Statistics and Applications {ISSN 2454-7395 (online)}
Special Issue on Recent Advances in Bayesian Statistics
and Machine Learning
Volume 24, No. 2, 2026 (New Series), pp 79–124
http://www.ssca.org.in/journal

Function Optimization with Posterior Gaussian Derivative Process

Sucharita Roy[1] **and Sourabh Bhattacharya**[2]

[1]*Department of Mathematics*
St. Xavier's College (Autonomous), Kolkata
[2]*Interdisciplinary Statistical Research Unit, Indian Statistical Institute, Kolkata*

Received: 1 September 2025; Revised: 24 May 2026; Accepted: 11 July 2026

Abstract

Function optimization is fundamental to all scientific disciplines, yet no single method dominates. We propose a Bayesian optimization algorithm for functions with known first and second partial derivatives. The objective function is modeled as a Gaussian process, inducing a Gaussian derivative process. Given a small set of initial function evaluations, we obtain the posterior of the derivative process and construct a posterior for stationary points by setting the derivative to zero. A recursive importance-resampling scheme with prior constraints on gradient norm and Hessian definiteness drives the algorithm toward the true optima.

We provide a rigorous theoretical foundation, proving almost sure convergence of the algorithm to the true optima as the number of stages tends to infinity. This relies on novel results establishing almost sure uniform convergence of the Gaussian process and its derivative process posteriors to the true function and its derivatives under fixed-domain infill asymptotics; rates of convergence are also derived. In addition, we give a Bayesian characterization of the number of optima using the information accumulated during the algorithm.

Five optimization problems (maxima, minima, saddle points, inconclusive cases) from one to 100 dimensions illustrate the method. On a real-world Poisson regression (AIDS deaths data), our procedure achieves a substantially smaller gradient norm at the MLE than Fisher scoring, BFGS, simulated annealing, and multi-start quasi-Newton. The posterior simulation nature of the algorithm yields consistently more accurate results. C/MPI code related to the AIDS data is available at `https://github.com/Sourabh-Bhattacharya/FUNCTION_OPT_GDP`.

Key words: Dirichlet process; Fixed-domain infill asymptotics; Function optimization; Gaussian derivative process; Importance resampling; Transformation based MCMC.

AMS Subject Classifications: 62F15, 62G08, 65K10

Corresponding Author: Sourabh Bhattacharya
Email: sourabh@isical.ac.in

1. Introduction

Function optimization is a cornerstone of scientific computing, with applications across virtually every discipline. Despite decades of research, no single optimization method dominates all problem classes; for any sufficiently large family of problems, it is remarkably difficult to identify a methodology that consistently outperforms others in terms of theoretical foundation, accuracy, computational efficiency, or robustness. Consequently, practitioners often resort to heuristic methods tailored to the specific problem at hand.

Among stochastic optimization methods, simulated annealing is a general-purpose technique, but its practical success depends critically on carefully chosen temperature schedules and proposal distributions, both of which become extremely challenging to tune in moderate to high dimensions.

In this article, we propose and develop a novel Bayesian algorithm for optimizing functions whose first and second partial derivatives are available. Our method assumes that the objective function can be evaluated exactly (no observation noise) and that its first and second partial derivatives are available analytically or via automatic differentiation. Thus the setting is that of deterministic optimization, typical in computer experiments and engineering design, rather than stochastic optimization with noisy evaluations. Our approach embeds the objective function, together with its derivatives, into a random function framework driven by Gaussian processes and their induced derivative processes. With data comprising a set of input points and the corresponding function values, we first obtain the posterior derivative process. We then construct the posterior distribution of the solutions obtained by setting the random partial derivative functions to the null vector. This posterior is expected to emulate the stationary points of the objective function.

We further incorporate a uniform prior over the domain, constrained so that the first partial derivatives have small Euclidean norm and the Hessian matrix is positive definite (for minimization) or negative definite (for maximization). These constraints help the posterior solutions approximate the true optima even when the dataset is small. However, for very small datasets the prior may be too restrictive to allow effective posterior simulation. Therefore, we begin with a weaker restriction (for instance, bounding the gradient norm by a relatively large constant) to obtain an initial set of posterior solution simulations. Then, in successive iterative stages, we refine the prior restrictions (making them stricter), and at each stage we augment the dataset with new realizations (and their function values) that satisfy the current, tighter constraints. As the number of stages tends to infinity, the posterior distributions converge to the true optima. These ideas lead to a general, effective, and adaptive optimization algorithm.

The convergence of our algorithm hinges on the convergence of the posterior derivative process to the true derivatives of the objective function, which in turn depends on a suitable design of the input points. Under an appropriate fixed-domain infill asymptotics setup, we prove almost sure uniform convergence of the posterior Gaussian process and its derivative process to the objective function and its derivatives. Notably, we also obtain rates of convergence under a specific infill design. To the best of our knowledge, these results are new and of independent interest. Using them, we prove almost sure convergence of our optimization algorithm to the true optima as the number of stages grows. As an aside, we also provide a Bayesian characterization of the number of optima, exploiting information

accumulated by the algorithm.

We illustrate our method on five optimization problems that include finding maxima, minima, saddle points, and even inconclusive cases. The examples range from simple one-dimensional problems to challenging 50- and 100-dimensional problems. In each case we obtain encouraging and insightful results. Along the way we discuss various computational and accuracy issues. A key strength of our approach is its ability to achieve significantly more accurate solutions than existing optimization algorithms, thanks to the posterior simulation paradigm embedded in the method. Indeed, by exploring neighborhoods around any solution provided by other methods, our algorithm almost surely finds a point at least as close to the true optimum.

We note that existing Bayesian optimization methods using Gaussian processes typically treat the objective function as a black box and assume derivatives are unavailable (see, for example, Frazier (2018) and references therein). Those methods are naturally less accurate when derivatives are accessible. Moreover, we are not aware of any convergence results for such derivative-free Gaussian process methods, which rely on heuristic acquisition functions. Because many acquisition functions exist, each with its own advantages and drawbacks, those methods lack a solid theoretical foundation and may be unreliable in practice. In this article we do not further discuss those existing approaches.

The remainder of the paper is organized as follows. Section 2 provides the derivation of the posterior Gaussian derivative process. Section 3 derives the posterior distribution for the random optima obtained by setting the derivative process to zero, together with additional restrictions based on the objective function's derivatives. Section 4 presents almost sure uniform convergence results for the Gaussian process and its derivative process posteriors, along with proofs. Section 5 introduces our general-purpose Bayesian optimization algorithm. Section 6 establishes a Bayesian characterization of the number of optima. Section 7 illustrates the algorithm on various optimization problems. Section 8 summarizes our contributions and offers concluding remarks.

2. Posterior Gaussian derivative process

2.1. Details of the objective function

Consider any function $f : \mathbb{R}^d \mapsto \mathbb{R}$, where $\mathbb{R}$ is the real line and d (≥ 1) is the dimension of the input space, assumed to be finite. We further assume that the second order partial derivatives of f, namely, $\partial^2 f(\boldsymbol{x})/\partial x_i \partial x_j$ exist and are continuous for all $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ for $i, j = 1, \ldots, d$. The objective is to optimize the function $f(\boldsymbol{x})$ with respect to $\boldsymbol{x} \in \mathcal{X}$. For theoretical purposes, we assume that $\mathcal{X}$ is compact; this assumption is not required for implementation of our methodology.

2.2. Data from the objective function

Assume that corresponding to arbitrary inputs $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \in \mathcal{X}$, where $n > 1$, the output vector $\boldsymbol{f}_n = (f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n))^T$ is available. Here ^{T} denotes transpose. Let $\boldsymbol{D}_n = \{(\boldsymbol{x}_i, f(\boldsymbol{x}_i) : i = 1, \ldots, n\}$.

2.3. Gaussian process representation of the objective function

Let $g : \mathbb{R}^d \mapsto \mathbb{R}$ denote a random function such that given $\boldsymbol{D}_n$, $g(\boldsymbol{x}_i) = f(\boldsymbol{x}_i)$ for $i = 1, \ldots, n$.

Since Gaussian processes have the above interpolation property, we model $g(\cdot)$ by a Gaussian process with mean function $\mu(\cdot)$ and covariance function $\sigma^2 c(\cdot, \cdot)$, where σ^2 is the process variance. In other words, $E[g(\boldsymbol{x})] = \mu(\boldsymbol{x})$ for any $\boldsymbol{x} \in \mathcal{X}$ and $Cov(g(\boldsymbol{x}), g(\boldsymbol{y})) = \sigma^2 c(\boldsymbol{x}, \boldsymbol{y})$, for all $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$. Let $\mu(\cdot)$ be continuous in $\mathcal{X}$ and $c(\cdot, \cdot)$ be Lipschitz continuous on $\mathcal{X} \times \mathcal{X}$. Then $g(\cdot)$ is actually a continuous-path Gaussian process with mean function $\mu(\cdot)$ and covariance function $\sigma^2 c(\cdot, \cdot)$.

We shall use the notation $\boldsymbol{g}_n$ to denote $(g(\boldsymbol{x}_1), \ldots, g(\boldsymbol{x}_n))^T$ when distribution of this vector will be considered and $\boldsymbol{f}_n$ when this vector is conditioned upon.

2.4. Gaussian derivative process

For $\boldsymbol{x} = (x_1, \ldots, x_d)$ and $\boldsymbol{y} = (y_1, \ldots, y_d)$, let us assume that the second order mixed partial derivatives

$$\frac{\partial^2 c(\boldsymbol{x}^*, \boldsymbol{y}^*)}{\partial x_i \partial y_i} = \left.\frac{\partial^2 c(\boldsymbol{x}, \boldsymbol{y})}{\partial x_i \partial y_i}\right|_{\boldsymbol{x}=\boldsymbol{x}^*, \boldsymbol{y}=\boldsymbol{y}^*}$$

are Lipschitz continuous on $\mathcal{X} \times \mathcal{X}$ for $i = 1, \ldots, d$.

With the above assumption on the covariance function and with the further assumption that $\mu(\cdot)$ is twice continuously differentiable with continuous mixed second order partial derivatives, for $\boldsymbol{x} = (x_1, \ldots, x_d)$,

$$g_i'(\boldsymbol{x}^*) = \frac{\partial g(\boldsymbol{x}^*)}{\partial x_i} = \left.\frac{\partial g(\boldsymbol{x})}{\partial x_i}\right|_{\boldsymbol{x}=\boldsymbol{x}^*},$$

corresponding to the original continuous-path Gaussian process $g(\cdot)$ exists for $i = 1, \ldots, d$, for all $\boldsymbol{x}^* \in \mathcal{X}$. Specifically, for $i = 1, \ldots, d$, $g_i'(\cdot) = \partial g(\cdot)/\partial x_i$ is a continuous-path Gaussian process with mean function $\mu_i'(\cdot) = \partial \mu(\cdot)/\partial x_i$ and covariance function $\sigma^2 \partial^2 c(\cdot, \cdot)/\partial x_i \partial y_i$.

For general details on Gaussian and Gaussian derivative processes, see, for example, Adler (1981), Adler and Taylor (2007).

2.5. Joint distribution of Gaussian variables and Gaussian derivative variables

Note that, given $\boldsymbol{x}^*, \boldsymbol{y}^* \in \mathcal{X}$,

$$Cov\left(g_i'(\boldsymbol{x}^*), g_j'(\boldsymbol{x}^*)\right) = \sigma^2 \left.\frac{\partial^2 c(\boldsymbol{x}, \boldsymbol{y})}{\partial x_i \partial y_j}\right|_{\boldsymbol{x}=\boldsymbol{x}^*, \boldsymbol{y}=\boldsymbol{x}^*}; \tag{1}$$

$$Cov\left(g_i'(\boldsymbol{x}^*), g(\boldsymbol{y}^*)\right) = \sigma^2 \left.\frac{\partial c(\boldsymbol{x}, \boldsymbol{y})}{\partial x_i}\right|_{\boldsymbol{x}=\boldsymbol{x}^*, \boldsymbol{y}=\boldsymbol{y}^*}. \tag{2}$$

With the above covariance forms, for given $\boldsymbol{x}^* \in \mathcal{X}$, we have

$$\left(g_1'(\boldsymbol{x}^*), \ldots, g_d'(\boldsymbol{x}^*), g(\boldsymbol{x}_1), \ldots, g(\boldsymbol{x}_n)\right)^T \sim N_{d+n}\left(\boldsymbol{\nu}_{d+n}, \sigma^2 \boldsymbol{\Sigma}^{\overline{d+n} \times \overline{d+n}}\right), \tag{3}$$

that is, the vector on the left hand side of (3) has the $(d+n)$-variate normal distribution with mean vector $\boldsymbol{\nu}_{d+n}$ and covariance matrix $\sigma^2\boldsymbol{\Sigma}^{\overline{d+n}\times\overline{d+n}}$. Here

$$\boldsymbol{\nu}_{d+n}(\boldsymbol{x}^*) = \left(\boldsymbol{\mu}_d'(\boldsymbol{x}^*)^T, \boldsymbol{\mu}_n^T\right)^T, \tag{4}$$

where $\boldsymbol{\mu}_d'(\boldsymbol{x}^*) = (\partial\mu(\boldsymbol{x}^*)/\partial x_1, \ldots, \partial\mu(\boldsymbol{x}^*)/\partial x_d)^T$ with $\partial\mu(\boldsymbol{x}^*)/\partial x_i = \left.\frac{\partial\mu(\boldsymbol{x})}{\partial x_i}\right|_{\boldsymbol{x}=\boldsymbol{x}^*}$ for $i = 1, \ldots, d$, and $\boldsymbol{\mu}_n = (\mu(\boldsymbol{x}_1), \ldots, \mu(\boldsymbol{x}_n))^T$. Also,

$$\boldsymbol{\Sigma}^{\overline{d+n}\times\overline{d+n}}(\boldsymbol{x}^*) = \begin{pmatrix} \boldsymbol{\Sigma}_{11}^{d\times d}(\boldsymbol{x}^*) & \boldsymbol{\Sigma}_{12}^{d\times n}(\boldsymbol{x}^*) \\ \boldsymbol{\Sigma}_{21}^{n\times d}(\boldsymbol{x}^*) & \boldsymbol{\Sigma}_{22}^{n\times n} \end{pmatrix}, \tag{5}$$

where $\boldsymbol{\Sigma}_{11}^{d\times d}$ is the d-th order correlation matrix with (i,j)-th element $\sigma^{-2}Cov\left(g_i'(\boldsymbol{x}^*), g_j'(\boldsymbol{x}^*)\right)$ where the covariance term is given by (1), $\boldsymbol{\Sigma}_{12}^{d\times n}(\boldsymbol{x}^*)$ is the $d\times n$ matrix with (i,j)-th element $\sigma^{-2}Cov\left(g_i'(\boldsymbol{x}^*), g(\boldsymbol{x}_j)\right)$ where the covariance term is of the same form as (2) with $\boldsymbol{y}^*$ replaced by $\boldsymbol{x}_j$, $\boldsymbol{\Sigma}_{21}^{n\times d}(\boldsymbol{x}^*)$ is the transpose of $\boldsymbol{\Sigma}_{12}^{d\times n}(\boldsymbol{x}^*)$, and $\boldsymbol{\Sigma}_{22}^{n\times n}$ is the $n\times n$ matrix with (i,j)-th element $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$, the correlation between $g(\boldsymbol{x}_i)$ and $g(\boldsymbol{x}_j)$. In our examples, we shall consider the squared exponential correlation function having the form

$$c(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{y})^T\boldsymbol{\Lambda}^{-1}(\boldsymbol{x}-\boldsymbol{y})\right), \tag{6}$$

for $\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}$, where $\boldsymbol{\Lambda}$ is a $d\times d$ diagonal matrix with positive diagonal elements λ_i; $i = 1, \ldots, d$. It follows that $\boldsymbol{\Sigma}_{11} = \boldsymbol{\Lambda}^{-1}$ and for $j = 1, \ldots, n$, the j-th column of $\boldsymbol{\Sigma}_{12}(\boldsymbol{x}^*)$ is $-\boldsymbol{\Lambda}^{-1}(\boldsymbol{x}^* - \boldsymbol{x}_j)c(\boldsymbol{x}^*, \boldsymbol{x}_j)$.

2.6. Posterior distribution of the Gaussian derivatives given the data and parameters

From (3) it follows that the joint posterior distribution of $\boldsymbol{g}'(\boldsymbol{x}^*) = (g_1'(\boldsymbol{x}^*), \ldots, g_d'(\boldsymbol{x}^*))^T$ given $\boldsymbol{x}^*$, $\boldsymbol{D}_n$, σ^2 and other parameters $\boldsymbol{\theta}$, is the following d-variate normal distribution:

$$\pi\left(\boldsymbol{g}'(\boldsymbol{x}^*)|\sigma^2, \boldsymbol{\theta}, \boldsymbol{D}_n\right) \equiv N_d\left(\tilde{\boldsymbol{\mu}}(\boldsymbol{x}^*), \sigma^2\tilde{\boldsymbol{\Sigma}}(\boldsymbol{x}^*)\right), \tag{7}$$

where

$$\tilde{\boldsymbol{\mu}}(\boldsymbol{x}^*) = \boldsymbol{\mu}_d'(\boldsymbol{x}^*) + \boldsymbol{\Sigma}_{12}(\boldsymbol{x}^*)\boldsymbol{\Sigma}_{22}^{-1}\left(\boldsymbol{f}_n - \boldsymbol{\mu}_n\right); \tag{8}$$

$$\tilde{\boldsymbol{\Sigma}}(\boldsymbol{x}^*) = \boldsymbol{\Sigma}_{11}(\boldsymbol{x}^*) - \boldsymbol{\Sigma}_{12}(\boldsymbol{x}^*)\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}(\boldsymbol{x}^*). \tag{9}$$

2.7. Prior and posterior distributions of the parameters

In our examples we assume that $\mu(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x})^T\boldsymbol{\beta}$, where $\boldsymbol{h}(\boldsymbol{x})^T = (1, x_1, \ldots, x_d)$ and $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_d)^T \in \mathbb{R}^d$. Let $\boldsymbol{H}^{n\times\overline{d+1}} = (\boldsymbol{h}(\boldsymbol{x}_1), \ldots, \boldsymbol{h}(\boldsymbol{x}_n))^T$.

2.7.1. Priors for the parameters

We assume that *a priori*

$$\pi(\boldsymbol{\beta}|\sigma^2) \equiv N_{d+1}\left(\boldsymbol{\beta}_0, \sigma^2\boldsymbol{\Sigma}_0\right), \tag{10}$$

where $\boldsymbol{\beta}_0$ is the mean vector and $\boldsymbol{\Sigma}_0$ is the positive definite covariance matrix, and

$$\pi\left(\sigma^{-2}\right) \equiv \mathcal{G}(a, b), \tag{11}$$

the gamma distribution with mean a/b and variance a/b^2, where $a, b > 0$.

