{\pi(x) \pi^*(x')},$$

and the proposal is *accepted* with probability

$$\alpha(x, x') = \min(h(x, x'), 1), \tag{7}$$

putting the new state $x = x'$; otherwise, x' is *rejected* and the chain remains at x . Note that in the actual computations we calculate $h(x, x')$ using the log-densities $\log \pi(x)$ and $\log \pi^*(x) = \log p_1^*(x_1) + \dots + \log p_d^*(x_d)$ (see Algorithm 2) for stability reasons.

As efficiency indicators of this MH algorithm for estimating the expected value $\mathbb{E}_\pi g$ of some functional $g(x)$, we consider the acceptance rate and the integrated autocorrelation time. In this section, we study how they depend on the approximation error in the PDF. Throughout we must assume that π is absolutely continuous with respect to π^* , which guarantees reversibility with respect to π (Tierney 1998), and that we can evaluate the importance ratio $w(x) = \pi(x)/\pi^*(x)$. We require that $w^* \equiv \|w\|_\infty < \infty$, which is equivalent to uniform geometric convergence (and ergodicity) of the chain (Roberts and Rosenthal 2011). (The essential supremum may be taken with respect to π or π^* .)

To simplify the presentation in this subsection, we assume again (without loss of generality) that the density is normalized.

Lemma 1 *Suppose that $\pi(x)$ is normalized and that the mean absolute error in the TT-CD sampling density satisfies*

$$\int |\pi^*(x) - \pi(x)| dx \leq \varepsilon/2.$$

⁴ A more simple scheme may be to use a multiple of π^* to bound π and then use a rejection algorithm. However, as noted in Liu (1996), the MH is more statistically efficient.

Then the rejection rate is bounded by ε , i.e.

$$\mathbb{E} [1 - \alpha(x, x')] \leq \varepsilon,$$

where the expectation is taken over the chain.

Proof Using ergodicity of the chain,

$$\mathbb{E} [1 - \alpha(x, x')] = \iint [1 - \alpha(x, x')] \pi(x) \pi^*(x') dx dx'.$$

Since $1 - \alpha \leq |1 - h|$,

$$\begin{aligned} [1 - \alpha(x, x')] \pi(x) \pi^*(x') &\leq |\pi(x) \pi^*(x') - \pi(x') \pi^*(x)| \\ &\leq \pi(x) |\pi^*(x') - \pi(x')| \\ &\quad + \pi(x') |\pi^*(x) - \pi(x)|, \end{aligned}$$

where the second step uses the triangle inequality. Integrating both sides with respect to x and x' , we obtain the claim of the lemma. \square

This lemma indicates that the rejection rate decreases proportionally to ε , where ε is the total error due to approximating π by a low-rank TT decomposition $\tilde{\pi}$, interpolating discrete values of $\tilde{\pi}$ on a grid, and taking the absolute values in Algorithm 2, Line 9.

Lemma 1 assumed a *mean* absolute error. We need the stronger statement of *local* relative error, which is $w^* < \infty$, to bound the integrated autocorrelation time (IACT) (Wolff 2004), defined as

$$\tau = \left(1 + 2 \sum_{t=1}^{\infty} \rho_{gg}(t) \right), \tag{8}$$

where $\rho_{gg}(t)$ is the autocorrelation coefficient for the chain in statistic g at lag t . Defined like this, $\tau \geq 1$ can be considered as a reduction factor in the efficiency of a particular MCMC chain compared to an ideal independent chain, asymptotically as the length of the chain goes to infinity. Note that $w^* < \infty$ implies that TT-MH is uniformly ergodic, but conversely the MCMC is not even geometrically ergodic if $w^* = \infty$ (Mengersen and Tweedie 1996, Thm. 2.1).

Lemma 2 *When $w^* < \infty$, for any $g \in L^2(\pi)$,*

$$\tau \leq \frac{1+a}{1-a},$$

where $a = 1 - 1/w^$.*

Proof Without loss of generality, we may consider $g \in L_0^2(\pi)$, i.e. $E_\pi[g] = 0$ (see, for example, Mira and Geyer 1999). Consider the transition kernel

$$P_a(x, dy) = (1 - a)\pi(dy) + a\delta_x(dy).$$

(This is the chain that proposes from π and accepts with probability $(1 - a)$.) P_a has a simple spectrum, consisting of 1, with right eigenvector $\mathbf{1}$, and a for the orthogonal complement. Hence, the asymptotic variance in a central limit theorem (CLT) for the chain in $g \in L^2_0(\pi)$ induced by P_a may be evaluated using the spectral measure (see, for example, Mira and Geyer 1999; Häggström and Rosenthal 2007), which reads $\mathcal{E}_g(S) = \delta_a(S)$ in this case, giving IACT equal to $(1+a)/(1-a)$. The transition kernel for the TT-MH chain is (Smith and Tierney 1996, Thm. 1 & Lem. 3)

$$P(x, dy) = \min(1/w(x), 1/w(y))\pi(dy) + \lambda(w(x))\delta_x(dy),$$

with λ given by Smith and Tierney (1996, Eq (5)). Since $\min(1/w(x), 1/w(y)) \geq 1/w^*$, P dominates P_a , in the sense of Peskun ordering (Tierney 1998; Mira and Geyer 1999), i.e. the off-diagonal terms in P are greater or equal than those in P_a , and hence, the IACT using P is less or equal than that using P_a (Mira and Geyer 1999, Thm 3.4). \square

For discrete state spaces, the result in Lemma 2 follows directly from Mira (2001, Eqn. (2.1)), while one could argue that this is sufficient for practical computation since computers are finite-dimensional.

The TT-cross method tends to introduce a more or less uniform error of magnitude ε on average. For regions where $\pi(x) \gg \varepsilon$, this leads to a bounded importance ratio $w(x) \leq 1 + \mathcal{O}(\varepsilon)$. When $\pi(x) \ll \varepsilon$, we will typically have $\pi^*(x) = \mathcal{O}(\varepsilon)$ and $w(x) < 1$. However, if $\pi(x) \approx \varepsilon$ and a negative error of order ε is committed, the two may cancel, resulting in a small $\pi^*(x)$ and consequently in a large $w(x)$. Numerical experiments demonstrate that $w^* - 1$ can indeed be much larger than the L_1 -norm error used in Lemma 1 (see Fig. 1). However, these cancellations (and hence the equality in $\min(1/w(x), 1/w(y)) \geq 1/w^*$) seem to be rare (see Fig. 2). Moreover, the practical IACT tends to be much smaller than the upper bound given by Lemma 2.

4.3 QMC samples and importance weights (TT-qIW)

Due to the central limit theorem, the rate of convergence of the statistical error of a Monte Carlo estimator for $\mathbb{E}_\pi g$, as the number of samples $N \rightarrow \infty$, is limited to $\mathcal{O}(N^{-1/2})$. The IACT of the chain induced by a MH sampler, such as the TT-MH sampler in the previous section, only affects the constant in this estimate.

Thus, it is tempting to use more structured quadrature points to obtain a better convergence rate. For example, the TT approximation of π provides the possibility to reduce the inherent multivariate integrals to a sequence of univariate integrals, as we did when forming the marginal distributions in Sec. 4.1, and use, for example, Gauss quadrature. Another option is to note that the TT-CD map is also well defined for

other seed points, such as those taken from a quasi-Monte Carlo (QMC) rule, that is, $\{(q_1^\ell, \dots, q_d^\ell)\}_{\ell=1}^N$ in Algorithm 2 are taken from a QMC lattice in $[0, 1]^d$, rather than i.i.d. samples from $\mathcal{U}(0, 1)^d$. Under certain assumptions on the smoothness of the quantity of interest, the QMC quadrature can give an error that converges with order N^{-1} instead of $N^{-1/2}$ when $N \rightarrow \infty$ (Niederreiter 1978; Dick et al. 2013). However, both those approaches provide estimates for $\mathbb{E}_{\pi^*} g$, which are biased due to the TT approximation, and this bias cannot be ‘corrected’ using a MH step, as for i.i.d. seeds. On the other hand, there are no suitable convergence results for MH algorithms based on QMC proposals.

A classical way to remove the bias in the estimate is via importance re-weighting. Writing the expectation as an integral, then multiplying and dividing by the approximate density function, gives

$$\mathbb{E}_\pi g = \frac{1}{Z} \int g(x)\pi(x)dx = \frac{1}{Z} \int g(x)w(x)\pi^*(x)dx, \quad (9)$$

where $Z = \int \pi(x)dx$ is the normalization constant and $w(x) = \pi(x)/\pi^*(x)$ is the importance weight. That is, the expectation of g with respect to π equals the expectation of the weighted function $g(x)w(x)$ with respect to the approximate density π^* . The normalization constant can be rewritten as $Z = \int w(x)\pi^*(x)dx$. Similarly to (7), we calculate $w(x) = \exp(\log \pi(x) - \log \pi^*(x))$ using log-densities in practice.

Thus, given a set of samples $\{x^\ell\}_{\ell=1}^N \sim \pi^*$ produced using the TT-CD algorithm, either from a set of i.i.d. samples on $[0, 1]^d$ or from a QMC lattice, we compute

$$\mathbb{E}_\pi g \approx \frac{1}{Z} \left(\frac{1}{N} \sum_{\ell=1}^N g(x^\ell)w(x^\ell) \right), \quad \tilde{Z} := \frac{1}{N} \sum_{\ell=1}^N w(x^\ell). \quad (10)$$

Note that, since $x^\ell \sim \pi^*$, the weight $w(x^\ell) < \infty$ with probability 1, and hence, the importance quadrature (10) is well defined. The convergence depends on the distance between $|\pi^* - \pi|$ and on the choice of samples x^ℓ . Most importantly, if the seeds $\{q^\ell\}$ for the TT-CD samples $\{x^\ell\}$ in Algorithm 2 are chosen according to a randomized QMC rule, and the integrand $g(x)w(x)$ is sufficiently smooth, we can expect a rate of convergence close to $\mathcal{O}(N^{-1})$, the estimator is unbiased, and under the right smoothness assumptions, the convergence rate is dimension independent (Dick et al. 2013).

4.4 Multilevel acceleration

Following recent works on multilevel MCMC (Hoang et al. 2013; Dodwell et al. 2015), we can also use the (cheap) surrogate π^* as a type of control variate to achieve variance reduction in the estimator.

In addition to π^* , we may also have a cheap ‘surrogate’ \tilde{g} for the integrand g . For example, in Sect. 5.3, we will build a TT surrogate $\tilde{u}_h(\theta)$ of the FE solution $u_h(\theta)$ of the stochastic diffusion equation, as a function of the stochastic parameters θ , that allows for a cheap approximation $\tilde{g}(\theta) = \phi(\tilde{u}_h)$ of any functional $g(\theta) = \phi(u_h)$ of the PDE solution, without having to solve the PDE for each sample. Otherwise, let $\tilde{g} = g$.

To exploit the multilevel ideas, we observe that

$$\mathbb{E}_\pi g = \mathbb{E}_{\pi^*} \tilde{g} + \left[\mathbb{E}_\pi g - \mathbb{E}_{\pi^*} \tilde{g} \right] \tag{11}$$

$$= \mathbb{E}_{\pi^*} \tilde{g} + \mathbb{E}_{\pi^*} \left[\frac{1}{\mathbb{E}_{\pi^*} w} g w - \tilde{g} \right]. \tag{12}$$

As in the previous section, given a set of N_0 samples $\{x^\ell\}_{\ell=1}^{N_0} \sim \pi^*$ produced using the TT-CD algorithm, the first term in (11) and (12) can be estimated by

$$\mathbb{E}_{\pi^*} \tilde{g} \approx \frac{1}{N_0} \sum_{\ell=1}^{N_0} \tilde{g}(x^\ell). \tag{13}$$

Since the expected value in (13) is with respect to π^* , no MH correction is necessary. Moreover, we can use, as in Sect. 4.3, QMC seed points $\{q^\ell\} \subset [0, 1]^d$ for the TT-CD samples $\{x^\ell\}$ in Algorithm 2, leading to a much faster convergence of the estimator with respect to N_0 . Alternatively, if both $\tilde{\pi}$ and \tilde{g} are represented in the TT format, (13) can be computed without sampling as the dot product of TT tensors as in Eigel et al. (2018).

In fact, if the evaluation of \tilde{g} is significantly faster than the evaluation of g , as in the stochastic diffusion problem below, the cost of estimating the first term in (11) and (12) becomes entirely negligible.

To estimate the second term in (11) and (12), we now proceed as in Sects. 4.2 and 4.3, respectively.

First consider a set of i.i.d. samples $\{x^\ell\}_{\ell=1}^{N_1} \sim \pi^*$, computed using Algorithm 2, and let $\{x_{\text{MH}}^\ell\}_{\ell=1}^{N_1}$ be the Markov chain of samples distributed according to π after Metropolis–Hastings ‘correction’ of $\{x^\ell\}_{\ell=1}^{N_1}$ using the acceptance probability defined in (7). We can define the following unbiased estimator:

$$\mathbb{E}_\pi g - \mathbb{E}_{\pi^*} \tilde{g} \approx \frac{1}{N_1} \sum_{\ell=1}^{N_1} g(x_{\text{MH}}^\ell) - \tilde{g}(x^\ell). \tag{14}$$

If $\pi^* \approx \pi$ and $\tilde{g} \approx g$, the pairs of samples $(\tilde{g}(x^\ell), g(x_{\text{MH}}^\ell))$ are strongly, positively correlated, and thus, the variance of $g(x_{\text{MH}}^\ell) - \tilde{g}(x^\ell)$ is much smaller than the variance of $g(x_{\text{MH}}^\ell)$. As a consequence, the number of samples N_1 necessary to achieve a prescribed statistical error can be chosen significantly smaller than in Sect. 4.2.

Alternatively, consider now the second term in (12) and let $\{x^\ell\}_{\ell=1}^{N_1}$ be obtained via Algorithm 2 from a set of N_1

randomized QMC seed points $\{q^\ell\}_{\ell=1}^{N_1} \subset [0, 1]^d$. Then we can define the following unbiased estimator:

$$\mathbb{E}_{\pi^*} \left[\frac{1}{\mathbb{E}_{\pi^*} w} g w - \tilde{g} \right] \approx \frac{1}{N_1} \sum_{\ell=1}^{N_1} \frac{1}{\tilde{Z}} g(x^\ell) w(x^\ell) - \tilde{g}(x^\ell). \tag{15}$$

Again, if $\pi^* \approx \pi$ and $\tilde{g} \approx g$ then $w \approx 1$ and the variance of $g(x^\ell)w(x^\ell)/\tilde{Z} - \tilde{g}(x^\ell)$ is small, so that the number of samples N_1 can be chosen significantly smaller than the number N of samples in (10). Moreover, since $\tilde{Z} = 1 + \frac{1}{N_1} \sum_{\ell=1}^{N_1} (w(x^\ell) - 1)$ and $\mathbb{V}_{\pi^*}[w - 1]$ is small, a small value for N_1 is also sufficient for the calculation of \tilde{Z} in (10). If $g w / \tilde{Z} - \tilde{g}$ is sufficiently smooth, the rate of convergence of the sampling error as $N_1 \rightarrow \infty$ should again be close to $\mathcal{O}(N_1^{-1})$. However, in contrast to the estimator in (10), we do not observe that better rate of convergence for the difference estimator in (15).

It would be possible to further optimize the complexity of the estimators in (13), (14) and (15) by a judicious choice of the TT accuracy ε , as well as the numbers of samples N_0 and N_1 . There is of course also scope for full multilevel estimators as in Hoang et al. (2013); Dodwell et al. (2015). In particular, the values of N_0 and N_1 can be determined by an adaptive greedy procedure (Kuo et al. 2017), which compares empirical variances and costs of the two levels and doubles N_ℓ on the level that has the maximum profit. However, we will not consider this further and leave it for future works.

5 Numerical examples

5.1 Shock absorber reliability

In this section, we demonstrate our algorithm on a problem of reliability estimation of a shock absorber. The time to failure of a type of shock absorber depends on some environmental conditions (covariates) such as humidity and temperature. We use data (O’Connor and Kleyner 2012) on the distance (in kilometres) to failure for 38 vehicle shock absorbers. Since there were no values of any covariates in this example, the values of D covariates were synthetically generated from the standard normal distribution as this would correspond to the case in which the covariates have been standardized to have zero mean and variance equal to one. The accelerated failure time regression model (Meeker and Escobar 1998) is widely used for reliability estimation with covariates. We use an accelerated failure time Weibull regression model, which was described as reasonable for these data in Meeker and Escobar (1998), where the density of time to failure is of the form

$$f(t|\theta_1, \theta_2) = \frac{\theta_2}{\theta_1} \left(\frac{t}{\theta_1}\right)^{\theta_2-1} \exp\left(-\left(\frac{t}{\theta_1}\right)^{\theta_2}\right)$$

and where θ_1, θ_2 are unknown scale and shape hyperparameters, respectively. The covariates are assumed to affect the failure time distribution only through the scale parameter θ_1 , via a standard logarithmic link function, that is

$$\theta_1(\beta_0, \dots, \beta_D) = \exp\left(\beta_0 + \sum_{k=1}^D \beta_k x_k\right),$$

where x_k are the covariates. The $D + 2$ unknown parameters β_0, \dots, β_D and θ_2 must be inferred from the observation data on the covariates x_k and the failure times, which in this example are subject to right censoring (marked with $^+$). The set T_f of failure times is given by:

6700	6950 ⁺	7820 ⁺	8790 ⁺	9120
9660 ⁺	9820 ⁺	11310 ⁺	11690 ⁺	11850 ⁺
11880 ⁺	12140 ⁺	12200	12870 ⁺	13150
13330 ⁺	13470 ⁺	14040 ⁺	14300	17520
17540 ⁺	17890 ⁺	18420 ⁺	18960 ⁺	18980 ⁺
19410 ⁺	20100	20100 ⁺	20150 ⁺	20320 ⁺
20900	22700	23490 ⁺	26510	27410 ⁺
27490	27890 ⁺	28100 ⁺		

To perform Bayesian inference on the unknown parameters, we use the prior specifications in Gutierrez-Pulido et al. (2005), namely an s -normal Gamma distribution $\pi_0(\beta_0, \dots, \beta_D, \theta_2)$ given by

$$\pi_0 = \frac{1}{Z} \theta_2^{\alpha-0.5} \prod_{k=0}^D \exp\left(-\frac{\theta_2(\beta_k - m_k)^2}{2\sigma_k^2}\right) \exp(-\gamma\theta_2),$$

where $\gamma = 2.2932, \alpha = 6.8757, m_0 = \log(30796), \sigma_0^2 = 0.1563, m_1 = \dots = m_D = 0, \sigma_1 = \dots = \sigma_D = 1$ and Z is the normalization constant. The parameter ranges

$$[m_0 - 3\sigma_0, m_0 + 3\sigma_0] \times [m_1 - 3\sigma_1, m_1 + 3\sigma_1]^D \times [0, 13]$$

are large enough to treat the probability outside as negligible.

The (unnormalized) Bayesian posterior density function is given by a product of Weibull probabilities, evaluated at each observation in T_f , and the prior distribution, i.e.

$$\pi(\beta, \theta_2) = \pi_0(\beta, \theta_2) \prod_{t \in T_f} P(t|\theta_1(\beta), \theta_2),$$

where

$$P(t|\theta_1, \theta_2) = \begin{cases} f(t|\theta_1, \theta_2) & \text{if } t \text{ is not censored,} \\ \exp\left(-\left(\frac{t}{\theta_1}\right)^{\theta_2}\right) & \text{if } t \text{ is censored.} \end{cases}$$

The formula for the censored case arises from the fact that the contribution of a censored measurement is the probability that t exceeds the measured value, that is, $P(t \geq t^+|\theta_1, \theta_2) = \int_{t^+}^{\infty} f(t|\theta_1, \theta_2)dt$. We introduce n uniform discretization points in β_0, \dots, β_D and θ_2 and compute the TT-cross approximation of the discretized density $\pi(\beta_0, \dots, \beta_D, \theta_2)$.

We consider two quantities of interest, the right 95% mean quantile and the right 95% quantile of the mean distribution, i.e.

$$\langle q(f) \rangle = \frac{1}{N} \sum_{i=1}^N \theta_1^i \log^{1/\theta_2^i}(1/0.05), \quad \text{and} \tag{16}$$

$$q(\langle f \rangle) = t \quad \text{s.t.} \quad \frac{1}{N} \sum_{i=1}^N \int_0^t f(s|\theta_1^i, \theta_2^i)ds = 0.95,$$

respectively. The nonlinear constraint in the computation of the second quantile is solved by Newton’s method. To estimate the quadrature error, we perform 32 runs of each experiment and compute an average relative error over all runs, i.e.

$$\mathcal{E}_q = \frac{1}{32} \sum_{i=1}^{32} \frac{\left|q(\langle f \rangle)_i - \frac{1}{32} \sum_{\ell=1}^{32} q(\langle f \rangle)_\ell\right|}{\frac{1}{32} \sum_{\ell=1}^{32} q(\langle f \rangle)_\ell}, \tag{17}$$

where i and ℓ enumerate different runs.

The error in the mean quantile is estimated similarly, and then, the average of those two error estimates is used in all our convergence studies. We used quantiles as the quantity of interest in order to illustrate that the TT surrogate captures the tails correctly.

5.1.1 Accuracy of TT approximation and CD sampler

We start by analysing the TT-MH sampling procedure, as described in Sect. 4.2. First, we consider how the errors in $\tilde{\pi}$ due to the tensor approximation and discretization propagate into the quality of the MCMC chain produced by the MH algorithm, i.e. the rate of rejections and the integrated autocorrelation time. The chain length is always set to $N = 2^{20}$, and the results are averaged over 32 runs. We choose a relatively low dimensionality $D = 2$, since it allows us to approximate π up to a high accuracy.