2.7.2. Posteriors for the parameters

Let us first obtain the posterior distribution of σ^{-2} given $\boldsymbol{D}_n$. Note that

$$\begin{aligned}\pi(\sigma^{-2}|\boldsymbol{D}_n) &\propto \pi\left(\sigma^{-2}\right)\pi\left(\boldsymbol{g}_n|\sigma^{-2}\right)\\ &= \pi\left(\sigma^{-2}\right)\int \pi\left(\boldsymbol{g}_n|\sigma^{-2},\boldsymbol{\beta}\right)\pi\left(\boldsymbol{\beta}|\sigma^2\right)d\boldsymbol{\beta}.\end{aligned} \tag{12}$$

To obtain $\pi\left(\boldsymbol{g}_n|\sigma^{-2}\right) = \int \pi\left(\boldsymbol{g}_n|\sigma^{-2},\boldsymbol{\beta}\right)\pi\left(\boldsymbol{\beta}|\sigma^2\right)d\boldsymbol{\beta}$ note that

$$\pi\left(\boldsymbol{g}_n|\sigma^{-2},\boldsymbol{\beta}\right) \equiv N_n\left(\boldsymbol{H}\boldsymbol{\beta}, \sigma^2\boldsymbol{\Sigma}_{22}\right) \tag{13}$$

and since $\pi\left(\boldsymbol{\beta}|\sigma^2\right)$ has the normal distribution (10), it follows that

$$\pi\left(\boldsymbol{g}_n|\sigma^{-2}\right) \equiv N_n\left(\boldsymbol{H}\boldsymbol{\beta}_0, \sigma^2\left(\boldsymbol{H}\boldsymbol{\Sigma}_0\boldsymbol{H}^T + \boldsymbol{\Sigma}_{22}\right)\right). \tag{14}$$

Combining (14) with (12) and (11) it follows that

$$\pi(\sigma^{-2}|\boldsymbol{D}_n) \equiv \mathcal{G}\left(a + \frac{d}{2}, b + \frac{1}{2}\left(\boldsymbol{f}_n - \boldsymbol{H}\boldsymbol{\beta}_0\right)^T\left(\boldsymbol{H}\boldsymbol{\Sigma}_0\boldsymbol{H}^T + \boldsymbol{\Sigma}_{22}\right)^{-1}\left(\boldsymbol{f}_n - \boldsymbol{H}\boldsymbol{\beta}_0\right)\right). \tag{15}$$

Also, combining (13) and (10) it is easy to see that

$$\pi\left(\boldsymbol{\beta}|\boldsymbol{D}_n, \sigma^2\right) \equiv N_{d+1}\left(\left(\boldsymbol{H}^T\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{H} + \boldsymbol{\Sigma}_0^{-1}\right)^{-1}\left(\boldsymbol{H}^T\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{f}_n + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\beta}_0\right), \sigma^2\left(\boldsymbol{H}^T\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{H} + \boldsymbol{\Sigma}_0^{-1}\right)^{-1}\right). \tag{16}$$

2.8. Marginal posterior distribution of the derivative process

Now, from (7) it follows that

$$\pi\left(\boldsymbol{g}'(\boldsymbol{x}^*)|\sigma^2,\boldsymbol{\beta},\boldsymbol{D}_n\right) \equiv N_d\left(\boldsymbol{A}\boldsymbol{\beta} + \boldsymbol{\Sigma}_{12}(\boldsymbol{x}^*)\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{f}_n - \boldsymbol{H}\boldsymbol{\beta}), \sigma^2\tilde{\boldsymbol{\Sigma}}(\boldsymbol{x}^*)\right), \tag{17}$$

where $\boldsymbol{A}^{d\times\overline{d+1}} = \left(\mathbf{0}^{d\times 1} \quad \mathbb{I}_d\right)$. Here $\mathbf{0}^{d\times 1}$ is the d-dimensional null vector and $\mathbb{I}_d$ is the identity matrix of order d. Integrating (17) with respect to (16) we obtain

$$\pi\left(\boldsymbol{g}'(\boldsymbol{x}^*)|\sigma^2,\boldsymbol{D}_n\right) \equiv N_d\left(\hat{\boldsymbol{\mu}}'(\boldsymbol{x}^*), \sigma^2\hat{\boldsymbol{\Sigma}}(\boldsymbol{x}^*)\right), \tag{18}$$

where $\hat{\boldsymbol{\mu}}'(\boldsymbol{x}^*)$ and $\hat{\boldsymbol{\Sigma}}(\boldsymbol{x}^*)$ are given by

$$\hat{\boldsymbol{\mu}}'(\boldsymbol{x}^*) = \boldsymbol{A}\hat{\boldsymbol{\beta}} + \boldsymbol{\Sigma}_{12}(\boldsymbol{x}^*)\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{f}_n - \boldsymbol{H}\hat{\boldsymbol{\beta}}); \tag{19}$$

$$\hat{\boldsymbol{\Sigma}}(\boldsymbol{x}^*) = \tilde{\boldsymbol{\Sigma}}(\boldsymbol{x}^*) + \left(\boldsymbol{A} - \boldsymbol{\Sigma}_{12}(\boldsymbol{x}^*)\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{H}\right)\left(\boldsymbol{H}^T\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{H} + \boldsymbol{\Sigma}_0^{-1}\right)^{-1}\left(\boldsymbol{A} - \boldsymbol{\Sigma}_{12}(\boldsymbol{x}^*)\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{H}\right)^T, \tag{20}$$

with

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{H}^T\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{H} + \boldsymbol{\Sigma}_0^{-1}\right)^{-1}\left(\boldsymbol{H}^T\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{f}_n + \boldsymbol{\Sigma}_0^{-1}\boldsymbol{\beta}_0\right). \tag{21}$$

Integrating (18) with respect to (15) we obtain

$$\pi\left(\boldsymbol{g}'(\boldsymbol{x}^*)|\boldsymbol{D}_n\right) \equiv t_d\left(\hat{\boldsymbol{\mu}}'(\boldsymbol{x}^*), \frac{\left(a + \frac{d}{2}\right)\hat{\boldsymbol{\Sigma}}(\boldsymbol{x}^*)^{-1}}{b + \frac{1}{2}\left(\boldsymbol{f}_n - \boldsymbol{H}\boldsymbol{\beta}_0\right)^T\left(\boldsymbol{H}\boldsymbol{\Sigma}_0\boldsymbol{H}^T + \boldsymbol{\Sigma}_{22}\right)^{-1}\left(\boldsymbol{f}_n - \boldsymbol{H}\boldsymbol{\beta}_0\right)}, 2\left(a + \frac{d}{2}\right)\right), \tag{22}$$

where for any d-dimensional vector $\boldsymbol{\mu}$, d-th order covariance matrix $\boldsymbol{\Sigma}$, and $\alpha > 0$, $t_d\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}, \alpha\right)$ is a d-variate Student's t distribution with density at $\boldsymbol{x} \in \mathbb{R}^d$ given by

$$t_d\left(\boldsymbol{x} : \boldsymbol{\mu}, \boldsymbol{\Sigma}^{-1}, \alpha\right) = C\left[1 + \alpha^{-1}(\boldsymbol{x} - \boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\right]^{-\frac{(\alpha+d)}{2}},$$

where

$$C = \frac{\Gamma\left(\frac{\alpha+d}{2}\right)}{\Gamma\left(\frac{\alpha}{2}\right)(\alpha\pi)^{\frac{d}{2}}},$$

with $\Gamma(\cdot)$ denoting the gamma function.

3. Posterior distribution of random optima corresponding to the posterior derivative process

From (22) we obtain the following posterior density at $\boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}$:

$$\pi\left(\boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}|\boldsymbol{D}_n\right) \propto \left[1 + \frac{\hat{\boldsymbol{\mu}}'(\boldsymbol{x}^*)^T\hat{\boldsymbol{\Sigma}}(\boldsymbol{x}^*)^{-1}\hat{\boldsymbol{\mu}}'(\boldsymbol{x}^*)}{2b + \left(\boldsymbol{f}_n - \boldsymbol{H}\boldsymbol{\beta}_0\right)^T\left(\boldsymbol{H}\boldsymbol{\Sigma}_0\boldsymbol{H}^T + \boldsymbol{\Sigma}_{22}\right)^{-1}\left(\boldsymbol{f}_n - \boldsymbol{H}\boldsymbol{\beta}_0\right)}\right]^{-(a+d)}. \tag{23}$$

Now, with prior $\pi(\boldsymbol{x}^*)$ on $\boldsymbol{x}^*$, the posterior of $\boldsymbol{x}^*$, given $\boldsymbol{g}'(\boldsymbol{x}^*)$ and $\boldsymbol{D}_n$ can be obtained as follows:

$$\begin{aligned}
\pi(\boldsymbol{x}^*|\boldsymbol{g}'(\boldsymbol{x}^*), \boldsymbol{D}_n) &\propto \pi(\boldsymbol{x}^*)\pi(\boldsymbol{g}'(\boldsymbol{x}^*), \boldsymbol{g}_n|\boldsymbol{x}^*) \\
&= \pi(\boldsymbol{x}^*)\pi(\boldsymbol{g}'(\boldsymbol{x}^*)|\boldsymbol{D}_n, \boldsymbol{x}^*)\pi(\boldsymbol{g}_n|\boldsymbol{x}^*) \\
&= \pi(\boldsymbol{x}^*)\pi(\boldsymbol{g}'(\boldsymbol{x}^*)|\boldsymbol{D}_n, \boldsymbol{x}^*)\pi(\boldsymbol{g}_n) \\
&\propto \pi(\boldsymbol{x}^*)\pi(\boldsymbol{g}'(\boldsymbol{x}^*)|\boldsymbol{D}_n, \boldsymbol{x}^*).
\end{aligned} \tag{24}$$

In the second step of (24), $\pi(\boldsymbol{g}_n|\boldsymbol{x}^*)$ is the marginal distribution of $\boldsymbol{g}_n$, integrated over the parameters. Since this does not depend upon $\boldsymbol{x}^*$, we denoted this as $\pi(\boldsymbol{g}_n)$ in the third step of (24). From (24) it then follows that

$$\pi(\boldsymbol{x}^*|\boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}, \boldsymbol{D}_n) \propto \pi(\boldsymbol{x}^*)\pi(\boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}|\boldsymbol{D}_n, \boldsymbol{x}^*). \tag{25}$$

Now, $\pi(\boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}|\boldsymbol{D}_n, \boldsymbol{x}^*)$ will also depend upon parameters of the covariance function, which will be unknown generally. It is legitimate to estimate them using the maximum likelihood estimation method (see, for example, Santner *et al.* (2003)) and treat them as fixed.

The formula (25) holds for any prior $\pi(\boldsymbol{x}^*)$ for $\boldsymbol{x}^*$. However, we shall consider a uniform prior on $\mathcal{X}$ constrained by the first and second derivatives of $f(\cdot)$. The details are presented below.

3.1. Prior for $\boldsymbol{x}^*$

Without loss of generality, let us assume that our objective is to obtain the minima of the function $f(\cdot)$ on $\mathcal{X}$. For $i, j = 1, \ldots, d$, let $f''_{ij}(\boldsymbol{x}^*) = \frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j}\Big|_{\boldsymbol{x}=\boldsymbol{x}^*}$ denote the second order partial derivatives of the objective function $f(\cdot)$ at any $\boldsymbol{x}^* \in \mathcal{X}$. Let $\boldsymbol{\Sigma}''(\boldsymbol{x}^*)$ stand for the $d \times d$ matrix of such second order partial derivatives at $\boldsymbol{x}^*$ with (i, j)-th element $f''_{ij}(\boldsymbol{x}^*)$. Let $\boldsymbol{\Sigma}''(\boldsymbol{x}^*) > 0$ denote that $\boldsymbol{\Sigma}''(\boldsymbol{x}^*)$ is positive definite. Then we consider the following prior for $\boldsymbol{x}^*$:

$$\pi(\boldsymbol{x}^*) \propto I_{B(\epsilon)}(\boldsymbol{x}^*), \tag{26}$$

where, for any set A and vector $\boldsymbol{x}$, $I_A(\boldsymbol{x}) = 1$ if $\boldsymbol{x} \in A$ and zero otherwise. Also, for any $\boldsymbol{x}$ and $\epsilon > 0$,

$$B(\epsilon) = \mathcal{X} \cap \{\boldsymbol{x} : \|\boldsymbol{f}'(\boldsymbol{x})\|_d < \epsilon\} \cap \{\boldsymbol{x} : \boldsymbol{\Sigma}''(\boldsymbol{x}) > 0\}, \tag{27}$$

where $\|\cdot\|_d$ denotes the Euclidean norm in the d-dimensional Euclidean space.

3.2. Interpretation of the posterior for $\boldsymbol{x}^*$ as a pseudo-posterior

Before proceeding to the convergence theory, we comment on the nature of the target in (25). The event $\{\boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}\}$ has probability zero under the continuous distribution of the derivative process. Therefore, the expression $\pi(\boldsymbol{x}^*|\boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}, \boldsymbol{D}_n)$ is not a conditional probability in the strict Kolmogorov sense. Instead, it should be understood as a density-based target obtained by evaluating the conditional density of the derivative at zero, which serves as a surrogate posterior for the stationary point. This construction is akin to a Gibbs posterior or an energy-based formulation; it is well-defined as a density and leads to a valid sampling scheme. Such pseudo-posteriors have been successfully used in various contexts (see, for example, Bissiri *et al.* (2016)) and we adopt a similar perspective here.

4. Almost sure uniform convergence of posterior Gaussian and Gaussian derivative processes

Consider the joint posterior distribution of

$$\pi(g(\cdot), \boldsymbol{g}'(\cdot)|\boldsymbol{D}_n) = \pi(g(\cdot)|\boldsymbol{D}_n)\pi(\boldsymbol{g}'(\cdot)|g(\cdot), \boldsymbol{D}_n). \tag{28}$$

Then the marginal posterior $\pi(\boldsymbol{g}'(\cdot)|\boldsymbol{D}_n)$ is of the same form as (22), and the marginal posterior distribution $\pi(g(\cdot)|\boldsymbol{D}_n)$ in (28) corresponds to the t_1 process, the form of which is not relevant for our purpose.

For $n \geq 1$, let $\boldsymbol{X}_n$ denote the n input points in $\boldsymbol{D}_n$. Note that even after marginalizing out the parameters of the Gaussian process with respect to their posteriors (here $\boldsymbol{\beta}$ and σ^2),

the interpolation property of $g(\cdot)$ given $\boldsymbol{D}_n$ is preserved. That is, the marginal posterior $\pi(g(\boldsymbol{x}^*)|\boldsymbol{D}_n)$ gives full posterior mass to $f(\boldsymbol{x}^*)$ if $\boldsymbol{x}^* \in \boldsymbol{X}_n$.

Let $g_n(\cdot)$ denote any random function associated with any non-null set of the marginalized posterior measure of $g(\cdot)$ given $\boldsymbol{D}_n$ (here, the t_1 posterior measure). Also, let $\boldsymbol{g}'_n(\cdot)$ denote any d-dimensional random function associated with any non-null set of the marginalized posterior measure of $\boldsymbol{g}'(\cdot)$ given $\boldsymbol{D}_n$, the form of which is provided explicitly by (22). Theorems 1, 2, 3 and 4 prove almost sure uniform convergence of $g_n(\cdot)$ and $\boldsymbol{g}'_n(\cdot)$ to $f(\cdot)$ and $\boldsymbol{f}'(\cdot)$ respectively, as $n \to \infty$. In particular, Theorems 3 and 4 also provide rates of such convergences.

Before presenting the theorems, we state a set of standing assumptions that will be used throughout this section.

Assumption 4.1 (Assumptions for convergence results)**:** We assume the following:

(A1) The domain $\mathcal{X} \subset \mathbb{R}^d$ is compact and convex.

(A2) The objective function f is twice continuously differentiable on an open set containing $\mathcal{X}$, with continuous mixed second-order partial derivatives.

(A3) The mean function $\mu(\cdot)$ of the Gaussian process prior is twice continuously differentiable with continuous mixed second-order partial derivatives.

(A4) The correlation function $c(\cdot,\cdot)$ is such that the mixed partial derivatives $\frac{\partial^2 c(\boldsymbol{x},\boldsymbol{y})}{\partial x_i \partial y_i}$ exist and are Lipschitz continuous on $\mathcal{X} \times \mathcal{X}$ for all $i = 1, \ldots, d$. For the higher-order convergence results (Theorems 3 and 4), we additionally require the existence and Lipschitz continuity of $\frac{\partial^4 c(\boldsymbol{x},\boldsymbol{y})}{\partial x_i \partial y_i \partial x_j \partial y_j}$ for all i, j.

(A5) The input points $\boldsymbol{X}_n$ satisfy a fixed-domain infill asymptotics condition: for any $\boldsymbol{x} \in \mathcal{X}$, there exists a sequence $\boldsymbol{x}_n \in \boldsymbol{X}_n$ such that

$$\|\boldsymbol{x}_n - \boldsymbol{x}\|_d \to 0 \text{ as } n \to \infty, \tag{29}$$

and moreover points of the form $\boldsymbol{x}_n + \boldsymbol{h}_n$ with $\boldsymbol{h}_n \to \boldsymbol{0}$ also belong to $\boldsymbol{X}_n$.

(A6) The correlation matrices $\boldsymbol{\Sigma}_{22}$ and their derivatives are uniformly non-degenerate: there exists a constant $c_0 > 0$ such that the smallest eigenvalue of $\boldsymbol{\Sigma}_{22}$ is bounded below by c_0 for all n.

Assumption (A6) is used in two essential places. First, the posterior distributions derived in Sections 2.6 and 2.8 explicitly depend on $\boldsymbol{\Sigma}_{22}^{-1}$; the uniform lower bound on the smallest eigenvalue guarantees that $\boldsymbol{\Sigma}_{22}$ is invertible for every n and that its inverse remains numerically stable as n increases. Second, in the proofs of Theorems 2 and 3 we require a uniform bound on the second derivatives $g''_{ijn}(\cdot)$ over n. Such a bound follows from the uniform boundedness of the conditional covariance $\tilde{\boldsymbol{\Sigma}}(\boldsymbol{x}^*)$ in (9) and the stability of the posterior of $\boldsymbol{g}'(\boldsymbol{x}^*)$; both rely on $\boldsymbol{\Sigma}_{22}^{-1}$ not becoming arbitrarily large. Consequently, (A6) ensures that the almost sure uniform convergence results hold without pathological degeneracy of the design points.

These assumptions are standard in the literature on fixed-domain asymptotics for Gaussian processes (see, for example, Stein (1999)) and are sufficient for the proofs that follow.

Lemma 1: Consider a sequence of real-valued continuous functions $\{f_n\}_{n=1}^{\infty}$ on any compact set $\mathcal{X}$ such that $f_n(\boldsymbol{x}) \to f(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$, where f is some real-valued continuous function on $\mathcal{X}$. Then

$$\lim_{n\to\infty} \sup_{\boldsymbol{x}\in\mathcal{X}} |f_n(\boldsymbol{x})| < \infty.$$

Proof: Note that f_n, for $n \geq 1$, and f are actually uniformly continuous since $\mathcal{X}$ is compact. Now let us first consider an arbitrary $\boldsymbol{x}_1 \in \mathcal{X}$. Then due to pointwise convergence of f_n to f, for any $\epsilon > 0$, there exists $n_1 \geq 1$ such that for $n \geq n_1$, $|f_n(\boldsymbol{x}_1) - f(\boldsymbol{x}_1)| < \epsilon$. Moreover, due to uniform continuity of f_n and f, there exists an open neighborhood $\mathcal{N}(\boldsymbol{x}_1)$ of $\boldsymbol{x}_1$ such that $|f_n(\boldsymbol{x}) - f(\boldsymbol{x})| < \epsilon_1$ for all $\boldsymbol{x} \in \mathcal{N}(\boldsymbol{x}_1)$, where ϵ_1 is some positive finite constant. Since f is continuous on the compact set $\mathcal{X}$, it is uniformly bounded. Hence, $\sup_{\boldsymbol{x}\in\mathcal{N}(\boldsymbol{x}_1)} |f_n(\boldsymbol{x})| < M_1$ for all $n \geq n_1$, where M_1 is some positive finite constant.

Now consider another point $\boldsymbol{x}_2 \in \mathcal{X}\backslash\mathcal{N}(\boldsymbol{x}_1)$. Then similar argument shows that $\sup_{\boldsymbol{x}\in\mathcal{N}(\boldsymbol{x}_2)} |f_n(\boldsymbol{x})| < M_2$ for all $n \geq n_2 \geq n_1$, where $\mathcal{N}(\boldsymbol{x}_2)$ is some appropriate open neighborhood of $\boldsymbol{x}_2$ and M_2 is some positive finite constant.

Thus, starting with $\mathcal{N}(\boldsymbol{x}_1)$ and the associated bound $\sup_{\boldsymbol{x}\in\mathcal{N}(\boldsymbol{x}_1)} |f_n(\boldsymbol{x})| < M_1$, continuing the procedure for $i \geq 2$, we can construct neighborhoods $\mathcal{N}(\boldsymbol{x}_i)$ with $\boldsymbol{x}_i \in \mathcal{X}\backslash \cup_{j=1}^{i-1} \mathcal{N}(\boldsymbol{x}_j)$ and bounds M_i such that for all $n \geq n_i \geq n_{i-1} \geq \cdots \geq n_2 \geq n_1$, $\sup_{\boldsymbol{x}\in\mathcal{N}(\boldsymbol{x}_i)} |f_n(\boldsymbol{x})| < M_i$. Note that $\mathcal{X} \subseteq \cup_{i=1}^{\infty}\mathcal{N}(\boldsymbol{x}_i)$. That is, the set of neighborhoods $\{\mathcal{N}(\boldsymbol{x}_i) : i = 1, 2, \ldots\}$ constitutes an open cover for $\mathcal{X}$. Since $\mathcal{X}$ is compact, there exists a finite sub-cover for $\mathcal{X}$, say, $\{\mathcal{N}(\boldsymbol{x}_{i_j}) : j = 1, 2, \ldots, K\}$, where K is finite. Now, by our construction, for $n \geq n_{i_j}$, $\sup_{\boldsymbol{x}\in\mathcal{N}(\boldsymbol{x}_{i_j})} |f_n(\boldsymbol{x})| < M_{i_j}$, for $j = 1, \ldots, K$. Let $n_0 = \max\{n_{i_j} : j = 1, \ldots, K\}$ and $M = \max\{M_{i_j} : j = 1, \ldots, K\}$. Then for all $n \geq n_0$, $\sup_{\boldsymbol{x}\in\mathcal{X}} |f_n(\boldsymbol{x})| < M < \infty$. □

Theorem 1: Consider a fixed-domain infill asymptotics framework satisfying Assumption (A5). Under Assumptions (A1)–(A4) and (A6), for almost all sequences $\{g_n(\cdot)\}_{n=1}^{\infty}$,

$$\sup_{\boldsymbol{x}\in\mathcal{X}} |g_n(\boldsymbol{x}) - f(\boldsymbol{x})| \to 0, \text{ as } n \to \infty. \tag{30}$$

Proof: Consider any $\boldsymbol{x} \in \mathcal{X}$. Then there exists $\boldsymbol{x}_n \in \boldsymbol{X}_n$ for $n \geq 1$ satisfying (29). Now by Taylor's series expansion up to the first order,

$$g_n(\boldsymbol{x}_n) = g_n(\boldsymbol{x}) + (\boldsymbol{x}_n - \boldsymbol{x})^T \boldsymbol{g}_n'(\boldsymbol{c}_n), \tag{31}$$

where $\boldsymbol{c}_n$ lies on the line joining $\boldsymbol{x}$ and $\boldsymbol{x}_n - \boldsymbol{x}$.