In Fig. 1, we vary the number of grid points n , fixing the relative stopping tolerance for the TT-cross algorithm at $\delta = 10^{-5}$, as well as benchmarking the algorithm for

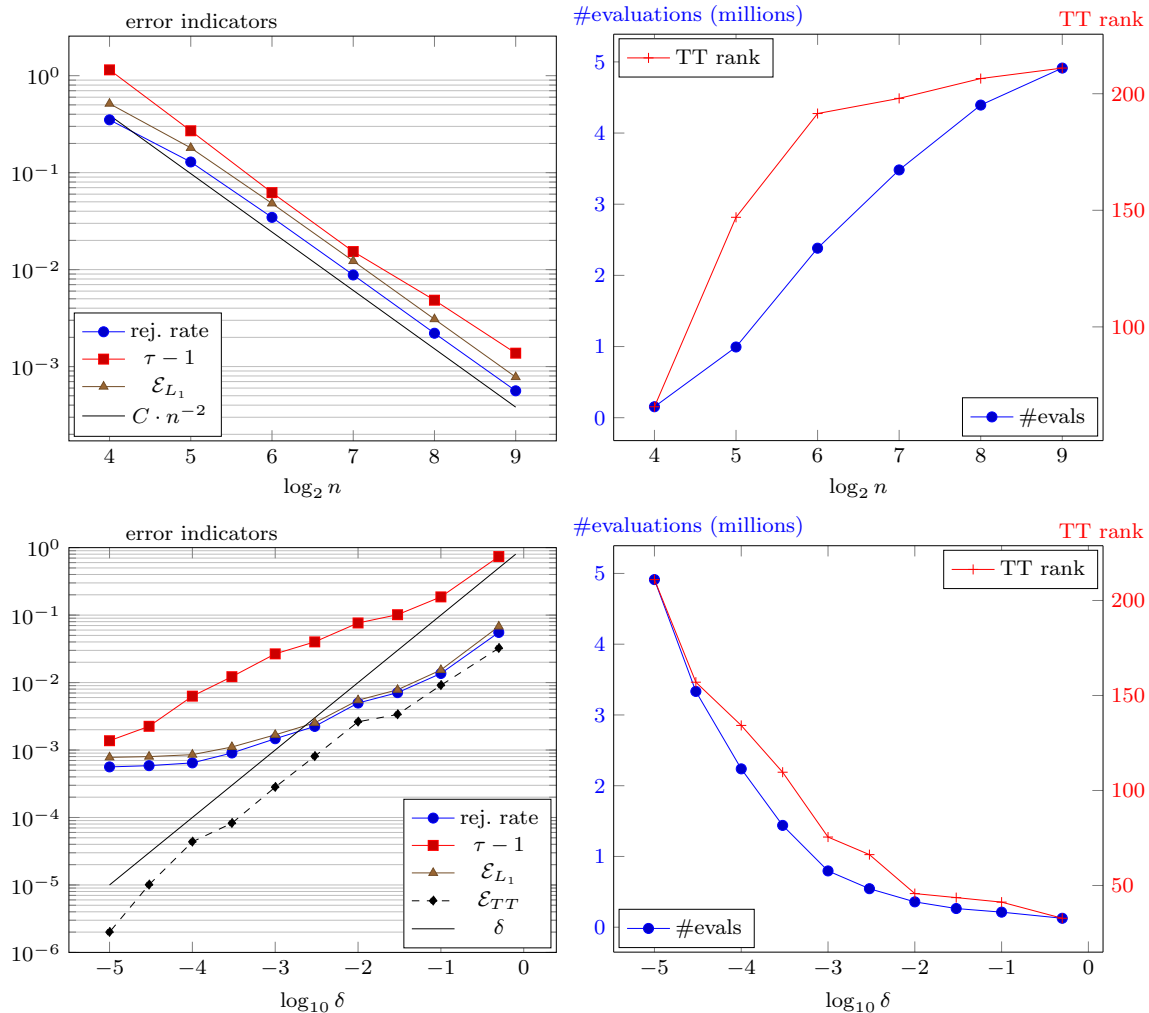


Fig. 1 Shock absorber example ($D = 2$): rejection rate, IACT and estimated errors (left), numbers of evaluations of π and maximal TT ranks for TT-cross (right) plotted against the grid size n in each direction (top) and against the relative TT error tolerance δ (bottom)

different thresholds δ , fixing $n = 512$. We track the relative empirical standard deviation of the TT approximation,

$$\mathcal{E}_{TT} = \sqrt{\frac{\frac{1}{31} \sum_{\ell=1}^{32} \left\| \tilde{\pi}_\ell - \frac{1}{32} \sum_{\ell=1}^{32} \tilde{\pi}_\ell \right\|_F^2}{\left\| \frac{1}{32} \sum_{\ell=1}^{32} \tilde{\pi}_\ell \right\|_F^2}}, \quad (18)$$

that can be computed exactly in the TT representation, as well as an importance-weighted QMC approximation to the L_1 -norm error used in Lemma 1,

$$\mathcal{E}_{L_1} = \frac{1}{N} \sum_{\ell=1}^N \left| w(x^\ell) - 1 \right| \approx \int |\pi(x) - \pi^*(x)| dx. \quad (19)$$

As shown in Lemma 1, the rejection rate is expected to be proportional to the approximation error in L_1 norm, as this error goes to zero. The TT approximation is computed on a tensor grid with n vertices and uses linear interpolation to

evaluate $\tilde{\pi}$ at intermediate values. Thus, it can be expected that the convergence of the interpolation error, as $n \rightarrow \infty$, should be of $\mathcal{O}(n^{-2})$, provided π is sufficiently smooth. Figure 1 (top left) shows that the rejection rate converges with $\mathcal{O}(n^{-2})$, suggesting that this is the case here. Figure 1 (bottom left) also suggests that the rejection rate is proportional to the TT approximation error when it is greater than the interpolation error.

The behaviour of the importance ratio and the integrated autocorrelation time (IACT) is more complicated. The IACT τ is tracked in Fig. 1 as well, while the essential supremum w^* of the importance ratio is shown in Table 1. The TT-cross algorithm tries to reduce the average approximation error. Pointwise relative error, however, is not guaranteed to be bounded. Although $w^* \rightarrow 1$ as $\delta \rightarrow 0$, it is orders of magnitude larger than \mathcal{E}_{L_1} . Regardless, Lemma 2 seems to give a too pessimistic estimate for the IACT, as the actual value $\tau - 1$ is much smaller than $w^* - 1$ and behaves similarly

Table 1 Shock absorber example ($D = 2$), maximal importance ratios for fixed $\delta = 10^{-5}$ and varying n (top) and for fixed $n = 512$ and varying δ (bottom)

n	32	64	128	256	512
w^*	2.49	1.57	2.04	1.79	2.87
δ	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
w^*	80.5	50.0	15.6	5.63	2.87

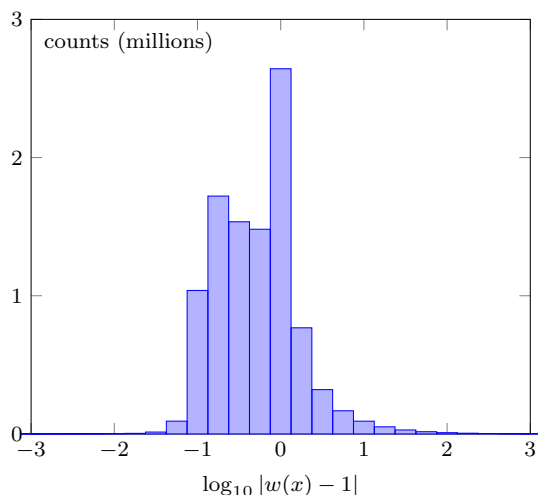


Fig. 2 Shock absorber example ($D = 2$), distribution of pointwise errors $|w - 1|$ over 10^7 samples, $n = 64$, $\delta = 10^{-3}$

to the rejection rate. The reason for this is that very large values of w are quite rare, as we show in Fig. 2.

The complexity of the TT-cross algorithm (in terms of both the number of evaluations of π and the computational time) grows only very mildly (sublinearly) with δ and n (notice the log-polynomial scale in Fig. 1, right). This makes the TT approach also well scalable for high accuracies.

5.1.2 Convergence studies and comparison with DRAM

Now we investigate the convergence of the quantiles and compare TT-MH with the delayed rejection adaptive Metropolis (DRAM) algorithm (Haario et al. 2006). The initial covariance for DRAM is chosen to be the identity matrix. In order to eliminate the effect of the burn-in period, we do not include the first $N/4$ elements of the DRAM chain in the computation of the quantiles. However, we will study the actual burn-in time empirically to have a fairer comparison of the ‘set-up cost’ of the two methods.

First, in Table 2, we fix $D = 6$ covariates and vary the discretization grid n and the TT approximation threshold δ . We present the rejection rates and the IACTs for TT-MH, with $n = 12, 16$ and 32 grid points in each direction, using values of $\delta = 0.5$ and $\delta = 0.05$, as well as for DRAM. In

Table 2 Comparison of TT-MH and DRAM; rejection rate, IACT and number of function evaluations to set up TT-cross and to burn in DRAM for the shock absorber ($D = 6$)

n	TT-MH				DRAM
	12	16	16	32	
δ	0.5	0.5	0.05	0.05	
rej. rate	0.61	0.33	0.28	0.12	0.5
τ	13.76	4.24	2.94	2.15	24.85
N_{setup}	35158	44389	101564	221116	49200

addition, we also give the set-up cost in terms of numbers of evaluations of π , i.e. the number of points needed to construct the TT approximation via the TT-cross algorithm for TT-MH and the burn-in in DRAM. The latter is estimated as the point of stabilization of 6 moments of β and θ_2 , approximated by averaging over 2^{14} random initial guesses. The coarsest TT approximation requires about $4 \cdot 10^4$ evaluations, whereas DRAM needs a burn-in of about $5 \cdot 10^4$ steps.

Next, in Fig. 3 (left) we show the estimate \mathcal{E}_q of the quadrature error defined in (17) for the two quantities of interest in (16), versus the total number N of samples in the MCMC chain, which is varied from 2^{10} to 2^{23} . We see that both MH methods (i.e. TT-MH and DRAM) converge with a rate of $N^{-1/2}$, as expected. To keep the set-up cost of the TT approximation low, we only consider fairly crude TT approximations (as in Table 2). However, all our approximations deliver a smaller sampling error for TT-MH than for DRAM when measured against the number of samples and an even greater reduction when plotted against CPU time (Fig. 3, right). More accurate TT approximations require more evaluations of π during the set-up in TT-cross, up to $2.5 \cdot 10^5$ for $\delta = 0.05$ and $n = 32$. This set-up cost is clearly visible in the vertical offset of the curves in Fig. 3 (right). Notice that only $N/4$ initial DRAM samples are discarded in Fig. 3, which is much smaller than the actual burn-in for $N < 10^5$. The reason for this is that we want to compare the methods based on their best possible performance for the desired quadrature error \mathcal{E}_q , regardless of the actual distribution the samples belong to for small N . Therefore, DRAM has zero set-up cost in Fig. 3 (right) in the limit of $N \rightarrow 1$. In spite of that, TT-MH is much faster than DRAM for the same number of evaluations, which yields a significant difference in terms of the total CPU time.

There are several reasons for this. For higher TT accuracies, the gains are mainly due to the significantly lower IACT of TT-MH, leading to a much better statistical efficiency of the MCMC chain. But also for low TT accuracies, the IACT of the TT-MH algorithm is still a fraction of that for DRAM. A further reason is the vectorization that is exploited in TT-cross, where a block of $\mathcal{O}(nr^2)$ samples is evaluated in each step. In DRAM, the function needs to be evaluated point by point in order to perform the rejection. Therefore, the num-

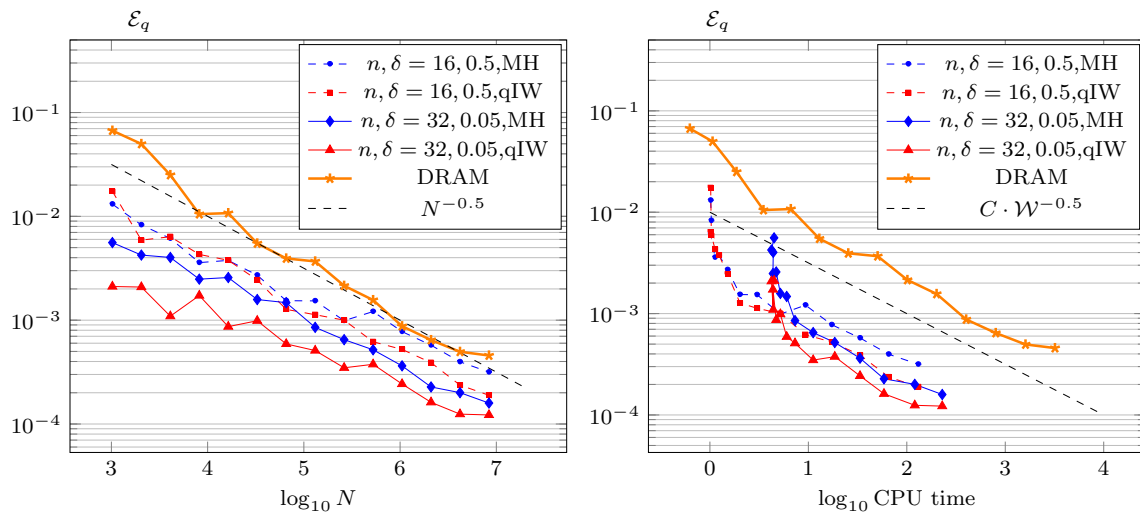


Fig. 3 Shock absorber example ($D = 6$): sampling error versus chain length (left) and versus total CPU time (right) for different choices of n and δ in the TT-cross method

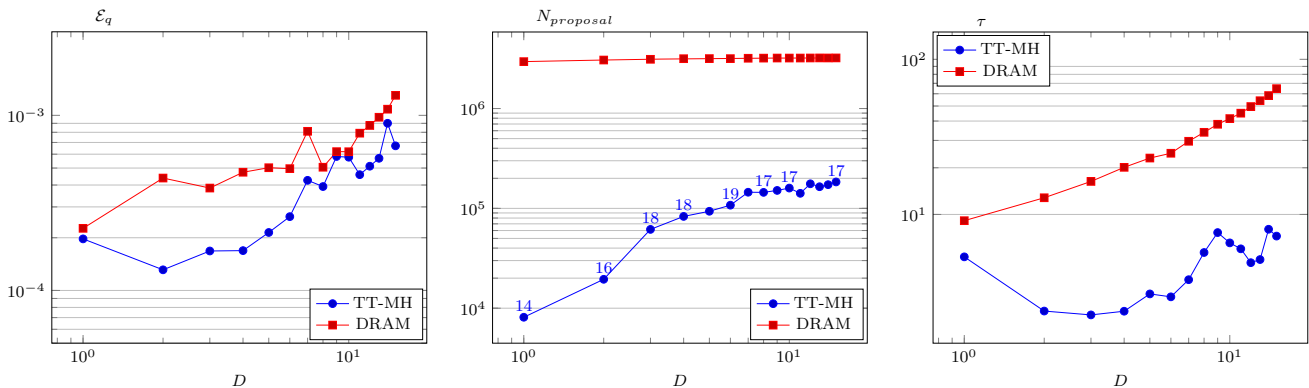


Fig. 4 Shock absorber example: error (left), number of π evaluations during the proposal stage (middle) and IACT (right), for different numbers of covariates and $n = 16, \delta = 0.05, N = 2^{22}$. Numbers above points in the middle plot denote TT ranks

ber of *distinct* calls to π in TT-cross is much smaller than N , reducing the corresponding overhead in MATLAB. In compiled languages (C, Fortran) on a single CPU, the difference may be less significant. However, parallel implementations will also benefit from the blocking, especially when each sample is expensive. More accurate TT approximations are worthwhile to compute if a highly accurate estimate of the expected value is required, since in that case the length of the MCMC chain will dominate the number of samples in the set-up phase.

In Fig. 3, we also present results with the TT-qIW approach described in Sec. 4.3, where the approximate density π^* is used as an importance weight and where the expected value and the normalizing constant are estimated via QMC quadrature. In particular, we use a randomized rank-1 lattice rule with product weight parameters $\gamma_k = 1/k^2$. The generating vector was taken from the file `lattice-39102-1024-1048576.3600`, available at

<http://web.maths.unsw.edu.au/~fkuo/>. Due to the non-smooth dependence of quantiles on the covariates, the rate of convergence for TT-qIW with respect to N is not improved in this example, but in absolute terms it consistently outperforms TT-MH, leading to even bigger gains over DRAM.

Finally, we fix the TT and the MCMC parameters to $n = 16, \delta = 0.05$ and $N = 2^{22}$ and vary the number of covariates D , and hence, the total dimensionality $d = D + 2$. In Fig. 4, we show the error in the quantiles, the number of evaluations of π , as well as the autocorrelation times and TT ranks. We see that the TT ranks are almost independent of d , and the TT-MH approach remains more efficient than DRAM over a wide range of dimensions.

5.2 Rosenbrock function

As a benchmark example with particularly long tails (and hence potentially large autocorrelation times in MCMC),

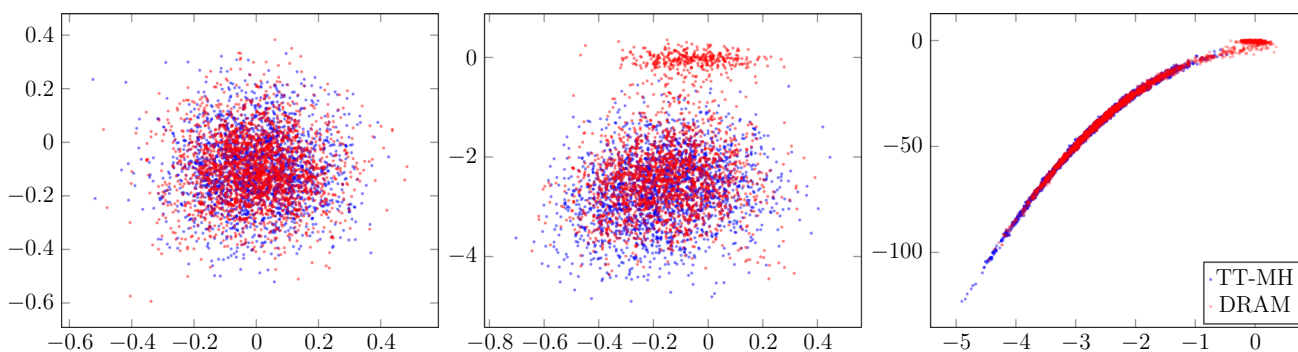


Fig. 5 Rosenbrock function ($d = 32$): $N = 2^{17}$ samples projected to the (θ_1, θ_2) (left), the $(\theta_{30}, \theta_{31})$ (middle) and the $(\theta_{31}, \theta_{32})$ -plane (right); TT-MH (blue) and DRAM (red). (Color figure online)

Table 3 Rosenbrock function example: IACT

d	2	4	8	16	32
TT-MH	1.096	1.080	1.100	1.079	1.084
DRAM	61.54	26.63	45.01	84.02	169.57

we consider the PDF induced by the Rosenbrock function $\pi(\theta) \propto \exp(-\frac{1}{2}r(\theta))$, where

$$r(\theta) = \sum_{k=1}^{d-1} \left[\theta_k^2 + \left(\theta_{k+1} + 5 \cdot (\theta_k^2 + 1) \right)^2 \right]. \tag{20}$$

The dimension d can be increased arbitrarily. The parameters for the TT approximation are chosen to be $\delta = 3 \cdot 10^{-3}$ and $n = 128$ for $\theta_1, \dots, \theta_{d-2}$, $n = 512$ for θ_{d-1} and $n = 4096$ for θ_d . Each θ_k is restricted to a finite interval $[-a_k, a_k]$, where $a_d = 200$, $a_{d-1} = 7$ and $a_k = 2$ otherwise.

Figure 5 shows certain projections of $N = 2^{17}$ sampling points produced with TT-MH and DRAM for $d = 32$. We see that although the density function is reasonably compact and isotropic in the first variables, it is highly concentrated in the last variable. DRAM requires a significant number of burn-in iterations, which are shown in Fig. 5 (middle and right) as the red cloud of samples that are not overlapped by blue ones. In order to eliminate the burn-in in DRAM, we compute 2^{20} samples and discard the first quarter of the chain. The difference is even more significant if we look at the integrated autocorrelation times in Table 3. For TT-MH, the IACT stays close to 1 for all considered dimensions, while for DRAM it exceeds 100 for larger d .

5.3 Inverse diffusion problem

Finally, we use our new TT-CD sampler to explore the posterior distribution arising from a Bayesian formulation of an

infinite-dimensional inverse problem, as formalized in Stuart (2010).

Let X and V be two infinite-dimensional function spaces—it is sufficient to consider separable Banach spaces—and let $\mathcal{G} : X \rightarrow V$ be a (measurable and well posed) forward map. Consider the inverse problem of finding $\kappa \in X$, an input to \mathcal{G} , given some noisy observations $y \in \mathbb{R}^{m_0}$ of some functionals of the output $u \in V$. In particular, we assume a (measurable) observation operator $Q : V \rightarrow \mathbb{R}^{m_0}$, such that

$$y = Q(\mathcal{G}(\kappa)) + \eta,$$

where $\eta \in \mathbb{R}^{m_0}$ is a zero-mean random variable that denotes the observational noise. The inverse problem is clearly under-determined when $m_0 \ll \dim(X)$, and in most mathematical models, the inverse of the map \mathcal{G} is ill-posed.

We do not consider prior modelling in any detail and present here a stylized Bayesian formulation designed to highlight the computational structure and cost. We simply state a prior measure μ_0 , to model κ in the absence of observations y . The posterior distribution μ^y over $\kappa|y$, the unknown coefficients conditioned on observed data, is given by Bayes’ theorem for general measure spaces,

$$\frac{d\mu^y}{d\mu_0}(u) = \frac{1}{Z} L(\kappa), \tag{21}$$

where the left-hand side is the Radon–Nikodym derivative, L is the likelihood function and Z is the normalizing constant (Stuart 2010).

For computing, we have to work with a finite-dimensional approximation $\kappa_d \in X_d \subset X$ of the latent field κ such that $\dim(X_d) = d \in \mathbb{N}$ and define κ_d as a deterministic function of a d -dimensional parameter $\theta := (\theta_1, \dots, \theta_d)$. Typically, we require that $\kappa_d \rightarrow \kappa$ as $d \rightarrow \infty$, but we will not focus on that convergence here and instead fix $d \gg 1$. To be able to apply the TT representation, we set $\theta_k \in [a_k, b_k]$ with $a_k < b_k$, for all $k = 1, \dots, d$, and then, κ_d maps the tensor

product domain $\Gamma_d := \prod_{k=1}^d [a_k, b_k]$ to X_d . We denote by $\pi_0(\theta)$ and $\pi(\theta) = \pi^y(\theta)$ the probability density functions of the pullback measures of the prior and posterior measures μ_0 and μ^y under the map $\kappa_d : \Gamma_d \rightarrow X_d$, respectively, and specify that map so that $\pi_0(\theta) = 1/|\Gamma_d|$, i.e. the prior distribution over θ is uniform.