Now, for $i = 1, \ldots, d$, consider the i-th partial derivative $g_{in}'(\cdot)$ of $g_n(\cdot)$. With any sequence $h_{in} \to 0$ as $n \to \infty$, we have

$$\frac{g_n(x_{1n}, \ldots, x_{i-1,n}, x_{in} + h_{in}, x_{i+1,n}, \ldots, x_{dn}) - g_n(\boldsymbol{x}_n)}{h_{in}} = g_{in}'(\boldsymbol{x}_n^*), \tag{32}$$

where $\boldsymbol{x}_n^* = (x_{1n}, \ldots, x_{i-1,n}, x_{in}^*, x_{i+1,n}, \ldots, x_{dn})$; here x_{in}^* lies between x_{in} and $x_{in} + h_{in}$. Since $(x_{1n}, \ldots, x_{i-1,n}, x_{in} + h_{in}, x_{i+1,n}, \ldots, x_{dn})^T \in \boldsymbol{X}_n$ and $\boldsymbol{x}_n \in \boldsymbol{X}_n$, $g_n(x_{1n}, \ldots, x_{i-1,n}, x_{in} + h_{in}, x_{i+1,n}, \ldots, x_{dn}) = f(x_{1n}, \ldots, x_{i-1,n}, x_{in} + h_{in}, x_{i+1,n}, \ldots, x_{dn})$ and $g_n(\boldsymbol{x}_n) = f(\boldsymbol{x}_n)$, almost surely. Hence, from (32) it follows that

$$g'_{in}(\boldsymbol{x}_n^*) = f'_i(\boldsymbol{z}_n), \text{ almost surely,} \tag{33}$$

with $\boldsymbol{z}_n = (x_{1n}, \ldots, x_{i-1,n}, z_{in}, x_{i+1,n}, \ldots, x_{dn})$, where z_{in} lies between x_{in} and $x_{in} + h_{in}$. Clearly, $\boldsymbol{z}_n \to \boldsymbol{x}$, as $n \to \infty$. Hence, taking limits of both sides of (33) as $n \to \infty$, and using continuity of $f'_i(\cdot)$, yields

$$\lim_{n\to\infty} g'_{in}(\boldsymbol{x}_n^*) = f'_i(\boldsymbol{x}), \text{ almost surely.} \tag{34}$$

Now, by the hypothesis of Lipschitz continuity of the second order mixed partial derivatives of the correlation function ensures existence and sample path continuity of the partial derivatives $g'_{in}(\cdot)$, for $i = 1, \ldots, d$, for any $n \geq 1$. Since $\mathcal{X}$ is compact, $g'_{in}(\cdot)$ are uniformly continuous on $\mathcal{X}$, for $i = 1, \ldots, d$, for any $n \geq 1$. Uniform continuity of $g'_{in}(\cdot)$ for all $n \geq 1$ implies that for any $\epsilon > 0$, $|g'_{in}(\boldsymbol{x}_n^*) - g'_{in}(\boldsymbol{x})| < \epsilon$, whenever $\|\boldsymbol{x}_n^* - \boldsymbol{x}\|_d < \delta$, where δ (> 0) depends upon ϵ only. Now, since $\boldsymbol{x}_n^* \to \boldsymbol{x}$ as $n \to \infty$, there exists n_0 (≥ 1) depending upon δ such that $\|\boldsymbol{x}_n^* - \boldsymbol{x}\|_d < \delta$ for $n \geq n_0$. Further, using (34) we obtain for any $\boldsymbol{x} \in \mathcal{X}$, the following:

$$\lim_{n\to\infty} g'_{in}(\boldsymbol{x}) = \lim_{n\to\infty} g'_{in}(\boldsymbol{x}_n^*) + \lim_{n\to\infty} (g'_{in}(\boldsymbol{x}) - g'_{in}(\boldsymbol{x}_n^*)) = f'_i(\boldsymbol{x}), \text{ almost surely.} \tag{35}$$

That is, $g'_{in}(\cdot)$ converges pointwise to $f'_i(\cdot)$ almost surely, as $n \to \infty$. Moreover, $g'_{in}(\cdot)$ is almost surely continuous on $\mathcal{X}$ for all $n \geq 1$ and $f'_i(\cdot)$ is continuous on $\mathcal{X}$. Since $\mathcal{X}$ is compact, we invoke Lemma 1 to conclude that there exists a positive, finite constant M depending upon $f'_i(\cdot)$; $i = 1, \ldots, d$ such that

$$\lim_{n\to\infty} \sup_{\boldsymbol{x}\in\mathcal{X}} |g'_{in}(\boldsymbol{x})| < M, \text{ almost surely, for } i = 1, \ldots, d. \tag{36}$$

Hence, using the Cauchy-Schwartz inequality in (31), boundedness of the partial derivatives $g'_{in}(\cdot)$ for $i = 1, \ldots, d$ for large enough n and (29), we obtain

$$|g_n(\boldsymbol{x}_n) - g_n(\boldsymbol{x})| = |(\boldsymbol{x}_n - \boldsymbol{x})^T \boldsymbol{g}'_n(\boldsymbol{c}_n)| \leq \|\boldsymbol{x}_n - \boldsymbol{x}\|_d \times \|\boldsymbol{g}'_n(\boldsymbol{c}_n)\|_d \to 0, \text{ as } n \to \infty. \tag{37}$$

Hence, using (37) and continuity of $f(\cdot)$ we obtain, for any $\boldsymbol{x} \in \mathcal{X}$,

$$\lim_{n\to\infty} g_n(\boldsymbol{x}) = \lim_{n\to\infty} g_n(\boldsymbol{x}_n) = \lim_{n\to\infty} f(\boldsymbol{x}_n) = f(\lim_{n\to\infty} \boldsymbol{x}_n) = f(\boldsymbol{x}), \tag{38}$$

proving pointwise convergence of $g_n(\cdot)$ to $f(\cdot)$.

Thus, we have shown that $g_n(\cdot)$ converges pointwise to $f(\cdot)$ on $\mathcal{X}$ almost surely, as $n \to \infty$ (equation (38)), and also that the partial derivatives of $g_n(\cdot)$ are uniformly bounded in the limit almost surely (equation (36)). The latter also implies that $g_n(\cdot)$ is almost surely Lipschitz continuous on $\mathcal{X}$. Since $\mathcal{X}$ is compact, by the stochastic Arzelà–Ascoli lemma (see, for example, Theorem 1.6.6 of van der Vaart and Wellner (1996)), which states that if a sequence of random continuous functions is pointwise convergent and has uniformly bounded derivatives (hence equicontinuous and tight), then uniform convergence follows, we conclude that (30) holds. □

Theorem 2: Consider the same setup as in Theorem 1 with the additional smoothness condition on the correlation function (fourth-order mixed derivatives Lipschitz). Then, for almost all sequences $\{\boldsymbol{g}_n'(\cdot)\}_{n=1}^{\infty}$,

$$\sup_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{g}_n'(\boldsymbol{x}) - \boldsymbol{f}'(\boldsymbol{x})\|_d \to 0, \text{ as } n \to \infty. \tag{39}$$

Proof: Note that for $i = 1, \ldots, d$, pointwise convergence of $g_{in}'(\cdot)$ to $f_i'(\cdot)$ as $n \to \infty$, is already shown by (35), in the proof of Theorem 1. Hence, if we can show that for $i, j = 1, \ldots, d$, the absolute values of the second order partial derivatives $|g_{ijn}''(\cdot)|$ are uniformly bounded on $\mathcal{X}$ as $n \to \infty$, then this would imply that $g_{in}'(\cdot)$ are almost surely Lipschitz continuous on $\mathcal{X}$ for large enough n. Since $\mathcal{X}$ is compact, this would then imply by the stochastic Arzelà–Ascoli lemma (Theorem 1.6.6 of van der Vaart and Wellner (1996)) that $\sup_{\boldsymbol{x}\in\mathcal{X}} |g_{in}'(\boldsymbol{x}) - f_i'(\boldsymbol{x})| \to 0$, almost surely, s $n \to \infty$, for $i = 1, \ldots, d$, which is equivalent to (39). Hence, in the rest of the proof, we show that $\lim_{n\to\infty}\sup_{\boldsymbol{x}\in\mathcal{X}} |g_{ijn}''(\boldsymbol{x})| < \infty$. The argument is given for $i = j$; mixed partials follow similarly.

Fix $\mathbf{x} \in \mathcal{X}$. By the infill design condition (A5) there exists a sequence $\mathbf{x}_n \in X_n$ (the set of design points) such that $\mathbf{x}_n \to \mathbf{x}$. Choose a scalar $h_n > 0$ with $h_n \to 0$ and such that $\mathbf{x}_n \pm h_n\mathbf{e}_i \in X_n$, where $\mathbf{e}_i$ is the i-th standard basis vector. This is possible because (A5) requires points of the form $\mathbf{x}_n + \mathbf{h}_n$ with $\mathbf{h}_n \to \mathbf{0}$ to belong to X_n.

Since f is twice continuously differentiable (Assumption A2), Taylor's theorem with Lagrange remainder gives

$$f(\mathbf{x}_n + h_n\mathbf{e}_i) = f(\mathbf{x}_n) + f_i'(\mathbf{x}_n)h_n + \tfrac{1}{2}f_{ii}''(\mathbf{x}_n + \theta_{1n}h_n\mathbf{e}_i)h_n^2, \tag{40}$$

$$f(\mathbf{x}_n - h_n\mathbf{e}_i) = f(\mathbf{x}_n) - f_i'(\mathbf{x}_n)h_n + \tfrac{1}{2}f_{ii}''(\mathbf{x}_n - \theta_{2n}h_n\mathbf{e}_i)h_n^2, \tag{41}$$

with $\theta_{1n}, \theta_{2n} \in (0, 1)$. Adding the two equalities and subtracting $2f(\mathbf{x}_n)$ yields

$$\frac{f(\mathbf{x}_n + h_n\mathbf{e}_i) - 2f(\mathbf{x}_n) + f(\mathbf{x}_n - h_n\mathbf{e}_i)}{h_n^2} = \tfrac{1}{2}\Big[f_{ii}''(\mathbf{x}_n + \theta_{1n}h_n\mathbf{e}_i) + f_{ii}''(\mathbf{x}_n - \theta_{2n}h_n\mathbf{e}_i)\Big].$$

Because f_{ii}'' is Lipschitz continuous on the compact set $\mathcal{X}$ (Assumption A4), there exists a constant $L_f < \infty$ such that $|f_{ii}''(\mathbf{y}) - f_{ii}''(\mathbf{z})| \le L_f\|\mathbf{y} - \mathbf{z}\|_d$ for all $\mathbf{y}, \mathbf{z} \in \mathcal{X}$. Consequently,

$$\left|\tfrac{1}{2}\Big[f_{ii}''(\mathbf{x}_n+\theta_{1n}h_n\mathbf{e}_i)+f_{ii}''(\mathbf{x}_n-\theta_{2n}h_n\mathbf{e}_i)\Big]-f_{ii}''(\mathbf{x})\right| \le \tfrac{L_f}{2}\Big(\|\mathbf{x}_n+\theta_{1n}h_n\mathbf{e}_i-\mathbf{x}\|_d+\|\mathbf{x}_n-\theta_{2n}h_n\mathbf{e}_i-\mathbf{x}\|_d\Big).$$

Since $\mathbf{x}_n \to \mathbf{x}$ and $h_n \to 0$, the right-hand side is bounded by $C_f h_n$ for some constant C_f (depending only on L_f and the rate of convergence of $\mathbf{x}_n$). Thus we obtain the remainder bound

$$\left|\frac{f(\mathbf{x}_n + h_n\mathbf{e}_i) - 2f(\mathbf{x}_n) + f(\mathbf{x}_n - h_n\mathbf{e}_i)}{h_n^2} - f_{ii}''(\mathbf{x})\right| \le C_f h_n. \tag{42}$$

The random function g_n is almost surely twice continuously differentiable because the covariance kernel satisfies the Lipschitz condition (A4) and the mean function is twice

differentiable (A3). Moreover, the interpolation property of the posterior gives $g_n(\mathbf{x}_n \pm h_n\mathbf{e}_i) = f(\mathbf{x}_n \pm h_n\mathbf{e}_i)$ and $g_n(\mathbf{x}_n) = f(\mathbf{x}_n)$ almost surely, since these points belong to X_n. Repeating the Taylor expansion for g_n yields

$$\frac{g_n(\mathbf{x}_n + h_n\mathbf{e}_i) - 2g_n(\mathbf{x}_n) + g_n(\mathbf{x}_n - h_n\mathbf{e}_i)}{h_n^2} = \tfrac{1}{2}\big[g''_{iin}(\mathbf{x}_n + \tilde{\theta}_{1n}h_n\mathbf{e}_i) + g''_{iin}(\mathbf{x}_n - \tilde{\theta}_{2n}h_n\mathbf{e}_i)\big],$$

with $\tilde{\theta}_{1n}, \tilde{\theta}_{2n} \in (0,1)$. From the proof of uniform boundedness of second derivatives (see the discussion preceding this theorem) we know that $\sup_n \sup_{\mathbf{y}\in\mathcal{X}} |g''_{iin}(\mathbf{y})| < \infty$ almost surely, and the sample paths are equicontinuous. Hence there exists a constant C_g (independent of n) such that

$$\left|\tfrac{1}{2}\big[g''_{iin}(\mathbf{x}_n + \tilde{\theta}_{1n}h_n\mathbf{e}_i) + g''_{iin}(\mathbf{x}_n - \tilde{\theta}_{2n}h_n\mathbf{e}_i)\big] - g''_{iin}(\mathbf{x})\right| \le C_g h_n. \tag{43}$$

The left-hand sides of (42) and (43) are exactly equal because they involve the same function values f at the three design points. Therefore we have

$$g''_{iin}(\mathbf{x}) + \tilde{R}_n = f''_{ii}(\mathbf{x}) + R_n,$$

where $|R_n| \le C_f h_n$ and $|\tilde{R}_n| \le C_g h_n$. Rearranging gives

$$|g''_{iin}(\mathbf{x}) - f''_{ii}(\mathbf{x})| = |\tilde{R}_n - R_n| \le (C_f + C_g)h_n.$$

Since $h_n \to 0$, we conclude

$$\lim_{n\to\infty} g''_{iin}(\mathbf{x}) = f''_{ii}(\mathbf{x}) \quad \text{almost surely.} \tag{44}$$

For $i \neq j$, a similar argument using second-order mixed differences (for example, with $\mathbf{h}_{ijn} = h_n\mathbf{e}_i + h_n\mathbf{e}_j$) or the polarization identity yields

$$\lim_{n\to\infty} g''_{ijn}(\mathbf{x}) = f''_{ij}(\mathbf{x}) \quad \text{almost surely.} \tag{45}$$

The remainder bounds again rely on the Lipschitz continuity of f'_{ij} and the uniform boundedness of g''_{ijn}.

Since $g''_{ijn}(\cdot)$ is almost surely continuous for all $n \ge 1$ and $f''_{ij}(\cdot)$ is continuous, with $\mathcal{X}$ being compact, the pointwise convergence results (44) and (45) lets us conclude, using Lemma 1, that $\lim\sup_{n\to\infty} \sup_{\boldsymbol{x}\in\mathcal{X}} |g''_{ijn}(\boldsymbol{x})| < \infty$.

□

Remark 1: The proof above never divides by $f'_{ii}(\mathbf{x})$, so it remains valid when $f'_{ii}(\mathbf{x}) = 0$. In that case the bound $|g''_{iin}(\mathbf{x})| \le (C_f + C_g)h_n$ directly forces convergence to zero. Hence no special treatment is required.

Theorems 1 and 2 prove almost sure uniform convergence of $g_n(\cdot)$ and $\boldsymbol{g}'_n(\cdot)$ to $f(\cdot)$ and $\boldsymbol{f}'(\cdot)$, respectively. However, the rates of convergence are not provided by these theorems. Further fine-tuning the structure of the set of input points $\boldsymbol{X}_n$ helps achieve desired rates of convergence, as we show next in Theorems 3 and 4.

Theorem 3: Let $\mathcal{X} = \prod_{i=1}^{d} \mathcal{X}_i$, where, for $i = 1, \ldots, d$, $\mathcal{X}_i$ are compact subsets of $\mathbb{R}$. For each $i \in \{1, \ldots, d\}$, let $x_{1i} < x_{2i} < \cdots < x_{\tilde{n}_i i}$ be an ordered set of points partitioning $\mathcal{X}_i$, with $h_i = \max_{1 \leq j \leq \tilde{n}_i - 1} (x_{\overline{j+1}i} - x_{ji})$. For $i = 1, \ldots, d$, and for $j = 1, \ldots, \tilde{n}_i$, let input points of the form $(x_1^*, \ldots, x_{i-1}^*, x_{ji}, x_{i+1}^*, \ldots, x_d^*)$ belong to $\boldsymbol{X}_n$, where $(x_1^*, \ldots, x_{i-1}^*, x_{i+1}^*, \ldots, x_d^*) \in \prod_{j \neq i} \mathcal{X}_j$ may be arbitrary.

For $\boldsymbol{x} = (x_1, \ldots, x_d)$ and $\boldsymbol{y} = (y_1, \ldots, y_d)$, let the correlation function be such that

$$\frac{\partial^4 c(\boldsymbol{x}^*, \boldsymbol{y}^*)}{\partial x_i \partial y_i \partial x_j \partial y_j} = \frac{\partial^4 c(\boldsymbol{x}, \boldsymbol{y})}{\partial x_i \partial y_i \partial x_j \partial y_j}\bigg|_{\boldsymbol{x}=\boldsymbol{x}^*, \boldsymbol{y}=\boldsymbol{y}^*}$$

exists for all $\boldsymbol{x}^*, \boldsymbol{y}^* \in \mathcal{X}$ and is Lipschitz continuous on $\mathcal{X} \times \mathcal{X}$ for $i, j = 1, \ldots, d$.

Then, letting $h = \max_{1 \leq i \leq d} h_i$, the following holds for almost all sequences $\{\boldsymbol{g}_n'(\cdot)\}_{n=1}^{\infty}$ with the above forms of the input points:

$$\sup_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{g}_n'(\boldsymbol{x}) - \boldsymbol{f}'(\boldsymbol{x})\|_d = O\left(\sqrt{h}\right), \text{ as } n \to \infty \text{ (equivalently, as } h \to 0). \tag{46}$$

The constant associated with $O\left(\sqrt{h}\right)$ depends only upon d and $f(\cdot)$.

Proof: For any $i \in \{1, \ldots, d\}$, and for arbitrary $\boldsymbol{X}_{-i}^* = (x_1^*, \ldots, x_{i-1}^*, x_{i+1}^*, \ldots, x_d^*)^T \in \prod_{j \neq i} \mathcal{X}_j$, for any $x \in \mathcal{X}_i$, let

$$g_{in}(x|\boldsymbol{X}_{-i}^*) = g_n(x_1^*, \ldots, x_{i-1}^*, x, x_{i+1}^*, \ldots, x_d^*); \tag{47}$$

$$f_i(x|\boldsymbol{X}_{-i}^*) = f(x_1^*, \ldots, x_{i-1}^*, x, x_{i+1}^*, \ldots, x_d^*). \tag{48}$$

Since $x \in \mathcal{X}_i$, it must belong to some interval of the form $[x_{ji}, x_{\overline{j+1}i}]$, for some $j \in \{1, 2, \ldots, \tilde{n}_i - 1\}$. Let us fix that j. For $y = x_{ji}$ and $y = x_{\overline{j+1}i}$, $g_{in}(y|\boldsymbol{X}_{-i}^*) = f_i(y|\boldsymbol{X}_{-i}^*)$ by interpolation property of the posterior Gaussian process, assuming that $(x_1^*, \ldots, x_{i-1}^*, y, x_{i+1}^*, \ldots, x_d^*) \in \boldsymbol{X}_n$ for $y = x_{ji}$ and $y = x_{\overline{j+1}i}$. That is, $g_{in}(y|\boldsymbol{X}_{-i}^*) - f_i(y|\boldsymbol{X}_{-i}^*) = 0$ for $y = x_{ji}$ and $y = x_{\overline{j+1}i}$. Hence, by Rolle's theorem, $g_{in}'(u|\boldsymbol{X}_{-i}^*) - f_i'(u|\boldsymbol{X}_{-i}^*) = 0$, for some $u \in \left(x_{ji}, x_{\overline{j+1}i}\right)$. This permits the following representation:

$$g_{in}'(x|\boldsymbol{X}_{-i}^*) - f_i'(x|\boldsymbol{X}_{-i}^*) = \int_u^x \left(g_{in}''(v|\boldsymbol{X}_{-i}^*) - f_i''(v|\boldsymbol{X}_{-i}^*)\right) dv, \tag{49}$$

The hypothesis of Lipschitz continuity of the 4-th order mixed partial derivatives of the correlation function ensures existence and sample path continuity of $g_{in}''(v|\boldsymbol{X}_{-i}^*)$. Hence, by the Cauchy-Schwartz inequality we obtain from (49), the following:

$$\begin{aligned}
&\left|g_{in}'(x|\boldsymbol{X}_{-i}^*) - f_i'(x|\boldsymbol{X}_{-i}^*)\right| \\
&\quad \leq \left[\int_u^x \left(g_{in}''(v|\boldsymbol{X}_{-i}^*) - f_i''(v|\boldsymbol{X}_{-i}^*)\right)^2 dv\right]^{1/2} \times |x - u|^{1/2} \\
&\quad \leq \left[\int_{\mathcal{X}_i} \left(g_{in}''(v|\boldsymbol{X}_{-i}^*) - f_i''(v|\boldsymbol{X}_{-i}^*)\right)^2 dv\right]^{1/2} \times h_i^{1/2} \\
&\quad \leq \sup_{\boldsymbol{X}_{-i}^* \in \prod_{j \neq i} \mathcal{X}_j} \left[\int_{\mathcal{X}_i} \left(g_{in}''(v|\boldsymbol{X}_{-i}^*) - f_i''(v|\boldsymbol{X}_{-i}^*)\right)^2 dv\right]^{1/2} \times h^{1/2}.
\end{aligned} \tag{50}$$

Now, since the hypotheses of this theorem constitute a special case of Theorem 2, the result of almost sure uniform boundedness of $|g''_{in}(\cdot)|$ as $n \to \infty$, is valid here. Specifically, from the proof of Theorem 2 in this case it holds that $\lim_{n\to\infty} \sup_{(u,\boldsymbol{X}^*_{-i})\in\mathcal{X}} |g''_{in}(v|\boldsymbol{X}^*_{-i})| < \infty$. This, along with continuity of $f''_i(v|\boldsymbol{X}^*_{-i})$ and compactness of $\mathcal{X}$, shows that the integral term is bounded by a constant C_i (depending only on f and d) almost surely for large n. Hence the right-hand side of (50) is $O(\sqrt{h})$.