We can then compute TT approximations of the posterior density $\pi(\theta)$ as in the previous examples by using Bayes’ formula (21), i.e.

$$\pi(\theta) = \frac{1}{Z} L(\kappa_d(\theta)), \quad \text{where } Z = \int_{\Gamma_d} L(\kappa_d(\theta)) d\pi_0(\theta).$$

Consider some quantity of interest in the form of another functional $F : V \rightarrow \mathbb{R}$ of the model output $\mathcal{G}(\kappa_d)$. The posterior expectation of F , conditioned on measured y , can be computed as

$$\mathbb{E}_\pi [F(\mathcal{G}(\kappa_d))] = \frac{\mathbb{E}_{\pi_0} [L(\kappa_d) F(\mathcal{G}(\kappa_d))]}{\mathbb{E}_{\pi_0} [L(\kappa_d)]}. \tag{22}$$

5.3.1 Stylized elliptic problem and parametrization

As an example, we consider the forward map defined by the stochastic diffusion equation

$$-\nabla \cdot (\kappa_d(\theta) \nabla u) = 0 \quad \text{on } D := (0, 1)^2, \tag{23}$$

with Dirichlet boundary conditions $u|_{x_1=0} = 1$ and $u|_{x_1=1} = 0$, as well as homogeneous Neumann conditions otherwise (Scheichl et al. 2017), which depends on an unknown (parametrized) diffusion coefficient $\kappa_d \in X_d \subset L_\infty(D)$.

For this example, we take each of the parameters θ_k , $k = 1, \dots, d$, to be uniformly distributed on $[-\sqrt{3}, \sqrt{3}]$. Then, for any $\theta \in \Gamma_d$ and $x = (x_1, x_2) \in D$, the logarithm of the diffusion coefficient at x is defined by the following expansion:

$$\begin{aligned} \ln \kappa_d(\theta, x) &= \sum_{k=1}^d \theta_k \sqrt{\eta_k} \cos(2\pi \rho_1(k)x_1) \cos(2\pi \rho_2(k)x_2), \\ \rho_1(k) &= k - \tau(k) \frac{(\tau(k) + 1)}{2}, \quad \rho_2(k) = \tau(k) - \rho_1(k), \\ \tau(k) &= \left\lfloor -\frac{1}{2} + \sqrt{\frac{1}{4} + 2k} \right\rfloor \quad \text{and} \\ \eta_k &= k^{-(\nu+1)} / K, \quad K = \sum_{k=1}^d k^{-(\nu+1)}. \end{aligned} \tag{24}$$

The expansion is similar to the one proposed in Eigel et al. (2014) and mimics the asymptotic behaviour of the Karhunen–Loève expansion of random fields with Matérn covariance function and smoothness parameter ν in two

dimensions, in which the norms of the individual terms decay algebraically with the same rate. However, realizations do not have the same qualitative features and we use it purely to demonstrate the computational efficiency of our new TT samplers.

To discretize the partial differential equation (PDE) in (23), we tessellate the spatial domain D with a uniform rectangular grid T_h with mesh size h . Then, we approximate the exact solution $u \in V := H^1(D)$ that satisfies the Dirichlet boundary conditions with the continuous, piecewise bilinear finite element (FE) approximation $u_h \in V_h$ associated with T_h . To find u_h , we solve the resulting Galerkin system using a sparse direct solver.

For this example, we take the observations to be m_0 noisy local averages of the PDE solution over some subsets $D_i \subset D$, $i = 1, \dots, m_0$, i.e.

$$Q_i(\mathcal{G}(\theta)) = \frac{1}{|D_i|} \int_{D_i} u_h(x, \theta) dx, \quad i = 1, \dots, m_0.$$

We take observation noise to be additive, distributed as i.i.d. zero-mean Gaussian noise with variance σ_n^2 , giving the likelihood function,

$$L(\theta) = \exp \left(-\frac{|Q(\mathcal{G}(\theta)) - y|^2}{2\sigma_n^2} \right),$$

and posterior distribution $\pi(\theta) = \frac{1}{Z} L(\theta)$, with the normalization constant $Z = \int_{[-\sqrt{3}, \sqrt{3}]^d} L(\theta) d\theta$.

In our experiments, the sets D_i are square domains with side length $2/(\sqrt{m_0} + 1)$, centred at the interior vertices of a uniform Cartesian grid on $D = [0, 1]^2$ with grid size $1/(\sqrt{m_0} + 1)$, that form an overlapping partition of D . We consider an academic problem with synthetic data for these m_0 local averages from some ‘true’ value θ_* . In particular, we evaluate the observation operator at $\theta_* = (\theta_0, \dots, \theta_0)$, for some fixed $0 \neq \theta_0 \in (-\sqrt{3}, \sqrt{3})$, and synthesize data by then adding independent normally distributed noise $\eta_* \sim \mathcal{N}(0, \sigma_n^2 I)$, such that $y = Q(\mathcal{G}(\theta_*)) + \eta_*$.

We consider two quantities of interest. The first is the average flux at $x^1 = 1$. This can be computed as Teckentrup et al. (2013)

$$\begin{aligned} F(\mathcal{G}(\theta)) &= - \int_0^1 \int_0^1 \kappa_d(x, \theta) \nabla w_h(x) \nabla u_h(x, \theta) dx \\ &= - \int_0^1 \kappa_d(x, \theta) \frac{\partial u_h(x, \theta)}{\partial \mathbf{n}} \Big|_{x^1=1} dx^2, \end{aligned} \tag{25}$$

where $w_h \in V_h$ is any FE function that satisfies the Dirichlet conditions at $x^1 = 0$ and $x^1 = 1$. This formula for the average flux is a smooth function of θ , which ensures a fast convergence for QMC-based quadrature rules, with an order close to N^{-1} . However, we also consider the discontinuous

Table 4 Default model and discretization parameters for the inverse diffusion example

ν	σ_n^2	θ_0	m_0	h	d	δ	n
2	0.01	1.5	9	2^{-6}	11	0.1	32

indicator function $\mathbb{I}_{F(\theta) > 1.5}$, to estimate the probability that the average flux in (25) becomes larger than 1.5, i.e.

$$P_{F > 1.5} = \text{Prob}(F(\mathcal{G}(\theta)) > 1.5) = \mathbb{E}_\pi [\mathbb{I}_{F(\theta) > 1.5}].$$

As we shall see, the non-smoothness of $\mathbb{I}_{F(\theta) > 1.5}$ reduces the order of convergence of the QMC quadrature to the basic Monte Carlo rate $N^{-1/2}$. For the same reason, this function lacks a low-rank TT decomposition, and hence, we cannot compute its expectation using a tensor product quadrature directly. The mean field flux $F|_{\theta=0} = 1$ (in the units used) and the probability $P_{F > 1.5}$ are both of the order of 0.1.

The default parameters used in the stochastic model and for function approximation are shown in Table 4. We will make it clear when we change any of those default parameters.

The TT approximation $\tilde{\pi}$ can be computed directly by the TT-cross algorithm, as in the previous examples. For a TT tolerance of $\delta = 0.1$, this requires about $10^4 - 10^5$ evaluations of π . However, since here the computation of each value of $\pi(\theta)$ involves the numerical solution of the PDE (23) this leads to a significant set-up time. This set-up time can be hugely reduced, by first building a TT approximation $\tilde{u}_h(\cdot, \theta)$ of the FE solution $u_h(\cdot, \theta)$ and then using $\tilde{u}_h(\cdot, \theta)$ in the TT-cross algorithm for building $\tilde{\pi}$ instead of $u_h(\cdot, \theta)$.

It was shown in Dolgov and Scheichl (2019) that a highly accurate approximation of $u_h(\cdot, \theta)$ in the TT format can be computed using a variant of the TT-cross algorithm, the alternating least-squares cross (ALS-cross) algorithm, that only requires $\mathcal{O}(r)$ PDE solves, if the TT ranks to approximate $u_h(\cdot, \theta)$ up to the discretization error are bounded by r . Moreover, the rank grows only logarithmically with the required accuracy. We will see, below, that $r < 100$ for this model problem for $h = 2^{-6}$, significantly reducing the number of PDE solves required in the set-up phase.

Since the observation operator Q consists of integrals of the PDE solution over subdomains of the spatial domain D , when applied to a function given in TT format it can be evaluated at a cost that is smaller than r PDE solves on T_h without any increase in the TT rank (Dolgov and Scheichl 2019). Finally, to compute an approximation of π via the TT-cross algorithm we use the significantly cheaper TT surrogate $Q(\tilde{u}_h(\cdot, \theta))$ in each evaluation of $\pi(\theta)$ instead of computing the actual FE solution $u_h(\cdot, \theta)$. Since $\tilde{u}_h(\cdot, \theta)$ is accurate up to the FE discretization error in V_h —which in this model problem for $h = 2^{-6}$ is of $\mathcal{O}(10^{-4})$ —this has essentially no

impact on the accuracy of the resulting TT approximation $\tilde{\pi}$ (especially for TT accuracy $\delta = 0.1$).

5.3.2 Set-up cost and accuracy of TT approximation

As in the shock absorber example, we test how the quality of the Markov chain produced by TT-MH depends on the error between $\tilde{\pi}$ and π . In Fig. 6 (left), we show the rejection rates, IACT and error estimates (18), (19) for different stopping tolerances δ and grid sizes n . In the top plot, we fix $\delta = 10^{-3}$ and vary n , while in the bottom plot, n is fixed to 512 and δ is varied. The other model parameters are set according to Table 4, and the chain length is $N = 2^{16}$. The behaviour is as in the shock absorber example and as predicted in Lemma 1.

In Fig. 6 (right), we demonstrate the benefit of first computing a TT approximation $\tilde{u}_h(\cdot, \theta)$ of the FE solution $u_h(\cdot, \theta)$, as described in the previous subsection. We see that the TT ranks to approximate u_h are significantly smaller than the TT ranks to approximate the density π to the same accuracy. In both cases, the TT ranks are independent of n , for n sufficiently large, and they show only a logarithmic dependence on the TT accuracy δ . However, the growth is significantly faster for π^* than for \tilde{u}_h . For the default parameters in Table 4, the ranks of $\tilde{u}_h(\cdot, \theta)$ and $\pi^*(\theta)$ are 26 and 82, respectively, and the numbers of function evaluations to build the TT surrogates are about 100 and about 53,000, respectively. The advantage is that with the surrogate \tilde{u}_h available those 53,000 evaluations of π , using \tilde{u}_h instead of u_h in the data misfit functional, are significantly cheaper and the PDE only has to be actually solved 100 times.

5.3.3 Convergence of the expected quantities of interest

In this section, we investigate the convergence of estimates of the expected value of the quantities of interest and the computational complexity of the different methods. For the TT approximation of the density function π , we fix $n = 32$ and $\delta = 0.1$. For the TT approximation of u_h , we choose a TT tolerance of 10^{-4} , which is equal to the discretization error for $h = 2^{-6}$.

To compute the posterior expectations of the QoIs in (22), we compare two approaches that use our TT-CD sampling procedure:

- [TT-MH] (Sec. 4.2) Metropolis–Hastings with independence proposals sampled via the TT-CD sampling procedure from the approximate distribution π^* .
- [TT-qIW] (Sec. 4.3) Using the approximate density π^* as an importance weight and estimating the expected value and the normalizing constant via a randomized QMC lattice rule.

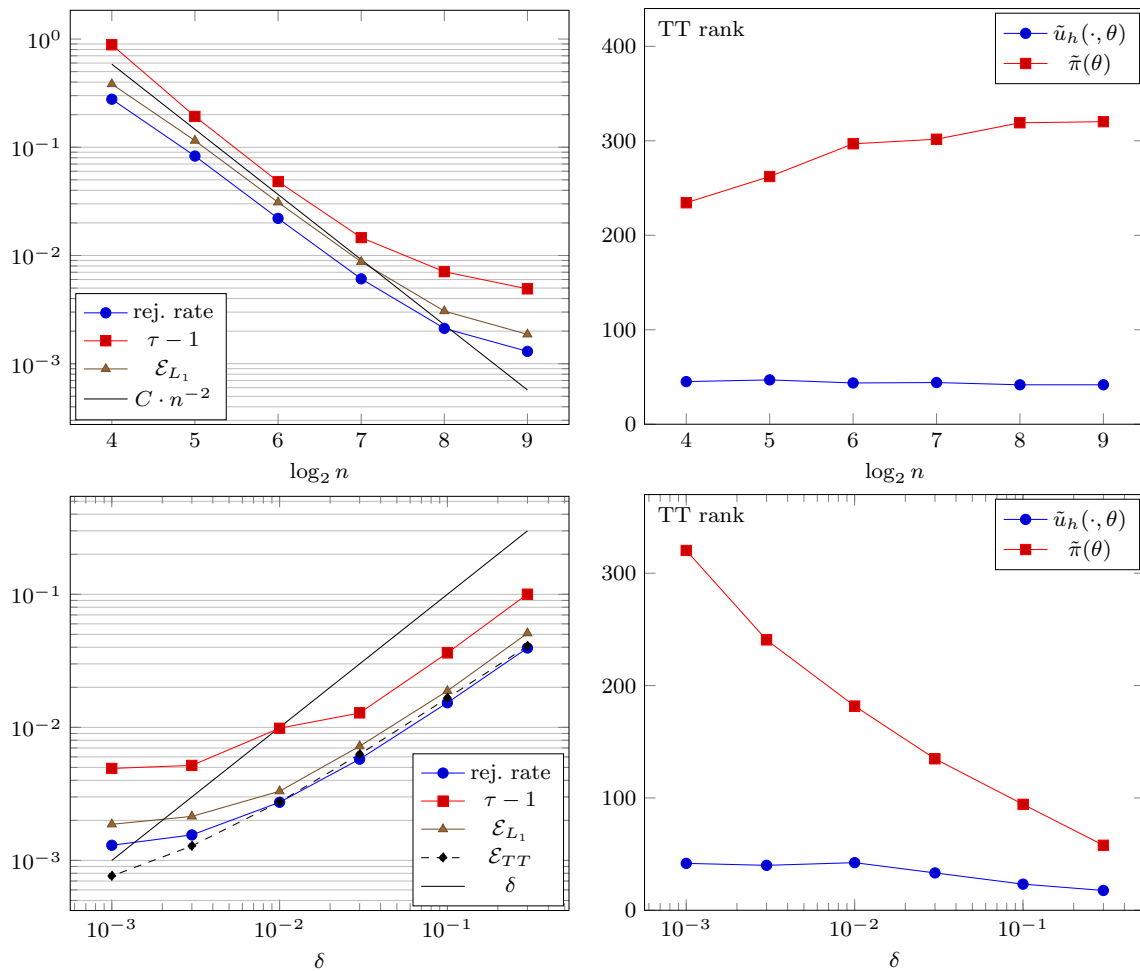


Fig. 6 Inverse diffusion problem: rejection rate, IACT and errors (left), as well as maximal TT ranks for $\tilde{u}_h(\cdot, \theta)$ and for $\tilde{\pi}(\theta)$ (right) for different grid sizes n (top) and values of δ (bottom)

Moreover, we test the two-level versions of both methods described in Sect. 4.4.

To benchmark the TT approaches, we use again DRAM with the initial covariance chosen to be the identity and discard the first $N/4$ samples. However, as a second benchmark, we also compute the posterior expectation directly by applying QMC to the two terms in the ratio estimate (QMC-rat), as defined in (22) and analysed in Scheichl et al. (2017). The QMC method in TT-qIW is again the randomized rank-1 lattice rule with product weights $\gamma_k = 1/k^2$ and generating vector from the file lattice-39102-1024-1048576.3600 at <http://web.maths.unsw.edu.au/~fkuo/>. In order to reduce random fluctuations in the results, we average 16 runs of each approach in each experiment. The rejection rate and the IACT for TT-MH and DRAM are shown in Table 5. Notice that the autocorrelation times of DRAM for the coordinates θ and for the quantity of interest F differ significantly, since the latter coordinates have a weaker influence on F .

Table 5 Inverse diffusion problem: rejection rates and IACTs for TT-MH and DRAM; τ_θ and τ_F are the IACT for the parameter vector θ and for the QoI in (25), respectively

	Rejection rate	τ_θ	τ_F
TT-MH	0.0853	1.1964	1.1903
DRAM	0.73	84.0	29.7

In Fig. 7, we present the relative errors in the quantities of interest versus the chain length N together with reference slopes. For the expected value $\mathbb{E}_\pi[F]$ of the flux in Fig. 7 (left), the QMC ratio estimator (QMC-rat) converges with a rate close to linear in $1/N$, so that it becomes competitive with the TT approaches for higher accuracies. However, by far the most effective approach is TT-qIW, where the TT approximation π^* is used as an importance weight in a QMC ratio estimator. Asymptotically, the convergence rate for TT-qIW is also $\mathcal{O}(N^{-1})$ for $\mathbb{E}_\pi[F]$ and the effectivity of the estimator is almost two orders of magnitude better than that

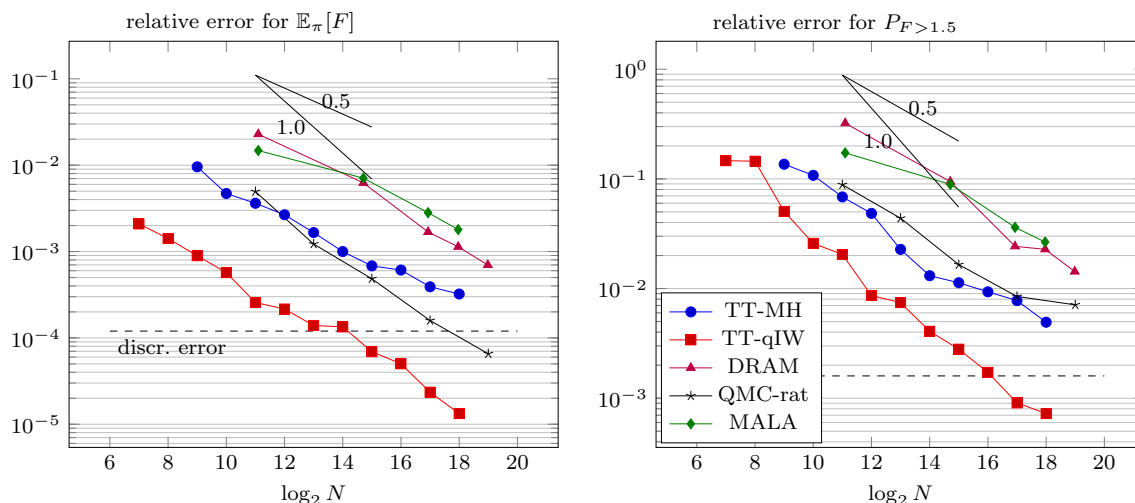


Fig. 7 Inverse diffusion problem: relative errors for the average flux (left) and for the probability of the flux exceeding 1.5 (right) for different numbers of samples N

of DRAM. All the other TT-based approaches and DRAM converge, as expected, with the standard MC order $N^{-1/2}$. For the non-smooth indicator function employed in $P_{F>1.5}$ in Fig. 7 (right), the relative performance of the different approaches is similar, although the QMC-rat estimator now also converges with the MC rate of order $\mathcal{O}(N^{-1/2})$. Somewhat surprisingly, the TT-qIW method seems to converge slightly better than $\mathcal{O}(N^{-1/2})$ also for $P_{F>1.5}$ and outperforms all other approaches by an order of magnitude.

The results in Fig. 7 are all computed for the same spatial resolution of the forward model. In a practical inverse problem, for the best efficiency, all errors (due to truncation, discretization and sampling) are typically equilibrated. Thus, it is useful to estimate the spatial discretization error. We achieve this by computing the posterior expectations of the QoIs on three discretization grids (with TT-qIW and $N = 2^{18}$) and by using these to estimate the error via Runge’s rule. The estimated error for $h = 2^{-6}$ is plotted as a horizontal dashed line in Fig. 7. We see that with the TT-qIW method $N = 2^{13}$ samples are sufficient to obtain a sampling error of the order of the discretization error for $\mathbb{E}_\pi[F]$, while all other approaches require at least $N = 2^{17}$ samples (up to $N > 2^{21}$ for DRAM).

In addition to DRAM, we also consider a version of the Metropolis-adjusted Langevin (MALA) algorithm with adapted empirical covariance matrix as a preconditioner (Atchadé 2006). However, the latter components of the gradient are rather small and give little information about the geometry. This makes the MALA convergence comparable to that of DRAM. Moreover, each step of MALA is about twice more expensive than a step of DRAM due to the computation of the gradient of $u_h(\cdot, \theta)$ in $\nabla \log \pi(\theta)$ in addition to u_h .

In Fig. 8, we compare the approaches in terms of total CPU time. The horizontal offset for all the TT-based methods is the time needed to build the TT approximation $\tilde{\pi}$. The error then initially drops rapidly. As soon as the number N of samples is big enough, the set-up cost becomes negligible and the relative performance of all the approaches is very similar to that in Fig. 7, since the computational time per sample is dominated by the PDE solve and all approaches that we are comparing evaluate π for each sample. It is possible to significantly reduce this sampling cost, if we do not evaluate the exact π for each sample, e.g. by simply computing the expected value of the QoIs with respect to the approximate density π^* using TT-CD and QMC quadrature. However, in that case the estimator will be biased and the amount of bias depends on the accuracy of the TT surrogate π^* . In that case, the total cost is dominated by the set-up cost. (A more detailed study of the cost of the various stages of our TT approach is included in Fig. 11.)

In Fig. 9, we include a more detailed study of the influence of the TT parameters n and δ . As expected, a more accurate TT surrogate provides a better proposal/importance weight and thus leads to a better performance, but it also leads to a higher set-up cost. So for lower accuracies, cruder approximations are better. However, the quality of the surrogate seems to be less important for Monte Carlo-based approaches. For the middle plot in Fig. 9, we used the importance weighting method described in Sec. 4.3 with random Monte Carlo samples (TT-rIW). It converges with almost the same rate as TT-MH, which might be due to independence proposals. The quality of the surrogate seems to be significantly more important for the QMC-based approaches, such as for TT-qIW (Fig. 9, right), since the mapped QMC samples carry the PDF approximation error.