Hence, switching to our usual notation, it follows that

$$\sup_{\boldsymbol{x}\in\mathcal{X}} \left(g'_{in}(\boldsymbol{x}) - f'_i(\boldsymbol{x})\right)^2 = O\left(h\right), \text{ almost surely, as } n \to \infty. \tag{51}$$

Since $\sup_{\boldsymbol{x}\in\mathcal{X}} \sum_{i=1}^d \left(g'_{in}(\boldsymbol{x}) - f'_i(\boldsymbol{x})\right)^2 \leq \sum_{i=1}^d \sup_{\boldsymbol{x}\in\mathcal{X}} \left(g'_{in}(\boldsymbol{x}) - f'_i(\boldsymbol{x})\right)^2$, it follows from (51) that

$$\sup_{\boldsymbol{x}\in\mathcal{X}} \|\boldsymbol{g}'_n(\boldsymbol{x}) - \boldsymbol{f}'(\boldsymbol{x})\|_d = O\left(\sqrt{h}\right), \text{ almost surely, as } n \to \infty \text{ (equivalently, as } h \to 0),$$

proving (46). Note that the constant associated with $O\left(\sqrt{h}\right)$ above depends only upon d and f. □

The following result holds as a consequence of Theorem 3.

Theorem 4: Under the conditions of Theorem 3, the following holds with the forms of the input points as specified in Theorem 3: for almost all sequences $\{g_n(\cdot)\}_{n=1}^\infty$,

$$\sup_{\boldsymbol{x}\in\mathcal{X}} |g_n(\boldsymbol{x}) - f(\boldsymbol{x})| = O\left(h^{3/2}\right), \text{ as } n \to \infty. \tag{52}$$

The constant associated with $O\left(h^{3/2}\right)$ depends only upon d and f.

Proof: As in the proof of Theorem 3, for any $x \in \mathcal{X}_i$, it must belong to some interval of the form $[x_{ji}, x_{\overline{j+1}i}]$, for some $j \in \{1, 2, \ldots, \tilde{n}_i - 1\}$. Let us fix that j. Now, $g_{in}(x_{ji}|\boldsymbol{X}^*_{-i}) = f_i(x_{ji}|\boldsymbol{X}^*_{-i})$ almost surely, by interpolation property of the posterior process $g_n(\cdot)$, assuming that $(x^*_1, \ldots, x^*_{i-1}, x_{ji}, x^*_{i+1}, \ldots, x^*_d) \in \boldsymbol{X}_n$. Hence,

$$g_{in}(x|\boldsymbol{X}^*_{-i}) - f_i(x|\boldsymbol{X}^*_{-i}) = \int_{x_{ji}}^x \left(g'_{in}(v|\boldsymbol{X}^*_{-i}) - f'_i(v|\boldsymbol{X}^*_{-i})\right) dv. \tag{53}$$

The Cauchy-Schwartz inequality applied to (53) gives

$$\left|g_{in}(x|\boldsymbol{X}^*_{-i}) - f_i(x|\boldsymbol{X}^*_{-i})\right| \leq \left[\int_{x_{ji}}^x \left(g'_{in}(v|\boldsymbol{X}^*_{-i}) - f'_i(v|\boldsymbol{X}^*_{-i})\right)^2 dv\right]^{1/2} \times |x - x_{ji}|^{1/2}. \tag{54}$$

From (50) it follows that the integral on the right hand side of (54) is $O\left(h\right) \times |x - x_{ji}|$, almost surely, as $n \to \infty$. Recall that the constant associated with $O\left(h\right) \times |x - x_{ji}|$ depends only upon d and f. Since $|x - x_{ji}|$ is bounded above by h, it follows that the right hand side of (54) is $O\left(h^{3/2}\right)$, almost surely, as $n \to \infty$. Switching to the usual notation, it is seen that (52) holds. □

Remark 2: As $h \to 0$, $\boldsymbol{g}'(\cdot)$ uniformly converges to $\boldsymbol{f}'(\cdot)$ at the rate $h^{1/2}$ and $g(\cdot)$ uniformly converges to $f(\cdot)$ at the rate $h^{3/2}$, almost surely with respect to their posteriors.

Remark 3: In Theorems 1, 2, 3 and 4 we have referred to the posterior (22), which corresponds to a linear mean structure of the Gaussian process prior and conjugate priors for $\boldsymbol{\beta}$ and σ^2. However, as can be seen from the proofs, both the theorems continue to hold for any mean function that has continuous second order mixed partial derivatives and any prior on the parameters, including the parameters of the correlation function such that the posteriors of the parameters are proper.

Remark 4: The hypotheses of Theorems 3 and 4 require input points of the form $(x_1^*, \ldots, x_{i-1}^*, x_{ji}, x_{i+1}^*, \ldots, x_d^*)$ to belong to $\boldsymbol{D}_n$ for $i = 1, \ldots, d$, and for $j = 1, \ldots, \tilde{n}_i$, where $(x_1^*, \ldots, x_{i-1}^*, x_{i+1}^*, \ldots, x_d^*) \in \prod_{j \neq i} \mathcal{X}_j$ may be chosen arbitrarily. Now observe that if $\mathcal{X}_i = [a, b]$, for $i = 1, \ldots, d$, for some $a < b$, then we can set $\tilde{n}_i = n$ and $h_i = h$, for $i = 1, \ldots, d$. In such cases, inclusion of the set of n^d points $\{(x_{j_1 1}, \ldots, x_{j_d d}) : j_1, \ldots, j_d \in \{1, \ldots, n\}\}$ in $\boldsymbol{D}_n$ is sufficient for Theorems 3 and 4 to hold. However, when d and n are even moderately large, n^d is an extremely large number, which would prohibit computation of Σ_{22}^{-1}, and hence computation of the posterior of $\boldsymbol{g}'(\cdot)$. Hence, for practical purposes it makes sense to refer to the general setup of Theorems 1 and 2.

5. Algorithm for optimization with the Gaussian process derivative method

We now propose a general methodology for function optimization, which judiciously exploits the posterior form (25). Without loss of generality, we consider the minimization problem for notational convenience. In a nutshell, the initial stage (say, the 0-th stage) of the methodology involves simulations from $\pi(\boldsymbol{x}^* | \boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}, \boldsymbol{D}_n)$ satisfying $\|\boldsymbol{f}'(\cdot)\|_d < \epsilon$ for some $\epsilon > 0$ and $\boldsymbol{\Sigma}''(\cdot) > 0$. In the subsequent stages $k = 1, 2, \ldots$, previous stage realizations satisfying $\|\boldsymbol{f}'(\cdot)\|_d < \eta_k$, where $\eta_k \to 0$ as $k \to \infty$, are successively augmented with $\boldsymbol{D}_n$ and realizations from the posterior associated with the augmented data are generated at each stage k by a judicious importance resampling strategy. As $k \to \infty$, the posteriors given the successively augmented data converge to the true optima.

The importance of the 0-th stage simulation algorithm for generating realizations from $\pi(\boldsymbol{x}^* | \boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}, \boldsymbol{D}_n)$ satisfying the restrictions $\|\boldsymbol{f}'(\cdot)\|_d < \epsilon$ for some $\epsilon > 0$ and $\boldsymbol{\Sigma}''(\cdot) > 0$, is enormous, particularly because the entire d-dimensional random variable $\boldsymbol{x}^*$ must be updated in a single block to meet the restrictions. Traditional MCMC algorithms are not known to be efficient in such problems as even for moderately large dimensions good proposal distributions are difficult to devise, and the acceptance rates can be poor, along with poor mixing properties. In this regard, the transformation based Markov Chain Monte Carlo (TMCMC) proposed by Dutta and Bhattacharya (2014) is an effective methodology. Indeed, TMCMC is designed to update all (or most of) the components of the high-dimensional random variable in a single block using appropriate deterministic transformations of some single (or low-dimensional) random variable. As such, this strategy drastically reduces effective dimensionality, which is responsible for maintaining good acceptance rates in spite of high dimensions. Good mixing properties can also be ensured by judiciously choosing the relevant "move-types", and judicious mixtures of additive and multiplicative transformations usually lead to desired mixing properties. For details on TMCMC and its properties, see Dutta and Bhattacharya (2014), Dey and Bhattacharya (2016), Dey and Bhattacharya (2017), Dey and

Bhattacharya (2019). As such, we recommend TMCMC for our optimization methodology. We provide the detailed Bayesian optimization methodology below as Algorithm 1.

A detailed analysis of the computational complexity of Algorithm 1 is presented in Appendix A. In summary, each TMCMC iteration (step (1)) requires $O(d^3 + d^2 n_0 + dn_0^2)$ operations, where d is the input dimension and n_0 the initial dataset size. The total complexity of step (1) is $O(N_{\text{TMCMC}} d^3)$ when d is moderate or large, with N_{TMCMC} the number of TMCMC iterations. The refinement stage (step (2)) adds $O(KN(d^3 + d^2 N_{\max} + dN_{\max}^2) + K_{\text{aug}} C_f)$ operations, where K is the number of stages, N the number of stored samples, $N_{\max}$ the maximum dataset size, K_{aug} the number of stages where augmentation occurs, and C_f the cost of evaluating the objective function f once. For the experimental settings in this paper, $C_f = O(d^2)$ and the TMCMC term $O(N_{\text{TMCMC}} d^3)$ dominates the runtime.

5.1. Practical guidance for Algorithm 1

To facilitate reproducibility and effective application of Algorithm 1, we provide the following practical recommendations.

5.1.1. Initial design

In all our experiments, we used a simple initial design consisting of $n = 10$ points uniformly spaced along the main diagonal of $\mathcal{X} = [-10, 10]^d$, that is, $\mathbf{x}_i = (x_i, \ldots, x_i)$ with $x_i = -10 + 2(i-1)$. This very sparse design suffices because the algorithm subsequently augments the dataset adaptively. For general applications, we recommend starting with a small number of points (for example, $n = 10$ to 20) chosen either along a diagonal or as a simple random sample; the algorithm is robust to the initial design as long as the domain is covered to some extent. More sophisticated space-filling designs (for example, Latin hypercube or Sobol sequences) may be used if n is kept small (for example, $n = 10d$) (for example, McKay *et al.* (1979), Sobol' (1967)), but we do not advocate dense grids because they would make the initial GP computations infeasible.

5.1.2. Choice of tolerance decay η_k

The sequence η_k should converge to zero slowly enough to maintain stable importance weights. In our experiments we used the following choices (with $k = 1, 2, \ldots$):

(a) For $d = 1, 2$: $\eta_k = 1/(10 + k - 1)^2$ (quadratic decay).

(b) For $d = 5$: $\eta_k = 1.5/\log(10 + k)$.

(c) For $d = 10$: $\eta_k = 7/\log(10 + k - 1)$.

(d) For $d = 50$: $\eta_k = 200/\log(10 + k - 1)$.

(e) For $d = 100$: $\eta_k = 850/\log(10 + k - 1)$.

In all cases $\eta_k \to 0$ and the decay is sufficiently slow to keep importance weights stable. For other problems, we recommend a logarithmic decay $\eta_k = C/\log(10 + k)$ for moderate

Algorithm 1 Optimization with Gaussian process derivatives

(1) First simulate N realizations $\{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_N^*\}$ from $\pi(\boldsymbol{x}^*|\boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}, \boldsymbol{D}_n)$ given by (25) using TMCMC, where the prior for $\boldsymbol{x}^*$ is given by (26) for some pre-fixed $\epsilon > 0$. This includes simulations from posteriors associated with most plausible functions $g(\cdot)$ satisfying $\boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}$ for $\boldsymbol{x}^* \in \mathcal{X}$, given $\boldsymbol{D}_n$. That is, $\{\boldsymbol{x}_i^*;\ i = 1, \ldots, N\}$, represents the set of solutions for $\boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}$ for functions $g(\cdot)$ that satisfy $g(\boldsymbol{x}_i) = f(\boldsymbol{x}_i)$; $i = 1, \ldots, n$. Thanks to the prior (26), these solutions further satisfy $\|\boldsymbol{f}'(\boldsymbol{x}_i^*)\|_d < \epsilon$ and $\boldsymbol{\Sigma}''(\boldsymbol{x}_i^*) > 0$, for $i = 1, \ldots, N$. Note that $\boldsymbol{\Sigma}''(\boldsymbol{x}^*) > 0$ can be checked by computing the eigenvalues and checking if all the eigenvalues are positive. But a more efficient alternative is to check if Cholesky decomposition of $\boldsymbol{\Sigma}''(\boldsymbol{x}^*)$ is possible, the information of which is provided by the subroutines of the BLAS and LAPACK libraries. We exploit the latter for our implementation.

(2) For stages $k = 1, 2, 3, \ldots$,

(i) For $i = 1, \ldots, N$, compute importance weights proportional to

$$w_k(\boldsymbol{x}_i^*) = \begin{cases} 1 & \text{if } k = 1; \\ w_{k-1}(\boldsymbol{x}_i^*) \times \frac{\pi(\boldsymbol{g}'(\boldsymbol{x}_i^*)=\boldsymbol{0}|\boldsymbol{D}_{n+\sum_{j=0}^{k-1} n_j}, \boldsymbol{x}_i^*)}{\pi(\boldsymbol{g}'(\boldsymbol{x}_i^*)=\boldsymbol{0}|\boldsymbol{D}_{n+\sum_{j=0}^{k-2} n_j}, \boldsymbol{x}_i^*)} & \text{if } k \geq 2, \end{cases} \tag{55}$$

where $n_0 = 0$.

(ii) Without replacement, select a subsample $\{\boldsymbol{x}_{i_1}^*, \ldots, \boldsymbol{x}_{i_M}^*\}$ from $\{\boldsymbol{x}_1^*, \ldots, \boldsymbol{x}_N^*\}$ with probabilities proportional to $w_k(\boldsymbol{x}_i^*)$; $i = 1, \ldots, N$. Note that, as $M \to \infty$ and $N \to \infty$ such that $M/N \to 0$, $\boldsymbol{x}_{i_j}^*$; $j = 1, \ldots, M$, follow the distribution $\pi(\boldsymbol{x}^*|\boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}, \boldsymbol{D}_{n+\sum_{j=1}^{k-1} n_j})$. The recursively computed importance weights $w_k(\boldsymbol{x}_i^*)$; $i = 1, \ldots, N$, are expected to be stable, since for each stage k, for $i = 1, \ldots, N$, the factors $\frac{\pi(\boldsymbol{g}'(\boldsymbol{x}_i^*)=\boldsymbol{0}|\boldsymbol{D}_{n+\sum_{j=1}^{k-1} n_j}, \boldsymbol{x}_i^*)}{\pi(\boldsymbol{g}'(\boldsymbol{x}_i^*)=\boldsymbol{0}|\boldsymbol{D}_{n+\sum_{j=1}^{k-2} n_j}, \boldsymbol{x}_i^*)}$ in (55) are not expected to be very different from 1 if n_{k-1} is not significantly greater than zero. Note that the importance weights $w_k(\boldsymbol{x}_i^*)$, for $i = 1, \ldots, N$, can be computed simultaneously on parallel processors. This, along with stability of the recursive formulation (55), is expected to make for an efficient computational strategy.

(iii) For $j = 1, \ldots, M$, check if $\|\boldsymbol{f}'(\boldsymbol{x}_{i_j}^*)\|_d < \eta_k$, where $\eta_k \to 0$ as $k \to \infty$. If $\boldsymbol{x}_{i_j}^*$ satisfies this condition, then $\boldsymbol{x}_{i_j}^*$ is a realization from $\pi(\boldsymbol{x}^*|\boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}, \boldsymbol{D}_{n+\sum_{j=1}^{k-1} n_j})$, where the prior for $\boldsymbol{x}^*$ is uniform on $B(\eta_k)$, the form of which is given by (27).

(iv) Let n_k (≥ 0) realizations among the M realizations satisfy the condition $\|\boldsymbol{f}'(\boldsymbol{x}_{i_j}^*)\|_d < \eta_k$. Without loss of generality, assume that $\boldsymbol{x}_{i_j}^*$; $j = 1, \ldots, n_k$ are such realizations. Compute $f(\boldsymbol{x}_{i_j}^*)$; $j = 1, \ldots, n_k$, and augment $(\boldsymbol{x}_{i_j}^*, f(\boldsymbol{x}_{i_j}^*)$; $j = 1, \ldots, n_k$, with $\boldsymbol{D}_{n+\sum_{j=0}^{k-1} n_j}$ to form $\boldsymbol{D}_{n+\sum_{j=0}^{k} n_j}$. In practice, to avoid excessive growth of the dataset and numerical instability, we recommend augmenting at most 5 points per stage (see also Remark 5).

(v) Store the realizations $\{\boldsymbol{x}_{i_j}^* :\ j = 1, \ldots, n_k\}$.

to high dimensions ($d \geq 5$), with C chosen such that the initial η_1 is about 10% of the typical gradient norm at random points (for example, C roughly proportional to d). For low dimensions ($d \leq 2$), a quadratic decay works well.

5.1.3. Stopping criteria

In our experiments, we simply fixed the number of refinement stages to $S = 40$ for dimensions $d = 2, 5, 10$ and $S = 100$ for $d = 50, 100$. In all cases, n_k became zero after a small number of stages (typically $k \leq 10$ for low dimensions, $k \leq 4$ for high dimensions). For general use, we recommend the following adaptive stopping criteria:

(a) Stop if $n_k = 0$ for three consecutive stages (no new points satisfy the gradient bound).

(b) Stop if the gradient norm $\|\mathbf{f}'(\mathbf{x}^*_{i_j})\|_d$ for the best candidate has not decreased by more than 1% over the last 10 stages.

(c) Stop after a maximum number of stages $K_{\max} = 100$ (safety bound).

With these criteria, the algorithm typically terminates after 10–40 stages in practice.

5.2. Pseudo-code for key components

To aid implementation, we provide pseudo-code for the TMCMC and the importance resampling components of Algorithm 1.

TMCMC (single iteration):

```
Input: current state x, target log-density L(x), scaling parameters a1, a2,
       move probabilities p, q (e.g., 0.5)
Output: next state x'

1. Draw U ~ Uniform(0,1)
2. if U < p:
      // Additive move
      epsilon ~ N(0,1)
      for j=1..d: b_j ~ Uniform({-1,1})
      y_j = x_j + b_j * a1 * |epsilon|
      accept with prob min(1, exp(L(y)-L(x)))
3. else:
      // Multiplicative move
      epsilon ~ Uniform(-1,1)
      for j=1..d: b_j ~ Uniform({-1,0,1})
      y_j = x_j * epsilon if b_j=1
          = x_j / epsilon if b_j=-1
          = x_j otherwise
      J = |epsilon|^(sum(b_j))
```

```
      accept with prob min(1, exp(L(y)-L(x)) * J)
4. Set x_tilde = accepted y or x
5. Draw U ~ Uniform(0,1)
6. if U < q:
      // Additive refinement
      epsilon ~ N(0,1)
      for j=1..d: y_j = x_tilde_j + a2 * |epsilon| with prob 0.5,
         else - a2 * |epsilon|
      accept with prob min(1, exp(L(y)-L(x_tilde)))
7. else:
      // Multiplicative refinement
      epsilon ~ Uniform(-1,1)
      if Uniform(0,1) < 0.5: y_j = x_tilde_j * epsilon, J = |epsilon|^d
      else: y_j = x_tilde_j / epsilon, J = |epsilon|^{-d}
      accept with prob min(1, exp(L(y)-L(x_tilde)) * J)
8. return accepted value
```

Importance resampling at stage k:

```
Input: prior samples {x_i} (i=1..N), current dataset D_{k-1}, target dataset
       D_k
Output: resampled indices {i_1,...,i_M}

1. For each i, compute w_i = w_{k-1}(x_i) * L_k(x_i) / L_{k-1}(x_i)
   where L_k(x) = pi(g'(x)=0 | D_k, x)
2. Normalize w_i to sum to 1
3. Sample M indices without replacement with probabilities w_i
4. Return the sampled indices
```

5.3. Further discussion of Algorithm 1

Algorithm 1 begins by simulating from the posterior of $\boldsymbol{x}^*$ satisfying $\|\boldsymbol{f}'(\boldsymbol{x}^*)\|_d < \epsilon$ and $\boldsymbol{\Sigma}''(\boldsymbol{x}^*) > 0$. In the subsequent steps $k \geq 1$, the set of realizations $\{\boldsymbol{x}^*_{i_j} : j = 1, \ldots, n_k\}$ generated by importance resampling further satisfy $\|\boldsymbol{f}'(\boldsymbol{x}^*_{i_j})\|_d < \eta_k$ along with $\boldsymbol{\Sigma}''(\boldsymbol{x}^*_{i_j}) > 0$, for $j = 1, \ldots, n_k$. The implication is that, ϵ may be chosen somewhat larger to achieve reasonably good TMCMC mixing and acceptance rates. Indeed, if n is not large enough, then $\boldsymbol{g}'$ is not expected to be sufficiently close to $\boldsymbol{f}'$, and hence for too small ϵ, $\{\boldsymbol{x}^* : \|\boldsymbol{f}'(\boldsymbol{x}^*)\|_d < \epsilon\}$ would be too small a region to contain the solutions $\{\boldsymbol{x}^* : \|\boldsymbol{g}'(\boldsymbol{x}^*)\|_d = 0\}$, given $\boldsymbol{D}_n$. This would result in poor TMCMC mixing.