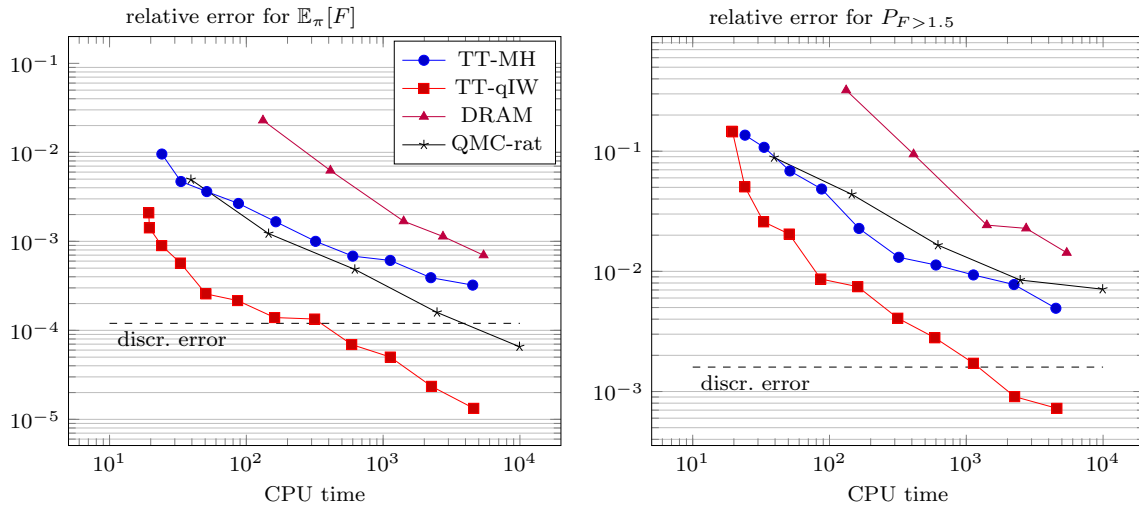


Fig. 8 Inverse diffusion problem: relative errors in the mean flux (left) and in the exceedance probability (right) plotted against the total CPU times (s)

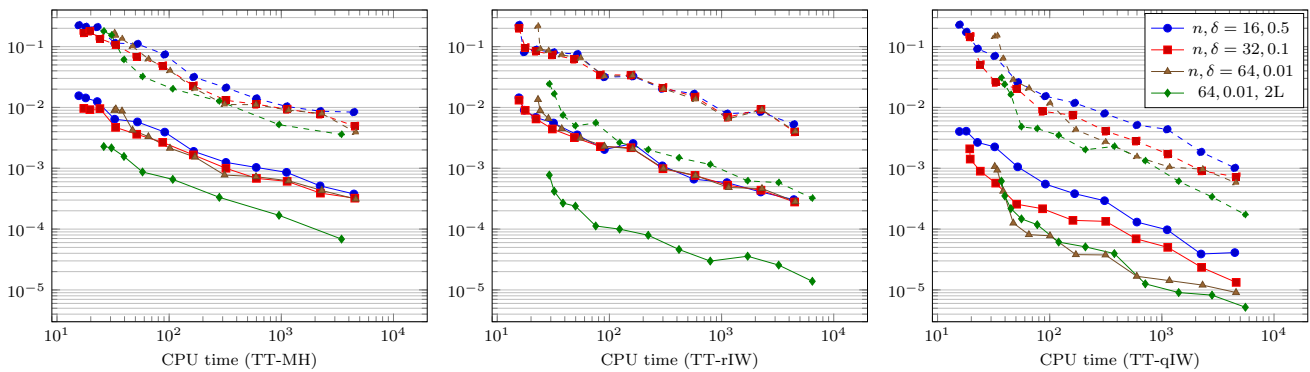


Fig. 9 Inverse diffusion problem: convergence of $\mathbb{E}_\pi[F]$ (solid lines) and $P_{F>1.5}$ (dashed lines) with TT-MH (left), TT-rIW (middle) and TT-qIW (right) for different choices of n and δ

Another thing we study in Fig. 9 is the two-level versions of TT-MH and of importance weighting described in Sect. 4.4. While the variance reduction and the induced cost reduction are significant compared to the single-level quadrature in the case of i.i.d. seed points in Algorithm 2 (in both TT-MH and TT-rIW), the difference in the case of QMC seeds in TT-qIW is marginal. This is because the rate of convergence of the QMC quadrature drops to $\mathcal{O}(N^{-1/2})$ when applied to the less smooth difference term in (15). In contrast, the single-level QMC estimator (10) converges with a noticeably higher rate.

We also benchmark the algorithms in a more challenging scenario of a smaller noise variance $\sigma_n^2 = 10^{-3}$. Due to non-linearity of the forward model, the posterior density function is concentrated along a complicated high-dimensional manifold, for smaller σ_n . This increases all complexity indicators: the ranks of the TT approximation, the IACT in TT-MH and in DRAM and the variances in the ratio estimators. Since the density function is more concentrated, we choose finer

parameters $n = 64$ and $\delta = 0.03$ for the TT approximation. Nevertheless, in Fig. 10 we see that even though the set-up cost is larger, the TT-based samplers are still all significantly more efficient than DRAM. Due to the stronger concentration of π , the performance of the basic ratio estimator QMC-rat is worse. On the other hand, the QMC estimator TT-qIW with TT importance weighting is again the most performant method. Note that it is the only method that reduces the quadrature error to the size of the discretization error within the considered limit of one million samples.

Finally, we profile the computational cost of all the various components in the TT approaches with respect to the total error (truncation, spatial discretization and quadrature). We vary the spatial mesh size h from 2^{-5} to 2^{-7} and estimate the convergence rate of the discretization error (Fig. 11, left). Then, we choose the other approximation parameters in order to equilibrate the errors. In particular, the number of random variables d and the number of samples N are chosen such that the KL truncation error in (24) and the quadrature error

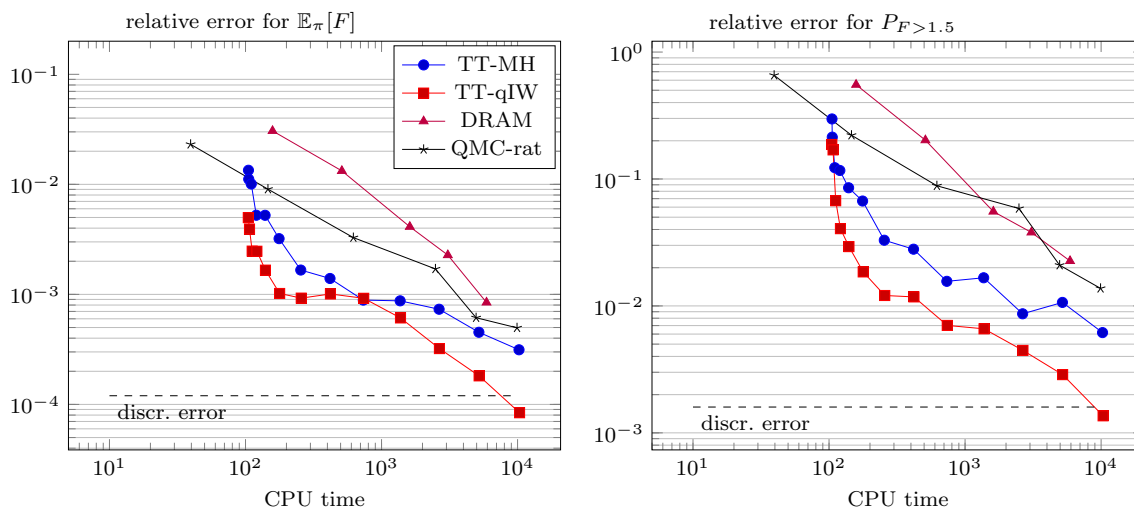


Fig. 10 Inverse diffusion problem: relative errors in the mean flux (left) and in the exceedance probability (right) plotted against the total CPU times (s) for $\sigma_n^2 = 10^{-3}$

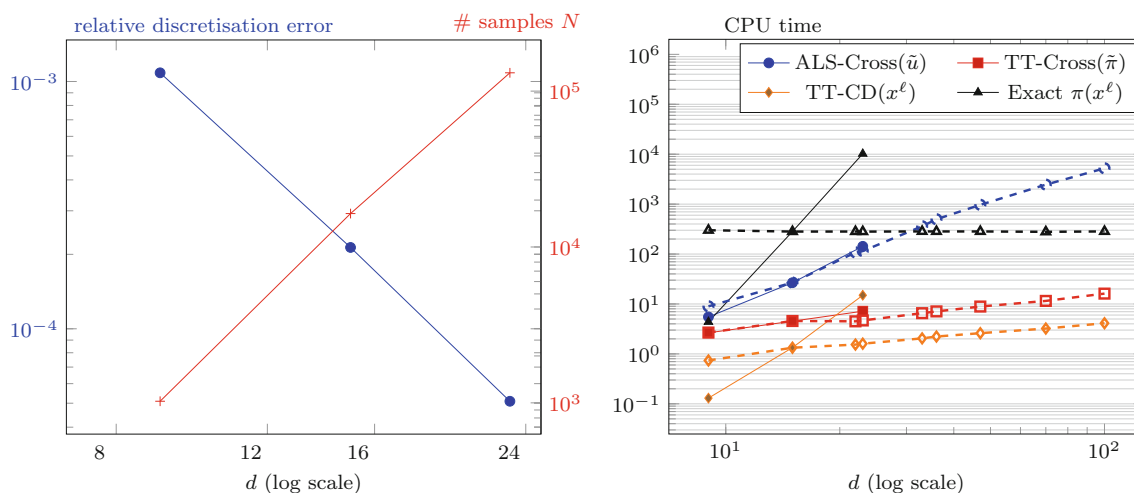


Fig. 11 Inverse diffusion problem: dimension (d) dependence of discretization error and numbers of samples (left) and CPU times of the various algorithmic components in TT-qIW (right); solid lines with equilibrated errors, dashed lines with $h = 2^{-6}$ and $N = 2^{14}$ fixed

of the TT-qIW method are equal to the discretization error, respectively (see Fig. 11, left).

The solid lines in Fig. 11 (right) give the computational times necessary for the various components of our algorithm (with all errors equilibrated), as a function of d (and thus also as a function of h^{-1} and N): the ALS-cross algorithm to build the TT surrogate of u_h , the TT-cross algorithm to build the TT surrogate of π , the TT-CD sampling procedure for the N samples x^ℓ , $\ell = 1, \dots, N$, and the evaluation of π at the N samples. Clearly the N PDE solves in the evaluation of π are the dominant part and the complexity of these evaluations grows fairly rapidly due to the spatial mesh refinement and the increase in N . The TT-cross algorithm for building $\tilde{\pi}$ (once a TT surrogate of the forward solution is available) and the cost of the TT-CD sampler depend on the dimension

d and on the TT ranks of $\tilde{\pi}$ (which grow very mildly with d and h^{-1}).

In addition, we also ran all the experiments with $h = 2^{-6}$ and $N = 2^{14}$ fixed, varying only d to explicitly see the growth with d . The timings for these experiments are plotted using dashed lines. The cost for the ALS-cross algorithm to build \tilde{u}_h grows cubically in d , while the cost to build the TT surrogate $\tilde{\pi}$ and the cost of the TT-CD sampling procedure grow linearly with d . Since the evaluation of π is dominated by the cost of the PDE solve, its cost does not grow with dimension. This shows that the TT-CD sampler is an extremely effective surrogate for high dimensions when the model admits a natural extension in d (e.g. it converges as $d \rightarrow \infty$, or the variables remain locally correlated).

6 Conclusion

We presented a method for computational inference based on function approximation of the target PDF. That task has traditionally been viewed as infeasible for general multivariate distributions due to the exponential growth in cost for grid-based representations. The advent of the tensor train representation, amongst other hierarchical representations, is a significant development that circumvents that ‘curse of dimensionality’. Our main contributions here have been showing that the conditional distribution method can be implemented efficiently for PDFs represented in (interpolated) TT format and that quasi-Monte Carlo quadrature is both feasible and efficient with bias correction through a control-variate structure or via importance weighting. The latter scheme was most efficient across all computed examples and parameter choices.

We adapted existing tools for *tensors*, i.e. multidimensional arrays, in particular the TT-cross approximation scheme, and tools for basic linear algebra. We expect that substantial improvement could be achieved with algorithms tailored for the specific tasks required, such as function approximation, and the setting of coordinates and bounding region. Nevertheless, the algorithms presented are already very promising, providing sample-based inference that is more computationally efficient than a benchmark MCMC, the DRAM MCMC. We demonstrated the algorithms in three stylized examples: a time-to-failure model; an inverse problem; and sampling from a non-Gaussian PDF. Extensive computations showed that in each example the methods performed as theory predicts and that scaling with dimension is linear.

We view the methods developed here as a promising development in Markov chain Monte Carlo methods. It is noteworthy, however, that our most efficient algorithm (TT-qIW) neither implements a Markov chain for the basic sampler nor uses standard Monte Carlo quadrature. Instead, points from a randomized quasi-Monte Carlo (QMC) lattice are mapped into state space by the inverse Rosenblatt transform, implemented in the TT-CD algorithm, with unbiased estimates available via importance-weighted QMC quadrature. Nevertheless, the basic structure remains a proposal mechanism that is modified to produce a sequence of points that is ergodic for the target distribution.

Numerical experiments were carried out in MATLAB R2016b on an Intel Xeon E5-2650 CPU at the Balena High Performance Computing Service at the University of Bath, using one core per run. We implemented Algorithm 2 in MATLAB and C+Python, using the TT-Toolbox in MATLAB (Oseledets et al. 2011) and Python (available at <http://github.com/oseledets/ttpy>), respectively. The code is available at <http://github.com/dolgov/tt-irt>; we welcome suggestions or feedback from users.

Acknowledgements SD is grateful for the support from the Engineering and Physical Sciences Research Council (EPSRC) through Fellowship EP/M019004/1. This research was started, while CF was Global Chair in the Institute for Mathematical Innovation (IMI) at the University of Bath.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Atchadé, Y.F.: An adaptive version for the metropolis adjusted langevin algorithm with a truncated drift. *Methodol. Comput. Appl. Probab.* **8**(2), 235–254 (2006). <https://doi.org/10.1007/s11009-006-8550-0>
- Ballani, J., Grasedyck, L.: Hierarchical tensor approximation of output quantities of parameter-dependent PDEs. *SIAM/ASA J. Uncertain. Quantif.* **3**(1), 852–872 (2015)
- Brooks, S., Gelman, A., Jones, G., Meng, X.L. (eds.): *Handbook of Markov Chain Monte Carlo*. CRC Press, Boca Raton (2011)
- Christen, J., Fox, C.: A general purpose sampling algorithm for continuous distributions (the t-walk). *Bayesian Anal.* **5**(2), 263–282 (2010)
- Devroye, L.: *Non-Uniform Random Variate Generation*. Springer, Berlin (1986)
- Dick, J., Kuo, F., Sloan, I.: High-dimensional integration: the quasi-Monte Carlo way. *Acta Numer.* **22**, 133–288 (2013)
- Dodwell, T., Ketelsen, C., Scheichl, R., Teckentrup, A.: A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow. *SIAM/ASA J. Uncertain. Quantif.* **3**(1), 1075–1108 (2015)
- Dolgov, S., Savostyanov, D.: Alternating minimal energy methods for linear systems in higher dimensions. *SIAM J. Sci. Comput.* **36**(5), A2248–A2271 (2014)
- Dolgov, S., Scheichl, R.: A hybrid alternating least squares—TT cross algorithm for parametric PDEs. *SIAM/ASA J. Uncertain. Quantif.* **7**(1), 260–291 (2019)
- Eigel, M., Gittelsohn, C., Schwab, C., Zander, E.: Adaptive stochastic Galerkin FEM. *Comput. Methods Appl.* **270**, 247–269 (2014)
- Eigel, M., Marschall, M., Schneider, R.: Sampling-free Bayesian inversion with adaptive hierarchical tensor representations. *Inverse Probl.* **34**(3), 035010 (2018)
- Fox, C., Nicholls, G.: Sampling conductivity images via MCMC. In: *The Art and Science of Bayesian Image Analysis*, Leeds Annual Statistics Research Workshop, pp. 91–100 (1997)
- Fox, C., Norton, R.: Fast sampling in a linear-Gaussian inverse problem. *SIAM/ASA J. Uncertain. Quantif.* **4**(1), 1191–1218 (2016)
- Fox, C., Parker, A.: Accelerated Gibbs sampling of normal distributions using matrix splittings and polynomials. *Bernoulli* **23**(4B), 3711–3743 (2017)
- Fox, C., Haario, H., Christen, J.: Inverse problems. In: Damien, P., Dellaportas, P., Polson, N., Stephens, D. (eds.) *Bayesian Theory and Applications*, pp. 619–643. Oxford University Press, Oxford (2013)
- Gilks, W., Wild, P.: Adaptive rejection sampling for Gibbs sampling. *Appl. Stat.* **41**, 337–348 (1992)
- Golub, G.H., Van Loan, C.F.: *Matrix Computations*, 4th edn. Johns Hopkins University Press, Baltimore (2013)

- Goreinov, S., Tyrtshnikov, E., Zamarashkin, N.: A theory of pseudoskeleton approximations. *Linear Algebra Appl.* **261**(1–3), 1–21 (1997)
- Goreinov, S., Oseledets, I., Savostyanov, D., Tyrtshnikov, E., Zamarashkin, N.: How to find a good submatrix. In: Olshevsky, V., Tyrtshnikov, E. (eds.) *Matrix Methods: Theory, Algorithms, Applications*, pp. 247–256. World Scientific, Singapore (2010)
- Gutierrez-Pulido, H., Aguirre-Torres, V., Christen, J.: A practical method for obtaining prior distributions in reliability. *IEEE Trans. Reliab.* **54**(2), 262–269 (2005)
- Haario, H., Laine, M., Mira, A., Saksman, E.: DRAM: efficient adaptive MCMC. *Stat. Comput.* **16**(4), 339–354 (2006)
- Hägström, O., Rosenthal, J.: On variance conditions for Markov chain CLTs. *Electron. Commun. Probab.* **12**, 454–464 (2007)
- Hoang, V.H., Schwab, C., Stuart, A.M.: Complexity analysis of accelerated MCMC methods for Bayesian inversion. *Inverse Probl.* **29**(8), 085010 (2013)
- Hörmann, W., Leydold, J., Derflinger, G.: *Automatic Nonuniform Random Variate Generation*. Springer, Berlin (2004)
- Johnson, M.: *Multivariate Statistical Simulation*. Wiley, New York (1987)
- Khoromskij, B.: Structured rank- (r_1, \dots, r_d) decomposition of function-related operators in \mathbb{R}^d . *Comput. Methods Appl. Math.* **6**(2), 194–220 (2006)
- Kuo, F., Scheichl, R., Schwab, C., Sloan, I., Ullmann, E.: Multilevel quasi-Monte Carlo methods for lognormal diffusion problems. *Math. Comput.* **86**, 2827–2860 (2017)
- Liu, J.: Metropolized independent sampling with comparisons to rejection sampling and importance sampling. *Stat. Comput.* **6**(2), 113–119 (1996)
- Martino, L., Read, J., Luengo, D.: Independent doubly adaptive rejection Metropolis sampling within Gibbs sampling. *IEEE Trans. Signal Process.* **63**(12), 3123–3138 (2015)
- Meeker, W., Escobar, L.: *Statistical Methods for Reliability Data*. Wiley, New York (1998)
- Mengersen, K.L., Tweedie, R.L.: Rates of convergence of the Hastings and Metropolis algorithms. *Ann. Stat.* **24**(1), 101–121 (1996)
- Meyer, R., Cai, B., Perron, F.: Adaptive rejection Metropolis sampling using Lagrange interpolation polynomials of degree 2. *Comput. Stat. Data Anal.* **52**(7), 3408–3423 (2008)
- Mira, A.: Ordering and improving the performance of Monte Carlo Markov chains. *Stat. Sci.* **16**(4), 340–350 (2001)
- Mira, A., Geyer, C.J.: Ordering Monte Carlo Markov chains. *Tech. Rep. 632*, Univ. of Minnesota (1999)
- Niederreiter, H.: Quasi-Monte Carlo methods and pseudo-random numbers. *Bull. Am. Math. Soc.* **84**(6), 957–1041 (1978)
- Norton, R., Christen, J., Fox, C.: Sampling hyperparameters in hierarchical models: improving on Gibbs for high-dimensional latent fields and large datasets. *Commun. Stat. Simul.* **47**(9), 2639–2655 (2018)
- O’Connor, P., Kleyner, A.: *Practical Reliability Engineering*, 5th edn. Wiley, New York (2012)
- Oseledets, I.: DMRG approach to fast linear algebra in the TT-format. *Comput. Methods Appl. Math.* **11**(3), 382–393 (2011a)
- Oseledets, I.: Tensor-train decomposition. *SIAM J. Sci. Comput.* **33**(5), 2295–2317 (2011b)
- Oseledets, I.: Constructive representation of functions in low-rank tensor formats. *Constr. Approx.* **37**(1), 1–18 (2013)
- Oseledets, I., Tyrtshnikov, E.: TT-cross approximation for multidimensional arrays. *Linear Algebra Appl.* **432**(1), 70–88 (2010)
- Oseledets, I., Dolgov, S., Kazeev, V., Savostyanov, D., Lebedeva, O., Zhlobich, P., Mach, T., Song, L.: *TT-Toolbox* (2011). <https://github.com/oseledets/TT-Toolbox>
- Roberts, G.O., Rosenthal, J.S.: Quantitative non-geometric convergence bounds for independence samplers. *Methodol. Comput. Appl. Probab.* **13**(2), 391–403 (2011)
- Rosenblatt, M.: Remarks on a multivariate transformation. *Ann. Math. Stat.* **23**(3), 470–472 (1952)
- Rue, H.: Fast sampling of Gaussian Markov random fields. *J. R. Stat. Soc. B* **63**, 325–338 (2001)
- Rue, H., Held, L.: *Gaussian Markov Random Fields: Theory and Applications*. Chapman & Hall, London (2005)
- Scheichl, R., Stuart, A., Teckentrup, A.: Quasi-Monte Carlo and multi-level Monte Carlo methods for computing posterior expectations in elliptic inverse problems. *SIAM/ASA J. Uncertain. Quantif.* **5**(1), 493–518 (2017)
- Schneider, R., Uschmajew, A.: Approximation rates for the hierarchical tensor format in periodic Sobolev spaces. *J. Complex.* **30**(2), 56–71 (2013)
- Smith, R.L., Tierney, L.: Exact transition probabilities for the independence Metropolis sampler. *Tech. rep.*, Univ. of North Carolina (1996)
- Stuart, A.: Inverse problems: a Bayesian perspective. *Acta Numer.* **19**, 451–559 (2010)
- Teckentrup, A., Scheichl, R., Giles, M., Ullmann, E.: Further analysis of multilevel Monte Carlo methods for elliptic PDEs with random coefficients. *Numer. Math.* **125**(3), 569–600 (2013)
- Tierney, L.: A note on Metropolis-Hastings kernels for general state spaces. *Ann. Appl. Probab.* **8**(1), 1–9 (1998)
- Tyrtshnikov, E.: Tensor approximations of matrices generated by asymptotically smooth functions. *Sbornik Math* **194**(6), 941–954 (2003)
- Wolff, U.: Monte Carlo errors with less errors. *Comput. Phys. Commun.* **156**(2), 143–153 (2004)

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.