Once adequate mixing with reasonable acceptance rates are achieved with relatively small n and relatively large ϵ, the subsequent steps increase the data size by augmenting the data with those values that satisfy $\|\boldsymbol{f}'(\boldsymbol{x}^*_{i_j})\|_d < \eta_k$. Thus, in the subsequent steps, these data points help better approximate the region around the stationary points of f by the posterior, and enables more simulations from the region $\|\boldsymbol{f}'(\boldsymbol{x}^*_{i_j})\|_d < \eta_k$, finally leading to convergence of the solutions $\{\boldsymbol{x}^* : \boldsymbol{g}'_k(\boldsymbol{x}^*) = \boldsymbol{0}, \boldsymbol{\Sigma}''(\boldsymbol{x}^*) > 0\}$ to $\{\boldsymbol{x}^* : \boldsymbol{f}'(\boldsymbol{x}^*) = \boldsymbol{0}, \boldsymbol{\Sigma}''(\boldsymbol{x}^*) > 0\}$, almost

surely, as $k \to \infty$, where $\boldsymbol{g}_k'(\cdot)$ denotes any realization from the posterior of $\boldsymbol{g}'(\cdot)$ given $\boldsymbol{D}_{n+\sum_{j=0}^{k-1} n_j}$. This intuition is formalized below as Theorem 5.

Theorem 5: Consider the setup of Theorem 2 (or more specifically, that of Theorem 3). Then, as $k \to \infty$, the set $\{\boldsymbol{x}_{i_j}^* : j = 1, \ldots, n_k\}$ of Algorithm 1 almost surely contains all the local minima of the objective function $f(\cdot)$, as $M \to \infty$ and $N \to \infty$ such that $M/N \to 0$.

Proof: Note that at stage k, as $M \to \infty$ and $N \to \infty$ such that $M/N \to 0$, $\boldsymbol{x}_{i_j}^*; j = 1, \ldots, n_k$, arise from $\pi(\boldsymbol{x}^* | \boldsymbol{g}'(\boldsymbol{x}^*) = \boldsymbol{0}, \boldsymbol{D}_{n+\sum_{j=0}^{k-1} n_j})$, subject to $\|\boldsymbol{f}'(\boldsymbol{x}_{i_j}^*)\|_d < \eta_k$ and $\boldsymbol{\Sigma}''(\boldsymbol{x}_{i_j}^*) > 0$ for $j = 1, \ldots, n_k$. These realizations are solutions of $\boldsymbol{g}_k'(\boldsymbol{x}^*) = \boldsymbol{0}$ and $\boldsymbol{\Sigma}''(\boldsymbol{x}^*) > 0$ when the data observed is $\boldsymbol{D}_{n+\sum_{j=0}^{k-1} n_j}$. By Theorem 2 (or more specifically by Theorem 3), as $k \to \infty$ (equivalently, as $h \to 0$ in Theorem 3), $\boldsymbol{g}_k'(\cdot)$ uniformly converges to $\boldsymbol{f}'(\cdot)$ almost surely. Hence, as $k \to \infty$,

$$\{\boldsymbol{x}^* : \boldsymbol{g}_k'(\boldsymbol{x}^*) = \boldsymbol{0}, \|\boldsymbol{f}'(\boldsymbol{x}^*)\|_d < \eta_k, \boldsymbol{\Sigma}''(\boldsymbol{x}^*) > 0\} \to \{\boldsymbol{x}^* : \boldsymbol{f}'(\boldsymbol{x}^*) = \boldsymbol{0}, \boldsymbol{\Sigma}''(\boldsymbol{x}^*) > 0\}, \quad (56)$$

almost surely.

Due to (56), as $k \to \infty$, the set $\{\boldsymbol{x}_{i_j}^* : j = 1, \ldots, n_k\}$ contains all the local minima of the objective function $f(\cdot)$, as $M \to \infty$ and $N \to \infty$ such that $M/N \to 0$. □

Remark 5: In step (2) (iv) of Algorithm 1 we have suggested augmentation of all realizations $(\boldsymbol{x}_{i_j}^*, f(\boldsymbol{x}_{i_j}^*)$ satisfying $\|\boldsymbol{f}'(\boldsymbol{x}_{i_j}^*)\|_d < \eta_k$ to the existing data $\boldsymbol{D}_{n+\sum_{j=0}^{k-1} n_j}$. In practice, augmentation of all such realizations may enlarge the dataset to such an extent that invertibility of the resultant Σ_{22} may be infeasible or numerically unstable, so that computation of the corresponding posterior densities of $\boldsymbol{g}'(\boldsymbol{x}_i^*) = \boldsymbol{0}$, and hence the importance weights (55), may not yield reliable results. Hence in practice, as a rule of thumb, we recommend augmenting at most 5 realizations satisfying $\|\boldsymbol{f}'(\boldsymbol{x}_{i_j}^*)\|_d < \eta_k$, which we consider in all our applications.

Remark 6: As the stage number k in step (2) of Algorithm 1 increases, n_k decreases. Hence, in practice, n_k will be zero after some large enough k. When d is large, due to the curse of dimensionality, only the first few stages are expect to yield positive n_k.

Remark 7: In step (1) of Algorithm 1, that is, in the TMCMC step, as well as in any stage k of step (2) of the algorithm, provided that n_k is sufficiently large, desired credible regions of the respective posterior distributions of $\boldsymbol{x}^*$ can be obtained. These quantify the stage-wise uncertainty in *a posteriori* learning about the optima, given $\|\boldsymbol{f}'(\cdot)\|_d < \epsilon$ or $\|\boldsymbol{f}'(\cdot)\|_d < \eta_k$. As $k \to \infty$, the uncertainty decreases, and the credibility regions shrink to the points representing the true optima. However, as mentioned in Remark 2, in practice, particularly for large d, n_k would be zero for most stages k, which would preclude computation of credible regions for most stages.

Remark 8: Note that if it is known beforehand that there is a single global minimum of $f(\cdot)$ on $\mathcal{X}$, then step (2) of Algorithm 1 is not required. It is then sufficient to report $\boldsymbol{x}_{i^*}^*$ as the (approximate) minimizer of $f(\cdot)$, where $i^* = \arg\min\{f(\boldsymbol{x}_i^*) : i = 1, \ldots, N\}$.

6. Bayesian characterization of the number of local minima of the objective function with recursive posteriors

Steps (2) (iii) and (2) (iv) can be combined to obtain a Bayesian characterization of the number of local minima of the objective function. In this regard, for stage j, let us define $Y_j = \sum_{r=1}^{M} I_{B(\eta_j)}(\boldsymbol{x}_{i_r}^*)$, where $B(\eta_j)$ is given by (27). Thus, $\{Y_j = m\}$ with probability p_{mj}, for $m = 0, 1, 2, \ldots, M$. Since $M \to \infty$, we allow Y_j to take values on the entire set of non-negative integers. That is, we set

$$P(Y_j = m) = p_{mj};\ m = 0, 1, 2, \ldots, \tag{57}$$

the infinite-dimensional multinomial distribution, where $0 \leq p_{mj} \leq 1$ for $m = 0, 1, 2, \ldots$ and $j \geq 1$. Further, $\sum_{m=0}^{\infty} p_{mj} = 1$ for all $j \geq 1$. We assume that the true probabilities $p_{m0} \in [0, 1]$; $m = 0, 1, 2, \ldots$, such that $\sum_{m=0}^{\infty} p_{m0} = 1$, are unknown. Indeed, if $f(\cdot)$ has finite number of local minima, then there must exist $\tilde{m} \geq 0$ such that $p_{\tilde{m}0} = 1$ and $p_{m0} = 0$ for $m \neq \tilde{m}$. For infinite number of local minima, we must have $p_{m0} = 0$ for any finite integer $m \geq 0$.

We adopt the approach of Roy and Bhattacharya (2020) based on Dirichlet process (Ferguson (1973)) to obtain the posterior distribution of the infinite set of parameters $\{p_{mk};\ m = 0, 1, 2, \ldots\}$. Roy and Bhattacharya (2020) came up with the following strategy for characterizing possibly infinite number of limit points of infinite series, which we usefully employ for our current purpose, albeit with a definition of Y_j that is different from that of Roy and Bhattacharya (2020).

Let $\mathcal{Y} = \{1, 2, \ldots\}$ and let $\mathcal{B}(\mathcal{Y})$ denote the Borel σ-field on $\mathcal{Y}$ (assuming every singleton of $\mathcal{Y}$ is an open set). Let $\mathcal{P}$ denote the set of probability measures on $\mathcal{Y}$. Then, at the first stage,

$$\pi(Y_1|P_1) \equiv P_1, \tag{58}$$

where $P_1 \in \mathcal{P}$. We assume that P_1 is the following Dirichlet process:

$$\pi(P_1) \equiv DP(G), \tag{59}$$

where, the probability measure G is such that, for every $j \geq 1$,

$$G(Y_j = m) = \frac{1}{2^m}. \tag{60}$$

Then

$$\pi(P_1|y_1) \equiv DP(G + \delta_{y_1}),$$

where, for any x, δ_x denotes point mass at x.

At the second stage, we set the prior for P_2 to be the posterior given y_1 corresponding to $DP\left(\left(1 + \frac{1}{2^2}\right)G\right)$ prior for P_1. That is, $\pi(P_2) \equiv DP\left((1 + \frac{1}{2^2})G + \delta_{y_1}\right)$. Then, with respect to this prior for P_2, the posterior of P_2 is given by

$$\pi(P_2|y_2) \equiv DP\left(\left(1 + \frac{1}{2^2}\right)G + \delta_{y_1} + \delta_{y_2}\right).$$

Continuing this recursive process we obtain that, at the k-th stage, the posterior of P_k to be a Dirichlet process, given by

$$\pi\left(P_k|y_k\right) \sim DP\left(\sum_{j=1}^{k}\frac{1}{j^2}G + \sum_{j=1}^{k}\delta_{y_j}\right). \tag{61}$$

It follows from (61) that

$$E\left(p_{mk}|y_k\right) = \frac{\frac{1}{2^m}\sum_{j=1}^{k}\frac{1}{j^2} + \sum_{j=1}^{k} I\left(y_j = m\right)}{\sum_{j=1}^{k}\frac{1}{j^2} + k}; \tag{62}$$

$$Var\left(p_{mk}|y_k\right) = \frac{\left(\sum_{j=1}^{k}\frac{1}{j^2} + \sum_{j=1}^{k} I\left(y_j = m\right)\right)\left(\left(1-\frac{1}{2^m}\right)\sum_{j=1}^{k}\frac{1}{j^2} + k - \sum_{j=1}^{k} I\left(y_j = m\right)\right)}{\left(\sum_{j=1}^{k}\frac{1}{j^2} + k\right)^2\left(\sum_{j=1}^{k}\frac{1}{j^2} + k + 1\right)}. \tag{63}$$

The theorem below characterizes the number of local minima of the objective function $f(\cdot)$ in terms of the limit of the marginal posterior probabilities of p_{mk}, denoted by $\pi_m\left(\cdot|y_k\right)$, as $k \to \infty$.

Theorem 6: Assume the conditions of Theorem 3, and in Algorithm 1, assume that $M \to \infty$ and $N \to \infty$ such that $M/N \to 0$. Then $f(\cdot)$ has $\tilde{m}$ (≥ 0) local minima if and only if

$$\pi_{\tilde{m}}\left(\mathcal{N}_1|y_k\right) \to 1, \text{ almost surely with respect to the posterior (25)}, \tag{64}$$

as $k \to \infty$. In the above, $\mathcal{N}_1$ is any neighborhood of 1 (one).

Proof: First, let us assume that $f(\cdot)$ has $\tilde{m}$ (≥ 0) local minima. Then, by Theorem 5, Algorithm 1 converges to the $\tilde{m}$ local minima almost surely, as $k \to \infty$, provided that $M \to \infty$ and $N \to \infty$ such that $M/N \to 0$. Hence, there exists $j_0 \geq 1$, such that for $j \geq j_0$, $y_j = \tilde{m}_0$. Thus, almost surely, $I(y_j = \tilde{m}) = 1$, for $j \geq j_0$. Consequently, it easily follows from (62) and (63) that almost surely, as $k \to \infty$,

$$E\left(p_{\tilde{m}k}|y_k\right) \to 1, \quad \text{and} \tag{65}$$

$$Var\left(p_{\tilde{m}k}|y_k\right) = O\left(\frac{1}{k}\right) \to 0. \tag{66}$$

Now let $\mathcal{N}_1$ denote any neighborhood of 1, and let ϵ (> 0) be sufficiently small such that $\mathcal{N}_1 \supseteq \{1 - p_{\tilde{m}k} < \epsilon\}$. Then using Markov's inequality we obtain

$$\begin{aligned}
\pi_{\tilde{m}}\left(\mathcal{N}_1|y_k\right) &\geq \pi_{\tilde{m}}\left(1 - p_{\tilde{m}k} < \epsilon|y_k\right) \\
&= 1 - \pi_{\tilde{m}}\left(1 - p_{\tilde{m}k} \geq \epsilon|y_k\right) \\
&\geq 1 - \frac{E\left(1 - p_{\tilde{m}k}|y_k\right)^2}{\epsilon^2} \\
&= 1 - \frac{1 - 2E\left(p_{\tilde{m}k}|y_k\right) + E\left(p_{\tilde{m}k}^2|y_k\right)}{\epsilon^2}.
\end{aligned} \tag{67}$$

Now, as $k \to \infty$, $E\left(p_{\tilde{m}k}|y_k\right) \to 1$ by (65), and $E\left(p^2_{\tilde{m}k}|y_k\right) = Var\left(p_{\tilde{m}k}|y_k\right) + \left[E\left(p_{\tilde{m}k}|y_k\right)\right]^2 \to 1$ by (65) and (66). Hence, the right hand side of (67) converges to 1 almost surely, as $k \to \infty$. This proves (64).

Now assume that (64) holds for any neighborhood $\mathcal{N}_1$ of 1. Let us fix $\eta \in (0,1)$. Then given any $\epsilon \in (0, 1-\eta)$,

$$\pi_{\tilde{m}}\left(1 - p_{\tilde{m}k} < \epsilon | y_k\right) \to 1, \tag{68}$$

almost surely as $k \to \infty$. The left hand side of (68) admits the following Markov's inequality:

$$\pi_{\tilde{m}}\left(1 - p_{\tilde{m}k} < \epsilon | y_k\right) = \pi_{\tilde{m}}\left(p_{\tilde{m}k} > 1 - \epsilon | y_k\right) < \frac{E\left(p_{\tilde{m}k}|y_k\right)}{1-\epsilon}. \tag{69}$$

Due to (68), validity of (69) for all $\epsilon \in (0, 1-\eta)$, and almost sure upper boundedness of $p_{\tilde{m}k}$ by 1, it follows that

$$E\left(p_{\tilde{m}k}|y_k\right) \to 1, \text{ almost surely, as } k \to \infty. \tag{70}$$

Now, if $f(\cdot)$ is assumed to have m^* local minima where $m^* \neq \tilde{m}$, then due to (65) we must have

$$E\left(p_{m^*k}|y_k\right) \to 1, \text{ almost surely, as } k \to \infty. \tag{71}$$

Also note that since $0 \leq p_{mk} \leq 1$ for all m and k, the dominated convergence theorem ensures the following, almost surely:

$$1 = \lim_{k\to\infty} \sum_{m=1}^{\infty} E\left(p_{mk}|y_k\right) = \sum_{m=1}^{\infty} \lim_{k\to\infty} E\left(p_{mk}|y_k\right),$$

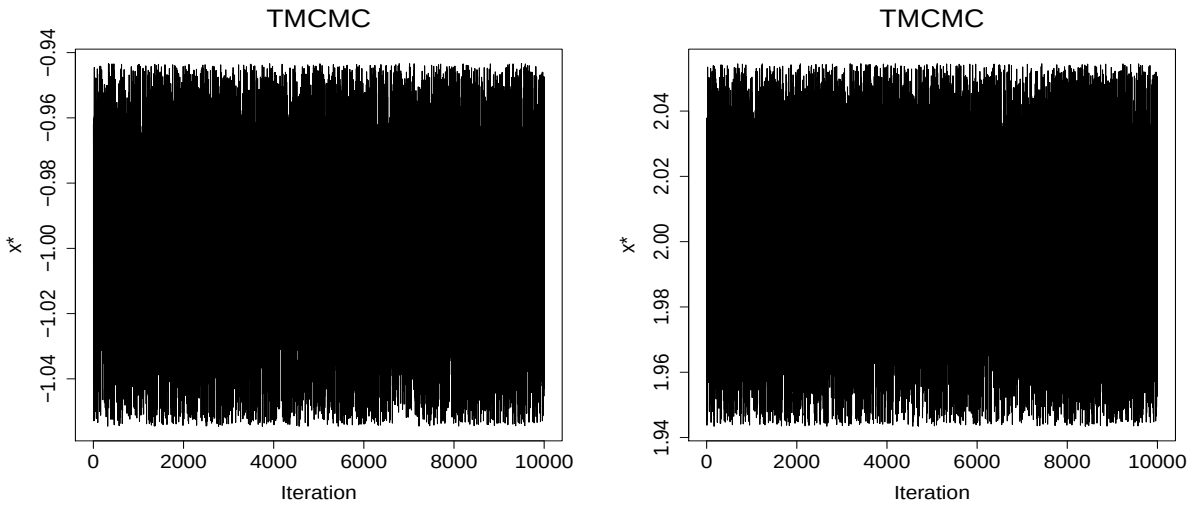
which implies, due to (71), that $E\left(p_{\tilde{m}k}|y_k\right) \to 0$, almost surely, as $k \to \infty$. But this would contradict (70). Hence, $f(\cdot)$ must have $\tilde{m}$ local minima. □

7. Experiments

We consider application of Algorithm 1 to 5 different optimization problems, ranging from simple to challenging, several of which are problems of finding both maxima and minima, and one is concerned with saddle points and inconclusiveness in addition to maximum. Encouragingly, all our experiments bring forth the versatility of Algorithm 1 in capturing all the optima, saddlepoints, as well as inconclusiveness of the problems.

As regards TMCMC, in our examples, we expectedly find that in the less challenging, low-dimensional problems, additive TMCMC is sufficient, while in the more challenging cases, we consider appropriate mixtures of additive and multiplicative moves, followed by a further move of a specialized mixture of additive and multiplicative transformations to improve mixing.

In most of our experiments, we discard the first 10^5 TMCMC iterations as burn-in and store every 10-th TMCMC realization in the next 5×10^5 iterations, to obtain 50000 realizations before proceeding to step (2) of Algorithm 1. However, in the 4-th example, due to inadequate mixing, we discard the first 10^6 iterations and store every 10-th realization in the next 5×10^6 iterations to obtain 5×10^5 TMCMC realizations. For the resample size in step (2), we set $M = 1000$.



(a) TMCMC for maximum. (b) TMCMC for minimum.

Figure 7.1: TMCMC trace plots for Example 1.

For the posterior of $\boldsymbol{g}'(\cdot) = \mathbf{0}$ given by (23), we set $a = b = 0.1$, $\boldsymbol{\beta}_0 = \mathbf{0}$ and $\boldsymbol{\Sigma}_0 = \mathbb{I}_d$, for all the examples, where d is the dimension relevant to the problem. In all the examples, we also set $\mathcal{X} = [-10, 10]^d$ and the initial input size $n = 10$. With $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{id})$ for $i = 1, \ldots, n$, we choose the inputs points as $x_{ik} = -10 + 2(i-1)$, for $i = 1, \ldots, n$ and for $k = 1, \ldots, d$. We then evaluate $f(\cdot)$ at each $\boldsymbol{x}_i$; $i = 1, \ldots, n$, to form $\boldsymbol{D}_n$ with $n = 10$. As we shall demonstrate, this strategy, in conjunction with the rest of the methodology, leads to adequate estimation of the optima in our examples.

7.1. Example 1

We begin with a simple example, where the goal is to obtain the maxima and minima of the function

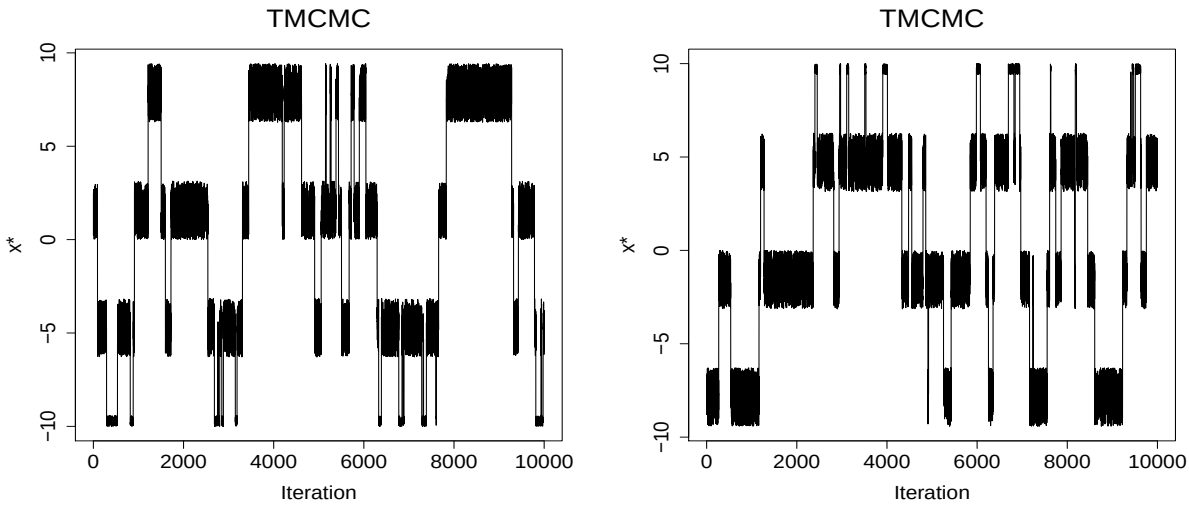
$$f(x) = 2x^3 - 3x^2 - 12x + 6. \tag{72}$$

Here $f'(x) = 6(x-2)(x+1)$ and $f''(x) = 6(2x-1)$. Hence, this function has a maximum at $x = -1$ and minimum at $x = 2$.

We apply step (1) of Algorithm 1 with $\epsilon = 1$, implementing additive TMCMC with equal move-type probabilities for forward and backward transformations. Specifically, at iteration $t = 1, 2, \ldots$, letting $x^{(t-1)}$ denote the TMCMC realization at iteration $t-1$, we draw $\varepsilon \sim N(0, 1)$ and consider the transformation $y = x^{(t-1)} + b|\varepsilon|$, where b takes the values 1 and -1 with equal probabilities. We set $x^{(t)} = y$ with probability

$$\alpha = \min\left\{1, \frac{\pi(y)\pi(g'(y) = 0|\boldsymbol{D}_n, y)}{\pi(x^{(t-1)})\pi(g'(x^{(t-1)}) = 0|\boldsymbol{D}_n, x^{(t-1)})}\right\},$$

and set $x^{(t)} = x^{(t-1)}$ with the remaining probability. This is of course same as the ordinary random walk Metropolis algorithm since the dimension here is $d = 1$; see Dutta and Bhattacharya (2014) for ramifications and detailed discussions. Figure 7.1 shows the trace plots of x^* for maximum and minimum, where to reduce the figure file size, we thinned the original TMCMC sample of size $N = 50000$ to 10000 by displaying every 5-th realization. The trace



(a) TMCMC for maximum. (b) TMCMC for minimum.

Figure 7.2: TMCMC trace plots for Example 2.

plots indicate adequate mixing properties of TMCMC. We run step (2) of Algorithm 1 for $k = 1, \ldots, S$ stages with $S = 40$, setting $\eta_k = 1/(10 + k - 1)^2$, computing the importance weights for the N TMCMC realizations at each stage on 100 64-bit cores in a VMWare parallel computing environment. The cores have 2.80 GHz speed, and have access to 1 TB memory. All our codes are written in C, using the Message Passing Interface (MPI) protocol for parallel processing. As such, our entire exercise is completed in about 2 minutes. We obtain $\hat{x}_{\max} = -0.999995$ and $\hat{x}_{\min} = 2.000023$, as our estimates of the maximum and the minimum, respectively, which are quite accurate.

7.2. Example 2

We now consider maximization and minimization of the function $f(x) = \sin(x)$ for $x \in [-10, 10]$. Here, the true maxima are $x_{\max} = \{\frac{\pi}{2} - 2\pi = -4.712389, \frac{\pi}{2} = 1.570796, \frac{\pi}{2} + 2\pi = 7.853982\}$, and the minima are $x_{\min} = \{-7.853982, -1.570796, 4.712389\}$.

We implement Algorithm 1 in the same way as in Example 1. Figure 7.2 displays the TMCMC trace plots, thinned to 10000 realizations to reduce figure file sizes. Clear tri-modality can be visualized from both the trace plots. After implementing step (2) of Algorithm 1 for $S = 40$ stages on our VMWare parallel computing architecture, we obtain $\hat{x}_{\max} = \{-4.712581, 1.570655, 7.860879\}$ and $\hat{x}_{\min} = \{-7.854051, -1.570423, 4.713222\}$, which turn out to be adequate approximations to the truths. Again, the entire exercise takes about 2 minutes.

7.3. Example 3

Let us now consider a two-dimensional example, given by $f(x_1, x_2) = x_1 x_2 (x_1 + x_2)(1 + x_2)$.

The first derivatives are given by $f_1'(x_1, x_2) = x_2(2x_1 + x_2)(x_2 + 1)$ and $f_2'(x_1, x_2) =$

$x_1(3x_2^2 + 2x_2(x_1 + 1) + x_1)$.

The second derivatives are $f''_{11}(x_1, x_2) = 2x_2(x_2 + 1)$, $f''_{12}(x_1, x_2) = f''_{21}(x_1, x_2) = 4x_1x_2 + 3x_2^2 + 2(x_1 + x_2)$ and $f''_{22}(x_1, x_2) = 2x_1(3x_2 + x_1 + 1)$.

Consider the determinant $D(x_1, x_2) = f''_{11}(x_1, x_2)f''_{22}(x_1, x_2) - [f''_{12}(x_1, x_2)]^2$. Now, if (a, b) is any critical point of $f(\cdot)$ satisfying $f'_1(a, b) = 0$ and $f'_2(a, b) = 0$, then (a, b) is a local maximum if $D(a, b) > 0$ and $f''_{11}(a, b) < 0$; (a, b) is a local minimum if $D(a, b) > 0$ and $f''_{11}(a, b) > 0$; (a, b) is a saddle point if $D(a, b) < 0$. Furthermore, if $D(a, b) = 0$, then (a, b) may be either maximum, minimum or even a saddle point, that is, the derivative test remains inconclusive in such cases.

In this example, it is easy to verify that there are four critical points $(0, 0)$, $(0, -1)$, $(1, -1)$ and $\left(\frac{3}{8}, -\frac{3}{4}\right)$. The last point is a local maximum; $(0, -1)$ and $(1, -1)$ are saddle points, and the derivative test remains inconclusive about $(0, 0)$.

We implement Algorithm 1 using the above conditions on $D(\boldsymbol{x}^*)$ and $f''_{11}(\boldsymbol{x}^*) > 0$, along with the condition $\|\boldsymbol{f}'(\boldsymbol{x}^*)\|_2 < \epsilon$, with $\epsilon = 1$, in the prior for $\boldsymbol{x}^* = (x_1^*, x_2^*)$, for detection of maxima, minima, saddle points and inconclusiveness.

We implement additive TMCMC with equal move-type probabilities for forward and backward transformations. In these cases, at iteration $t = 1, 2, \ldots$, letting $\boldsymbol{x}^{(t-1)}$ denote the TMCMC realization at iteration $t - 1$, we draw $\varepsilon \sim N(0, 1)$ and consider the transformation $\boldsymbol{y} = \boldsymbol{x}^{(t-1)} + \boldsymbol{b}|\varepsilon|$, where, $\boldsymbol{b} = (b_1, b_2)^T$ and each of b_1 and b_2 independently takes the values 1 and -1 with equal probabilities. We set $\boldsymbol{x}^{(t)} = \boldsymbol{y}$ with probability

$$\alpha = \min\left\{1, \frac{\pi(\boldsymbol{y})\pi(\boldsymbol{g}'(\boldsymbol{y}) = \boldsymbol{0}|\boldsymbol{D}_n, \boldsymbol{y})}{\pi(\boldsymbol{x}^{(t-1)})\pi(\boldsymbol{g}'(\boldsymbol{x}^{(t-1)}) = \boldsymbol{0}|\boldsymbol{D}_n, \boldsymbol{x}^{(t-1)})}\right\},$$

and set $\boldsymbol{x}^{(t)} = \boldsymbol{x}^{(t-1)}$ with the remaining probability. Note that unlike the one-dimensional setup, this additive TMCMC is no longer equivalent to random walk Metropolis (see Dutta and Bhattacharya (2014) for details).

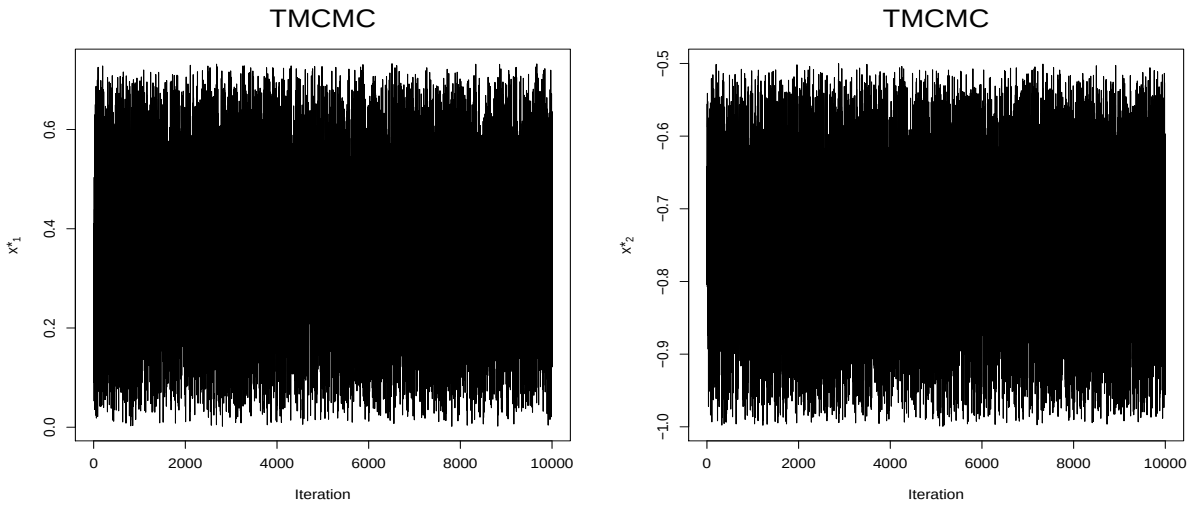
Implementation of Algorithm 1 for obtaining the maximum and the saddle points took about 7 minutes to complete on our VMWare, implemented on 100 cores.

7.3.1. Maximum

Figure 7.3 displaying the TMCMC trace plots for maxima finding, indicates quite adequate mixing. Running step (2) of Algorithm 1 for $S = 40$ steps yields $\hat{\boldsymbol{x}}_{max} = (0.376858, -0.752406)$, which is reasonably close to the true maximum.

7.3.2. Saddle points

Our initial TMCMC investigations revealed two modal regions roughly around $(0.1, -1.1)$ and $(1.1, -1.1)$. For better exploration of the two modal regions, we implemented two separate TMCMC runs beginning at the above two points, and continued Algorithm 1 to ultimately obtain two separate results after $S = 40$ stages at step (2), associated with the two different starting points. Figures 7.4 and 7.5 display the TMCMC trace plots associated with the two different starting points.



(a) TMCMC for maximum: first co-ordinate. (b) TMCMC for maximum: second co-ordinate.

Figure 7.3: TMCMC trace plots for Example 3 for finding maxima.

Running step (2) of Algorithm 1 for $S = 40$ steps yields $\hat{\boldsymbol{x}}^{(1)}_{saddle} = (-0.000309, -0.999580)$ and $\hat{\boldsymbol{x}}^{(2)}_{saddle} = (0.999433, -0.999915)$ as estimates of two saddle points, which are both reasonably close to the true saddle points.

7.3.3. Inconclusiveness

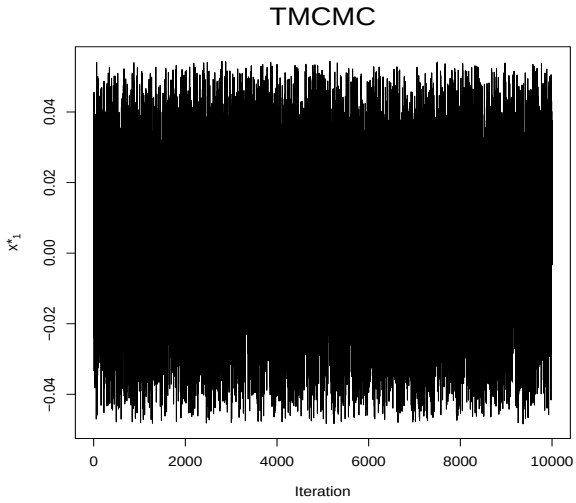
Investigation of situations where $D(a, b) = 0$ for any critical point (a, b) yielded the TMCMC trace plots displayed as Figure 7.6. On completion of step (2) of Algorithm 1, we obtain $\hat{\boldsymbol{x}}_{incon} = (-0.000087, -0.000265)$ as the estimate of the stationary point regarding which conclusion can not be drawn using the second derivatives. Observe that $\hat{\boldsymbol{x}}_{incon}$ quite adequately estimates the actual point $(0, 0)$ where conclusion fails.

7.3.4. Minimum

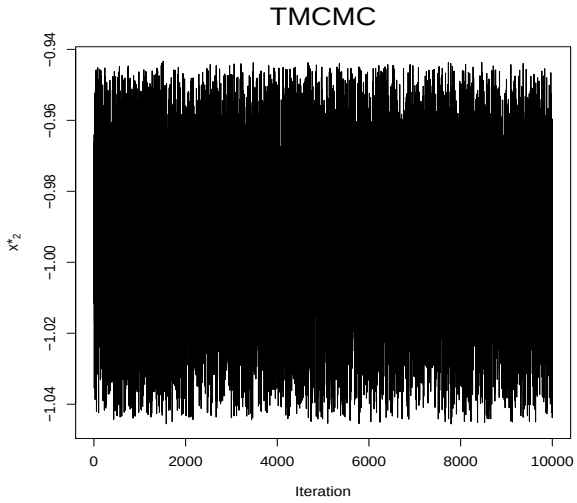
Our attempt to implement TMCMC with the restrictions $D(\boldsymbol{x}^*) > 0$ and $f''_{11}(\boldsymbol{x}^*) > 0$ with $\|\boldsymbol{f}'(\boldsymbol{x}^*)\|_2 < \epsilon$ did not yield any acceptance, even for arbitrary initial values. In other words, we could not obtain any solution that satisfies all the above restrictions, and hence conclude that there is no critical point on $\mathcal{X}$ that satisfy the above restrictions.

7.4. Example 4

Table 14.2 of Lange (2010) reports quarterly data on AIDS deaths in Australia during $1983 - 1986$, which is considered for illustration of fitting Poisson regression model. Specifically, for $i = 1, \ldots, 14$, Lange (2010) considers the model $Y_i \sim Poisson(\lambda_i)$, with $\lambda_i = \exp(\beta_0 + i\beta_1)$. Lange (2010) computed the maximum likelihood estimate (MLE) of $\boldsymbol{\beta} = (\beta_0, \beta_1)$ using Fisher's scoring method, which is equivalent to Newton's method in this

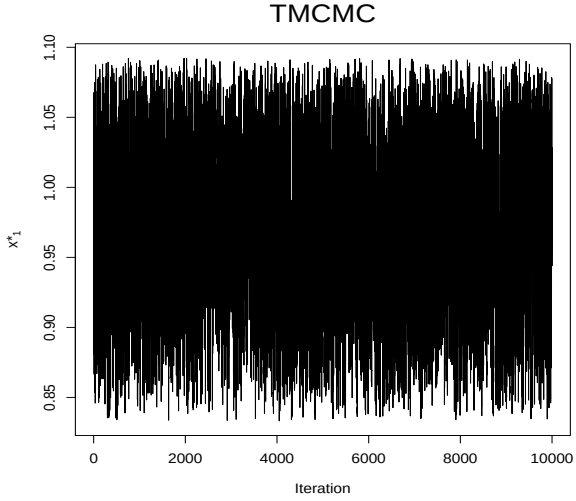


(a) TMCMC for first saddle point: first coordinate.

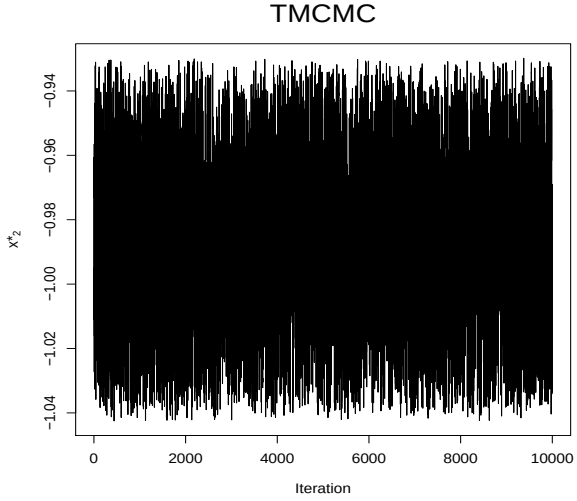


(b) TMCMC for first saddle point: second coordinate.

Figure 7.4: TMCMC trace plots for Example 3 for finding the first saddle point.

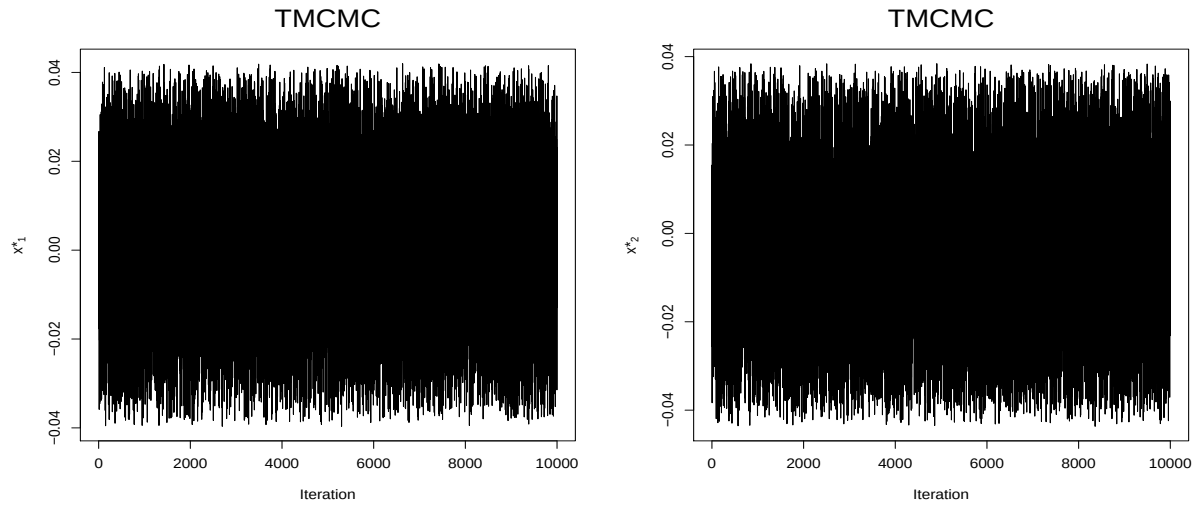


(a) TMCMC for second saddle point: first coordinate.



(b) TMCMC for second saddle point: second coordinate.

Figure 7.5: TMCMC trace plots for Example 3 for finding the second saddle point.



(a) TMCMC for inconclusiveness: first coordinate.

(b) TMCMC for inconclusiveness: second coordinate.

Figure 7.6: TMCMC trace plots for Example 3 for investigating inconclusiveness.

case. The final estimate obtained by Lange (2010) is $\hat{\boldsymbol{\beta}}_{MLE} = (0.3396, 0.2565)$.

In our notation, the function to maximize is $f(x_1, x_2) = -\sum_{i=1}^{14} \exp(x_1 + ix_2) + \sum_{i=1}^{14} y_i(x_1 + ix_2)$, with respect to $\boldsymbol{x} = (x_1, x_2)$. Note that this is a concave maximization problem and hence the second derivative is irrelevant. We thus consider the only constraint $\|\boldsymbol{f}'(\boldsymbol{x}^*)\|_2 < \epsilon$ for our implementation, with $\epsilon = 1$. However, additive TMCMC did not exhibit adequate mixing properties in this example, and hence we consider a mixture of additive and multiplicative TMCMC that is expected to improve mixing by using a mixture of localised moves of additive TMCMC and non-local ("random dive") moves of multiplicative TMCMC (see Dutta (2012), Dey and Bhattacharya (2016) for details). We strengthen the mixture TMCMC strategy with a further step of a mixture of specialized additive and multiplicative moves, which has parallels with Liu and Sabatti (2000). The detailed TMCMC algorithm, for general dimension d, is provided below as Algorithm 2.

In our case, $d = 2$, and we choose $a_j^{(1)} = a_j^{(2)} = 0.05$, for $j = 1, 2$; we also set $p = q = 1/2$. However, in spite of such a sophisticated TMCMC algorithm, we failed to achieve excellent mixing in this example, even with long TMCMC runs, discarding the first 10^6 iterations and storing every 10-th realization in the next 5×10^6 iterations. The trace plots of all stored 5×10^5 realizations shown in Figure 7.7 indeed demonstrate that the TMCMC chain does not have excellent mixing properties. In fact, the trace plots correspond to a reasonable initial value, chosen as the optimum obtained by the Fisher scoring method, which we implemented in R independently of Lange (2010). However, our goal here is to demonstrate that even when TMCMC mixing is less than ideal, the estimates obtained by our method can still significantly outperform existing techniques.

Since this is a concave maximization problem, step (2) of Algorithm 1 is unnecessary. Following Remark 8, we set $i^* = \arg\min\{f(\mathbf{x}_i^*) : i = 1, \ldots, N\}$ and report $\mathbf{x}_{i^*}^*$ as the approximate maximizer of $f(\cdot)$. With the $N = 5 \times 10^5$ TMCMC realizations shown in

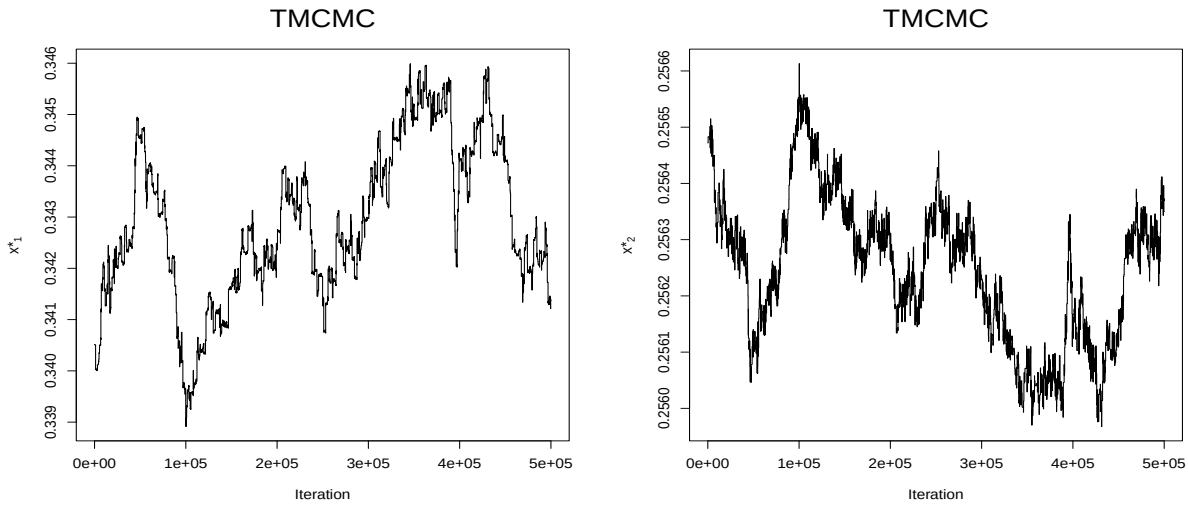
Algorithm 2 Mixture TMCMC

(1) Fix $p, q \in (0,1)$. Set an initial value $\boldsymbol{x}^{(0)}$.

(2) For $t = 1, \ldots, N$, do the following:

1. Generate $U \sim U(0,1)$.
   (a) If $U < p$, then do the following:
      (i) Generate $\varepsilon \sim N(0,1)$, $b_j \overset{iid}{\sim} U(\{-1,1\})$ for $j = 1, \ldots, d$, and set $y_j = x_j^{(t-1)} + b_j a_j^{(1)} |\varepsilon|$, for $j = 1, \ldots, d$. Here $a_j^{(1)}$ are positive scaling constants.
      (ii) Evaluate
$$\alpha_1 = \min\left\{1, \frac{\pi(\boldsymbol{y})\pi(\boldsymbol{g}'(\boldsymbol{y}) = \boldsymbol{0}|\boldsymbol{D}_n, \boldsymbol{y})}{\pi(\boldsymbol{x}^{(t-1)})\pi(\boldsymbol{g}'(\boldsymbol{x}^{(t-1)}) = \boldsymbol{0}|\boldsymbol{D}_n, \boldsymbol{x}^{(t-1)})}\right\}.$$
      (iii) Set $\tilde{\boldsymbol{x}}^{(t)} = \boldsymbol{y}$ with probability α_1, else set $\tilde{\boldsymbol{x}}^{(t)} = \boldsymbol{x}^{(t-1)}$.
   (b) If $U \geq p$, then
      (i) Generate $\varepsilon \sim U(-1,1)$, $b_j \overset{iid}{\sim} U(\{-1,0,1\})$ for $j = 1, \ldots, d$, and set $y_j = x_j^{(t-1)}\varepsilon$ if $b_j = 1$, $y_j = x_j^{(t-1)}/\varepsilon$ if $b_j = -1$ and $y_j = x_j^{(t-1)}$ if $b_j = 0$, for $j = 1, \ldots, d$. Calculate $|J| = |\varepsilon|^{\sum_{j=1}^d b_j}$.
      (ii) Evaluate
$$\alpha_2 = \min\left\{1, \frac{\pi(\boldsymbol{y})\pi(\boldsymbol{g}'(\boldsymbol{y}) = \boldsymbol{0}|\boldsymbol{D}_n, \boldsymbol{y})}{\pi(\boldsymbol{x}^{(t-1)})\pi(\boldsymbol{g}'(\boldsymbol{x}^{(t-1)}) = \boldsymbol{0}|\boldsymbol{D}_n, \boldsymbol{x}^{(t-1)})} \times |J|\right\}.$$
      (iii) Set $\tilde{\boldsymbol{x}}^{(t)} = \boldsymbol{y}$ with probability α_2, else set $\tilde{\boldsymbol{x}}^{(t)} = \boldsymbol{x}^{(t-1)}$.
2. Generate $U \sim U(0,1)$.
   (a) If $U < q$, then do the following
      (i) Generate $\tilde{U} \sim U(0,1)$ and $\varepsilon \sim N(0,1)$. If $\tilde{U} < 1/2$, set $y_j = \tilde{x}_j^{(t)} + a_j^{(2)}|\varepsilon|$, for $j = 1, \ldots, d$; else, set $y_j = \tilde{x}_j^{(t)} - a_j^{(2)}|\varepsilon|$, for $j = 1, \ldots, d$. Here $a_j^{(2)}$ are positive scaling constants.
      (ii) Evaluate
$$\alpha_3 = \min\left\{1, \frac{\pi(\boldsymbol{y})\pi(\boldsymbol{g}'(\boldsymbol{y}) = \boldsymbol{0}|\boldsymbol{D}_n, \boldsymbol{y})}{\pi(\tilde{\boldsymbol{x}}^{(t)})\pi(\boldsymbol{g}'(\tilde{\boldsymbol{x}}^{(t)}) = \boldsymbol{0}|\boldsymbol{D}_n, \tilde{\boldsymbol{x}}^{(t)})}\right\}.$$
      (iii) Set $\boldsymbol{x}^{(t)} = \boldsymbol{y}$ with probability α_3, else set $\boldsymbol{x}^{(t)} = \tilde{\boldsymbol{x}}^{(t)}$.
   (b) If $U \geq q$, then
      (i) Generate $\varepsilon \sim U(-1,1)$ and $\tilde{U} \sim U(0,1)$. If $\tilde{U} < 1/2$, set $y_j = \tilde{x}_j^{(t)}\varepsilon$ for $j = 1, \ldots, d$ and $|J| = |\varepsilon|^d$, else set $y_j = \tilde{x}_j^{(t)}/\varepsilon$ for $j = 1, \ldots, d$ and $|J| = |\varepsilon|^{-d}$.
      (ii) Evaluate
$$\alpha_4 = \min\left\{1, \frac{\pi(\boldsymbol{y})\pi(\boldsymbol{g}'(\boldsymbol{y}) = \boldsymbol{0}|\boldsymbol{D}_n, \boldsymbol{y})}{\pi(\tilde{\boldsymbol{x}}^{(t)})\pi(\boldsymbol{g}'(\tilde{\boldsymbol{x}}^{(t)}) = \boldsymbol{0}|\boldsymbol{D}_n, \tilde{\boldsymbol{x}}^{(t)})} \times |J|\right\}.$$
      (iii) Set $\boldsymbol{x}^{(t)} = \boldsymbol{y}$ with probability α_4, else set $\boldsymbol{x}^{(t)} = \tilde{\boldsymbol{x}}^{(t)}$.

(3) Store the realizations $\left\{\boldsymbol{x}^{(t)}; t = 0, 1, \ldots, N\right\}$ for Bayesian inference.



(a) TMCMC for MLE: first co-ordinate.

(b) TMCMC for MLE: second co-ordinate.

Figure 7.7: TMCMC trace plots for Example 4 for finding MLE.

Figure 7.7, we obtain the estimate $\hat{\mathbf{x}}_{\text{MLE}} = (0.339627, 0.256524)$, for which $\|\mathbf{f}'(\hat{\mathbf{x}}_{\text{MLE}})\|_2 = 0.005108$. In contrast, the MLE obtained by Fisher's scoring (reported to four digits by Lange (2010)) is $\hat{\boldsymbol{\beta}}_{\text{MLE}} = (0.339634, 0.256524)$, with a gradient norm of 0.011791 — substantially farther from zero.

We also compared our approach with other popular optimization methods implemented in R: the BFGS (Broyden–Fletcher–Goldfarb–Shanno) method, simulated annealing followed by BFGS refinement, and a multi-start quasi-Newton method with 10 random starts uniformly sampled from $[-2, 2]^2$. The BFGS and simulated annealing results were obtained using R's `optim` function with default settings. For the multi-start quasi-Newton method, we report the mean of the estimates from the 10 runs; we deliberately did not select the best solution because the other competing methods rely on single starts only. Thus, the multi-start results are comparable to an average, not to a minimum-norm solution over multiple starts.

Up to the first six decimal places, Fisher's scoring, BFGS, and simulated annealing produced identical optima. Table 7.1 shows that in every case our Bayesian procedure achieves a substantially smaller gradient norm. All methods completed in a few seconds, so we do not report run times; gradient norms were computed analytically.

Thus, our Bayesian approach proves more reliable than existing methods. This may be a general phenomenon: by starting from good initial values (for instance, optima obtained by popular methods), TMCMC can explore regions of $\mathcal{X}$ that almost surely contain points with smaller gradient norms than those found by the initial method. It is also encouraging that for this example, TMCMC took well under a minute to generate 6×10^6 iterations on a single processor of our VMWare system.

7.5. Example 5

Hunter and Lange (2000) refer to a nonlinear optimization problem of the form $Y_i \sim$

Table 7.1: Comparison of optimization methods for Example 4 (Poisson regression MLE). The true gradient norm at the optimum is zero.

Method	$\hat{\beta}_0$	$\hat{\beta}_1$	$\|\nabla f\|_2$	$f(\hat{\beta}_0, \hat{\beta}_1)$
Fisher's scoring	0.339634	0.256524	0.011791	472.062548
BFGS	0.339634	0.256524	0.011791	472.062548
Simulated annealing	0.339634	0.256524	0.011791	472.062548
Multi-start quasi-Newton (10 starts)	0.339850	0.256501	0.123481	472.062547
Our method (TMCMC + posterior)	0.339627	0.256524	0.005108	472.062548

Our method achieves the smallest gradient norm, indicating closer proximity to the true MLE.

$N(\mu_i, \sigma^2)$ for $i = 1, \ldots, m$, where

$$\mu_i = \sum_{j=1}^{d} \left[\exp\left(-z_{ij}\theta_j^2\right) + z_{ij}\theta_{d-j+1}\right];$$

z_{ij} being the i-th observation of the j-th covariate, for $i = 1, \ldots, m$ and $j = 1, \ldots, d$. The goal is to compute the MLE of $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$, assuming that σ is known. In our notation, the objective is to minimize

$$f(\boldsymbol{x}) = \sum_{i=1}^{m} \left(y_i - \sum_{j=1}^{d} \left[\exp\left(-z_{ij}x_j^2\right) + z_{ij}x_{d-j+1}\right] \right)^2,$$

with respect to $\boldsymbol{x}$.

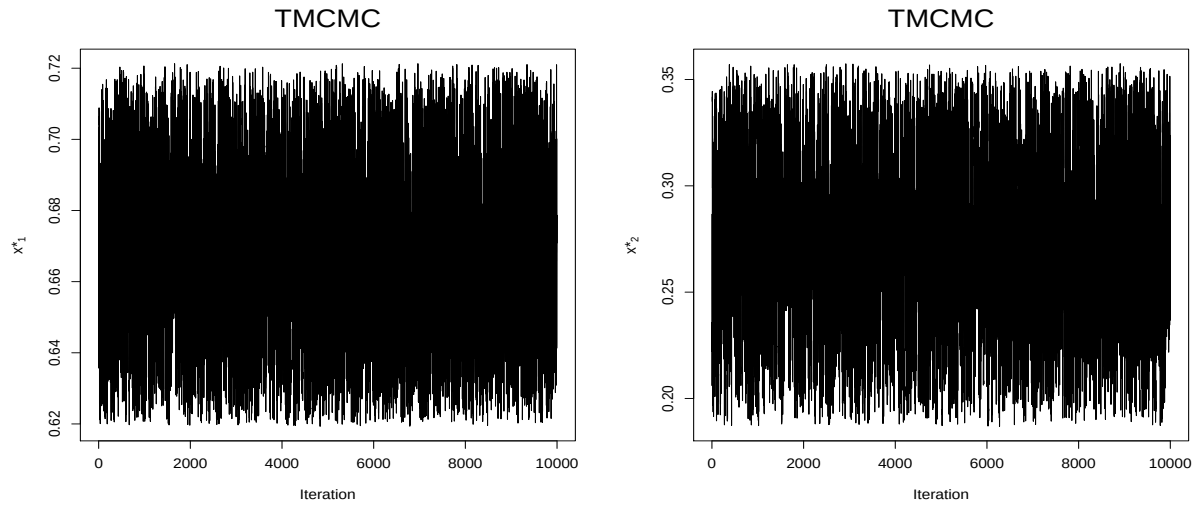
We consider 5 simulation experiments in this regard for $d = 2, 5, 10, 50, 100$. In each case we generate $\theta_{0j} \sim U(-1, 1)$ independently for $j = 1, \ldots, d$, $z_{ij} \sim N(0, 1)$ independently, for $i = 1, \ldots, m$ and $j = 1, \ldots, d$, and set, for $i = 1, \ldots, m$, $\mu_{0i} = \sum_{j=1}^{d} \left[\exp\left(-z_{ij}\theta_{0j}^2\right) + z_{ij}\right.$ $\left.\theta_{0,d-j+1}\right]$. We finally generate the response data by simulating $Y_i \sim N(\mu_{0i}, \sigma_0^2)$ independently for $i = 1, \ldots, m$, where we set $\sigma_0^2 = 0.1$. For $d = 2, 5, 10, 50, 100$, we generate datasets of sizes $m = 10, 10, 20, 75, 200$.

Note that this is not a convex minimization problem and the matrix of second derivatives $\boldsymbol{\Sigma}''(\cdot)$ plays an important role, along with $\|\boldsymbol{f}'(\cdot)\|_d$. Thus it is important to check positive definiteness of $\boldsymbol{\Sigma}''(\cdot)$ for any dimension d. We use the LAPACK library function "*dpotrf*" for Cholesky decomposition of $\boldsymbol{\Sigma}''(\cdot)$, which contains a parameter "$info$". Given any $\boldsymbol{x}$, $info = 0$ indicates positive definiteness of $\boldsymbol{\Sigma}''(\boldsymbol{x})$, while other values of $info$ rules out positive definiteness. Note that since this is not a convex minimization problem, step (2) of Algorithm 1 is necessary, unlike in Example 4.

We implement the mixture TMCMC algorithm 2 for all values of d, with $p = q = 1/2$ and set $a_j^{(1)} = a_j^{(2)} = 0.05$ for $j = 1, \ldots, d$.

7.5.1. Case 1: $d = 2$

In the TMCMC step, we set $\epsilon = 1$ in the restriction $\|\boldsymbol{f}'(\cdot)\|_2 < \epsilon$. The trace plots, shown in Figure 7.8, exhibit adequate mixing. Running step (2) of Algorithm 1 till $S = 40$



(a) TMCMC for MLE: first co-ordinate.

(b) TMCMC for MLE: second co-ordinate.

Figure 7.8: TMCMC trace plots for Example 5 for finding MLE for dimension $d = 2$.

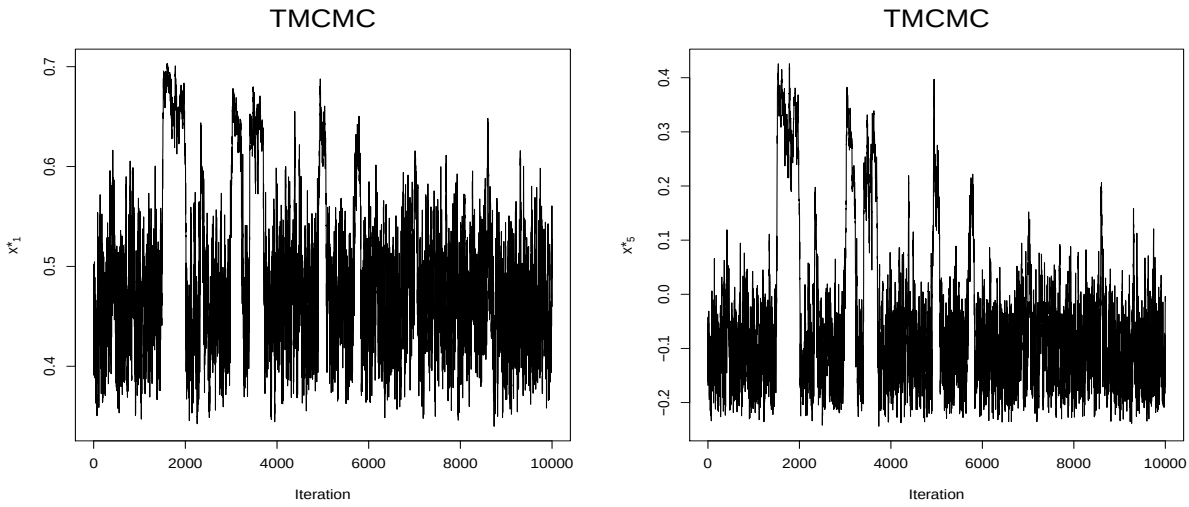
stages with $\eta_k = 1/(10 + k - 1)$ for $k = 1, \ldots, S$, yielded the estimate of the MLE to be $\hat{\boldsymbol{x}}_{MLE} = (0.678854, 0.293575)$, for which $\|\boldsymbol{f}'(\hat{\boldsymbol{x}}_{MLE})\|_2 = 0.012514$. The exercise takes 3 minutes on our VMWare, implemented in parallel on 100 cores.

However, examination of the samples obtained by importance resampling at the different stages of step (2) of Algorithm 1 did not reveal any evidence of multimodality, and hence it is pertinent to consider that estimate which corresponds to the minimum of $\{\|\boldsymbol{f}'(\boldsymbol{x}_i^*)\|_2 : i = 1, \ldots, N\}$, where $\boldsymbol{x}_i^*$ are the original TMCMC samples, with $N = 50000$. As such, we modify the previous estimate to $\hat{\boldsymbol{x}}_{MLE} = (0.678912, 0.293809)$, which yields $\|\boldsymbol{f}'(\hat{\boldsymbol{x}}_{MLE})\|_2 = 0.007221$, which is somewhat closer to zero compared to that for the previous estimate. Note however, that the two estimates of MLE are quite close to each other.

7.5.2. Case 2: $d = 5$

Here, for the 5-dimensional TMCMC, we set $\epsilon = 3$ in the restriction $\|\boldsymbol{f}'(\cdot)\|_5 < \epsilon$, as smaller values of ϵ led to poor convergence. Note that the Euclidean norm increases with dimension (see, for example, Giraud (2015)), and so it is a natural requirement to increase ϵ as dimension increases. Similarly, we had to increase η_k to $\eta_k = 1.5/\log(10 + k)$ for implementing step (2) of Algorithm 1. The rest of the parameters of Algorithm 2 remain the same as for $d = 2$.

The trace plots exhibited reasonable mixing; those for the first and the last (5-th) co-ordinate of $\boldsymbol{x}^*$ are depicted in Figure 7.9. Running step (2) of Algorithm 1 till $S = 40$ stages we obtain $\hat{\boldsymbol{x}}_{MLE} = (0.663327, 0.431669, 0.045598, 0.239091, 0.301665)$, which corresponds to $\min\{\|\boldsymbol{f}'(\boldsymbol{x}_i^*)\|_5 : i = 1, \ldots, N\} = 0.254217$, with $N = 50000$. Note the increase in $\|\boldsymbol{f}'(\cdot)\|_5$ compared to those of the smaller dimensions. Again, this is clearly to be expected because of the curse of dimensionality. The exercise takes 4 minutes to complete on our VMWare.



(a) TMCMC for MLE: first co-ordinate. (b) TMCMC for MLE: fifth co-ordinate.

Figure 7.9: TMCMC trace plots for Example 5 for finding MLE for dimension $d = 5$.

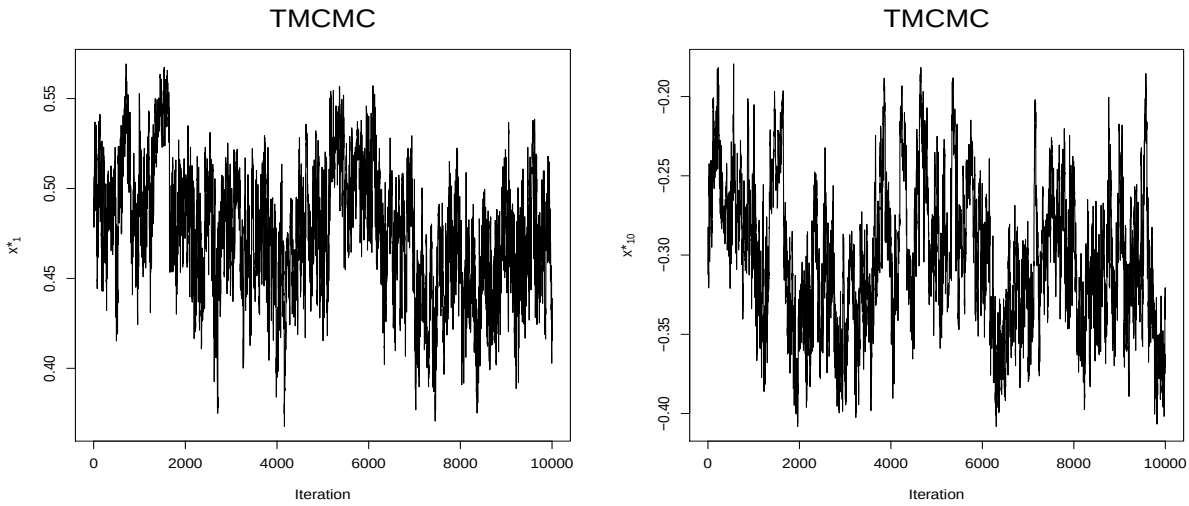
7.5.3. Case 3: $d = 10$

Here, for $d = 10$, we had to set $\epsilon = 6$ for the restriction $\|\boldsymbol{f}'(\cdot)\|_{10} < \epsilon$ in TMCMC for adequate convergence. We also set $\eta_k = 7/\log(10 + k - 1)$ for implementing step (2) of Algorithm 1.

Our investigation shows that the mixing of TMCMC is not inadequate. The trace plots of the first and the last co-ordinate of $\boldsymbol{x}^*$ shown in Figure 7.10 also bear evidence to this. After implementing step (2) of Algorithm 1 till $S = 40$ stages, we obtain $\hat{\boldsymbol{x}}_{MLE} = (0.472309, 0.10124, 0.079194, 0.108072, 0.096287, -0.511566, 0.517567, -0.887624, 0.637796, -0.317721)$ with $\|\boldsymbol{f}'(\hat{\boldsymbol{x}}_{MLE})\|_{10} = 1.917746$. On the other hand, $\min\{\|\boldsymbol{f}'(\boldsymbol{x}^*_i)\|_{10} : i = 1, \ldots, 50000\} = 1.902788$, which corresponds to $\hat{\boldsymbol{x}}_{MLE} = (0.471200, 0.100131, 0.078085, 0.101934, 0.095178, -0.504813, 40.516459, -0.888733, 0.631659, -0.323859)$. Thus, both the estimates as well as the corresponding gradients are quite close to each other. Again note the increase in $\|\boldsymbol{f}'(\cdot)\|_{10}$ compared to those of the smaller dimensions. The entire exercise takes 17 minutes on our VMWare to complete.

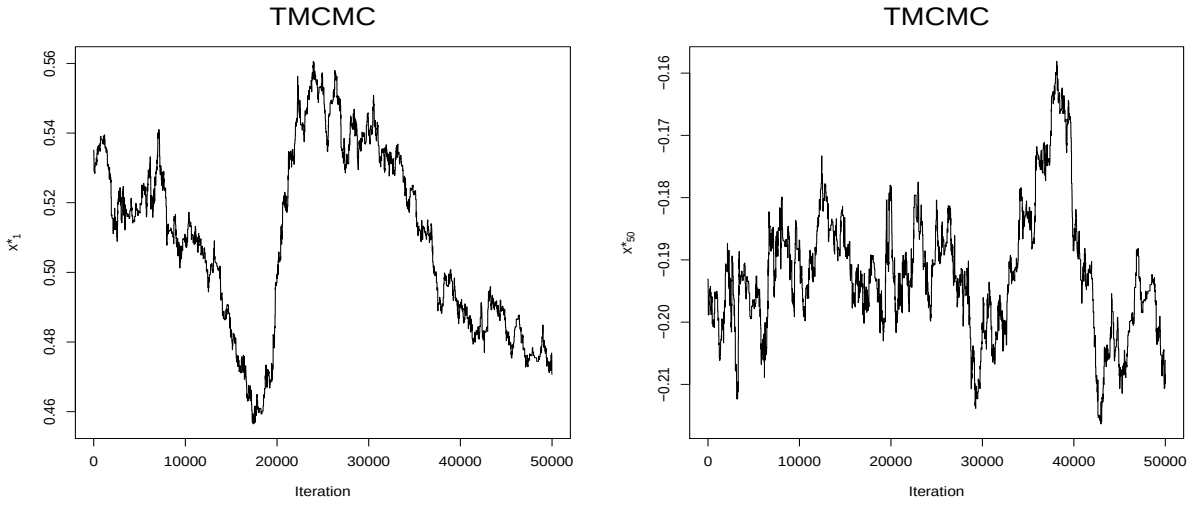
7.5.4. Case 4: $d = 50$

For this somewhat large dimension, we had to set $\epsilon = 100$ and $\eta_k = 200/\log(10+k-1)$. Figure 7.11, which displays all stored 50000 TMCMC realizations for x^*_1 and x^*_{50} do not indicate excellent mixing, in spite of the sophistication of Algorithm 2. However, due to the high-dimension and complexity of the posterior this is not unexpected. As demonstrated by Example 4, we can still expect to get closer to the MLE compared to other optimization methods. In this case, we obtain $\min\{\|\boldsymbol{f}'(\boldsymbol{x}^*_i)\|_{50} : i = 1, \ldots, 50000\} = 73.05261$ while step (2) of Algorithm 1 implemented for $S = 100$ stages, of which only the first four stages consisted of positive n_k, yielded $\|\boldsymbol{f}'(\hat{\boldsymbol{x}}_{MLE})\|_{50} = 76.28244$. Thus, the norms of the gradients have increased considerably in this high dimension, compared to the previous $d = 2, 5, 10$. Given the high dimension in this problem, the above gradient values 73.05261 and 76.28244



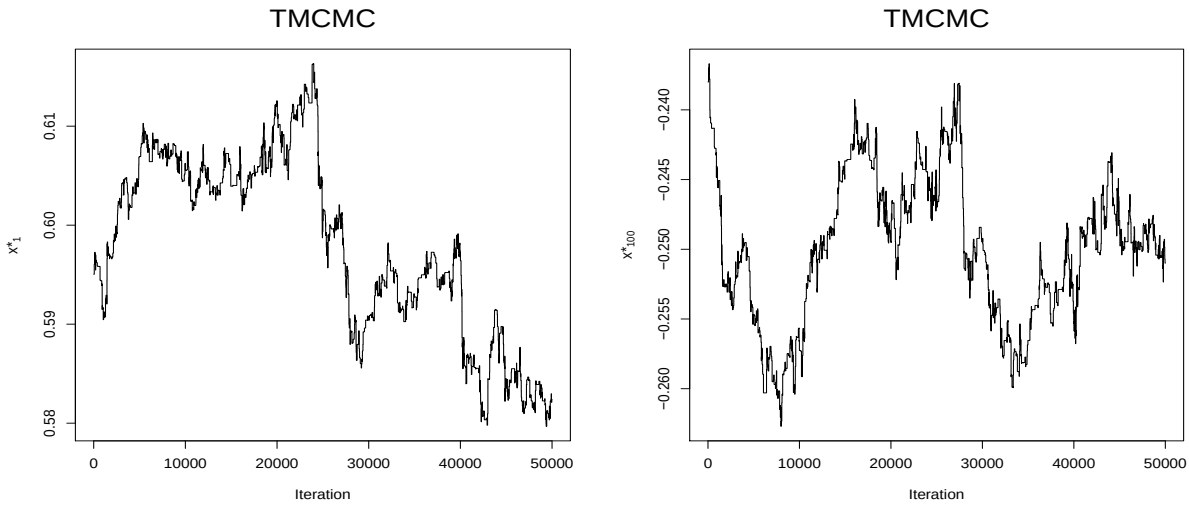
(a) TMCMC for MLE: first co-ordinate. (b) TMCMC for MLE: 10-th co-ordinate.

Figure 7.10: TMCMC trace plots for Example 5 for finding MLE for dimension $d = 10$.



(a) TMCMC for MLE: first co-ordinate. (b) TMCMC for MLE: 50-th co-ordinate.

Figure 7.11: TMCMC trace plots for Example 5 for finding MLE for dimension $d = 50$.



(a) TMCMC for MLE: first co-ordinate.

(b) TMCMC for MLE: 100-th co-ordinate.

Figure 7.12: TMCMC trace plots for Example 5 for finding MLE for dimension $d = 100$.

are quite close.

The TMCMC implementation took about 12 hours on a single core in our VMWare and step (2) of Algorithm 1 took additional 2 hours on 100 cores.

7.5.5. Case 5: $d = 100$

The curse of dimensionality now forced us to set $\epsilon = 400$ for TMCMC convergence and $\eta_k = 850/\log(10 + k - 1)$. It took around 15 hours to complete the TMCMC run; the trace plots shown in Figure 7.12 do not bear evidence of non-convergence, even though mixing is expectedly inadequate in such high dimension. Here we obtain $\min\{\|\boldsymbol{f}'(\boldsymbol{x}_i^*)\|_{50} : i = 1, \ldots, 50000\} = 330.4483$, while step (2) of Algorithm 1 implemented for till $S = 100$ stages, of which only the first three stages yielded positive n_k, produced $\hat{\boldsymbol{x}}_{MLE}$ such that $\|\boldsymbol{f}'(\hat{\boldsymbol{x}}_{MLE})\|_{100} = 341.2009$. Given the high dimension, this value is quite close to the above minimum gradient. Implementation of step (2) of Algorithm 1 took 2 hours 36 minutes on 100 cores on our VMWare.

8. Summary and conclusion

8.1. The key ideas in a nutshell

In this article, we have proposed and developed a novel Bayesian algorithm for general function optimization that judiciously exploits the derivatives of the objective function in conjunction with the posterior Gaussian derivative process, given data consisting of input points from the function domain and their corresponding function evaluations. The posterior simulation approach inherent to our method ensures improved accuracy of the obtained solutions compared to existing optimization algorithms. Another important feature is that, for any desired degree of accuracy, Bayesian credible regions of the optima are readily available at any stage of the algorithm.

Under an appropriate fixed-domain infill asymptotics setup, we prove that the algorithm converges almost surely to the true optima. Along the way, we establish almost sure uniform convergence of the posteriors corresponding to the Gaussian process and its derivative process to the objective function and its derivatives, and we provide rates of convergence for a specific infill design. Furthermore, we present a Bayesian characterization of the number of optima of the objective function by exploiting the information contained in our algorithm.

8.2. High-dimensional and mixing issues – i.i.d. sampling as an alternative to TMCMC

The application of our Bayesian optimization algorithm to a variety of examples, ranging from simple to challenging, has produced encouraging and insightful results. The choice of initial values for the TMCMC sampler can affect mixing; however, as argued in the paper, the optima obtained by good existing optimization algorithms can serve as effective initial values for TMCMC. Moreover, we have shown that even when mixing is less than ideal, our Bayesian algorithm can still significantly outperform popular optimization methods. Consequently, the TMCMC mixing issue may not be critical, at least for low-dimensional problems.

Dimensionality poses a much more serious challenge. Our experimental results demonstrate that as the dimension increases, the accuracy of the algorithm deteriorates. High dimensionality also severely impairs TMCMC mixing by excessively restricting the input space through the prior constraints. The reason is that the prior requires the gradient norm to be small and the Hessian positive definite; in high dimensions, the region of the parameter space that satisfies these conditions is extremely small, making it difficult for a Markov chain to explore effectively.

Fortunately, the TMCMC mixing problem can be completely circumvented by replacing TMCMC with the general i.i.d. sampling procedure developed by Bhattacharya (2025) (see also Bhattacharya (2021b), Bhattacharya (2021a), Bhattacharya (2022), Bhattacharya *et al.* (2025) for extensions). Using i.i.d. sampling, sampling accuracy can be maintained almost perfectly even as the dimension increases, which is expected to dramatically improve the quality of the optima obtained by our method. This approach eliminates the need for careful tuning of proposal distributions and the risk of poor mixing, making it particularly attractive for high-dimensional optimization problems.

8.3. Towards black-box optimization: a natural extension

Although the current work assumes that the first and second partial derivatives of the objective function are available, a natural and promising extension is to adapt the algorithm to settings where f is a black box – only function evaluations are possible, and derivatives are unavailable. In this case, one can replace the true derivatives $\mathbf{f}'$ and $\mathbf{\Sigma}''$ in the prior and the refinement step with their Gaussian process surrogates $\mathbf{g}_n'$ and the GP-based Hessian. Because Theorems 2 and 3 establish that $\mathbf{g}_n'$ converges uniformly to $\mathbf{f}'$ as the dataset grows, the modified algorithm would still converge to the true stationary points of f. The refinement condition $\|\mathbf{f}'(\mathbf{x})\|_d < \eta_k$ would be replaced by $\|\mathbf{g}_n'(\mathbf{x})\|_d < \eta_k$, and the Hessian check would use the GP posterior Hessian. This adaptation turns our derivative-based method into a

general-purpose, provably convergent black-box optimization algorithm – a rare combination in the Bayesian optimization literature. We intend to pursue this direction in future work, providing rigorous convergence guarantees and demonstrating its performance on real-world black-box problems.

8.4. Discussion of scalability and potential extensions

The cubic complexity $O(n^3)$ of the Gaussian process posterior, which arises from the inversion of the $n \times n$ correlation matrix $\mathbf{\Sigma}_{22}$, limits the method to moderate dataset sizes n (typically $n \leq 500$ in our experiments). For high-dimensional problems, the number of refinement stages k is small and the dataset size n remains modest. To scale the method to very large input dimensions or very large dataset sizes, several extensions could be explored.

One promising direction is the use of sparse Gaussian process approximations, such as the fully independent training conditional (FITC) method of Quiñonero-Candela and Rasmussen (2005). By introducing a small set of m inducing points with $m \ll n$, the complexity of the GP posterior reduces from $O(n^3)$ to $O(nm^2)$, making it feasible to handle larger datasets.

Another approach is to employ additive kernels, where the covariance function is assumed to decompose as a sum of one-dimensional kernels: $c(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{d} c_k(x_k, y_k)$. This additive structure allows separate Gaussian process modeling per dimension, effectively reducing the dimensionality of the problem and enabling more efficient computation; see Duvenaud *et al.* (2011).

Random Fourier features offer an alternative that approximates the covariance kernel using a finite set of random basis functions; see Rahimi and Recht (2007). This approximation yields linear complexity in the number of data points n and is particularly suitable for high-dimensional problems, as the number of features needed to achieve a given accuracy does not depend exponentially on d.

Finally, if the covariance kernel is separable, the posterior of the derivative process factorizes across dimensions. This factorization can be exploited for parallel or distributed computation, where each dimension is processed independently on a separate processor, leading to significant speedups in high dimensions.

These extensions, together with the black-box adaptation and i.i.d. sampling, constitute a rich agenda for future research. They would enable the application of our Bayesian optimization method to very high-dimensional optimization problems and to problems with large numbers of data points, while retaining the theoretical guarantees that are the hallmark of our approach.

Acknowledgment

We sincerely thank the reviewer whose comments have led to significant enhancement of the quality of our manuscript.

Conflict of interest

The authors declare no conflicts of interest.

Data and code availability statement

Our MPI-based parallel C code for implementing the AIDS data problem is available at `https://github.com/Sourabh-Bhattacharya/FUNCTION_OPT_GDP`. The data is contained in the code itself, and is also available in Lange (2010).

A. Computational complexity of Algorithm 1

A.1. Preliminary notation

- d – dimension of the input space $\mathcal{X}$.
- n_0 – size of the initial dataset $\mathcal{D}_{n_0}$.
- N_{TMCMC} – number of TMCMC iterations performed in step (1) (including burn-in and thinning).
- N – number of stored samples after thinning (for example, $N = 50000$).
- M – number of points resampled at each stage of step (2) (for example, $M = 1000$).
- K – total number of refinement stages.
- L – maximum number of points augmented per stage (typically $L = 5$).
- n_k – number of points actually augmented at stage k ($0 \leq n_k \leq L$).
- N_k – dataset size before stage k; $N_1 = n_0$, $N_{k+1} = N_k + n_k$.
- $N_{\max} = \max_k N_k \leq n_0 + LK$.
- $K_{\text{aug}} = \#\{k : n_k > 0\}$ – number of stages where augmentation occurs.

A.2. Assumptions

(i) *Dense linear algebra.* All matrix operations (multiplication, inversion, Cholesky decomposition) use standard algorithms. Multiplication of a $p \times q$ matrix by a $q \times r$ matrix costs $O(pqr)$; inversion of an $m \times m$ matrix costs $O(m^3)$; Cholesky decomposition costs $O(m^3)$.

(ii) *Analytical derivatives.* The first and second partial derivatives of the objective function f are available in closed form. Evaluating the gradient $\mathbf{f}'(\mathbf{x})$ costs $O(d)$, assembling the Hessian $\mathbf{\Sigma}''(\mathbf{x})$ costs $O(d^2)$, and testing positive definiteness via Cholesky decomposition costs $O(d^3)$.

(iii) *Dataset augmentation.* When new points are added to the dataset, the correlation matrix $\mathbf{\Sigma}_{22}$ and its inverse are recomputed from scratch (no low-rank updates).

(iv) *TMCMC implementation.* In each TMCMC iteration, the density of the current point is stored. Hence only one density evaluation (for the proposal) is performed per iteration. We assume additive TMCMC for simplicity.

A.3. Cost of one posterior density evaluation

A.3.1. Precomputed quantities

Before the TMCMC loop, the following matrices are computed once:

$$\mathbf{P} = \left(\mathbf{H}^T \boldsymbol{\Sigma}_{22}^{-1} \mathbf{H} + \boldsymbol{\Sigma}_0^{-1}\right)^{-1},$$

and the vector $\mathbf{v} = \boldsymbol{\Sigma}_{22}^{-1}(\mathbf{f}_n - \mathbf{H}\hat{\boldsymbol{\beta}})$, where $\hat{\boldsymbol{\beta}}$ is the posterior mean of $\boldsymbol{\beta}$. These quantities do not depend on the candidate point $\mathbf{x}$ and are reused throughout.

Lemma 2: For a fixed dataset $\mathcal{D}_n$ of size n and a point $\mathbf{x} \in \mathcal{X}$, the computation of

$$\pi\left(\mathbf{g}''(\mathbf{x}) = \mathbf{0} \mid \mathcal{D}_n, \mathbf{x}\right)$$

requires $O\left(dn^2 + d^2 n + d^3\right)$ floating-point operations, assuming that all quantities that do not depend on $\mathbf{x}$ (for example, $\boldsymbol{\Sigma}_{22}^{-1}$, $\mathbf{P}$, $\mathbf{A}\hat{\boldsymbol{\beta}}$) have been precomputed.

Proof: The evaluation follows equations 22, 19, 20, 8 and 9 of the main paper. We list the steps and their costs.

(1) Compute $\boldsymbol{\Sigma}_{12}(\mathbf{x})$ $(d \times n)$. Each column requires $O(d)$ work $\implies O(dn)$.

(2) Compute $\hat{\boldsymbol{\mu}}'(\mathbf{x}) = \mathbf{A}\hat{\boldsymbol{\beta}} + \boldsymbol{\Sigma}_{12}(\mathbf{x})\,\mathbf{v}$, where $\mathbf{v}$ is a precomputed n-vector. $\boldsymbol{\Sigma}_{12}\mathbf{v}$ costs $O(dn)$; addition of the constant $O(d)$.

(3) Compute $\mathbf{M} = \boldsymbol{\Sigma}_{12}(\mathbf{x})\,\boldsymbol{\Sigma}_{22}^{-1}$: $d \times n$ times $n \times n$ $\implies O(dn^2)$.

(4) Compute $\tilde{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{11} - \mathbf{M}\,\boldsymbol{\Sigma}_{12}(\mathbf{x})^T$: $\mathbf{M}\boldsymbol{\Sigma}_{12}^T$ is $d \times n$ times $n \times d$ $\implies O(d^2 n)$; subtraction $O(d^2)$.

(5) Compute $\mathbf{Q} = \mathbf{A} - \mathbf{MH}$: $\mathbf{MH}$ is $d \times n$ times $n \times (d+1)$ $\implies O(d^2 n)$; subtraction $O(d^2)$.

(6) Compute $\mathbf{QPQ}^T$: $\mathbf{QP}$: $d \times (d+1)$ times $(d+1) \times (d+1)$ $\implies O(d^3)$. Then $(\mathbf{QP})\mathbf{Q}^T$: $d \times (d+1)$ times $(d+1) \times d$ $\implies O(d^3)$. Add to $\tilde{\boldsymbol{\Sigma}}$: $O(d^2)$.

(7) Invert $\hat{\boldsymbol{\Sigma}}$ (size $d \times d$): $O(d^3)$ (Cholesky or LU).

(8) Compute the quadratic form $(\hat{\boldsymbol{\mu}}')^T \hat{\boldsymbol{\Sigma}}^{-1} \hat{\boldsymbol{\mu}}'$: matrix-vector product $O(d^2)$, dot product $O(d)$.

(9) Combine with the precomputed constant denominator: $O(1)$.

Summing the dominating terms gives $O(dn^2 + d^2 n + d^3)$. □

A.4. Cost of one TMCMC iteration (step (1))

Lemma 3: Each iteration of the TMCMC loop executed in step (1) of Algorithm 1 has complexity

$$O\Big(d^3 + d^2 n_0 + d n_0^2\Big).$$

Proof: An iteration proposes a new point $\mathbf{y}$ (cost $O(d)$), evaluates its density using Lemma 2 with $n = n_0$ (cost as above), and performs an acceptance test (constant time). The density of the current point is stored from the previous iteration. Hence the total cost is dominated by the density evaluation. □

A.5. Cost of one stage of the refinement loop (step (2))

Lemma 4: Consider stage k of step (2), where before the stage the dataset size is N_k (with $N_1 = n_0$) and after augmentation it becomes $N_{k+1} = N_k + n_k$ ($0 \le n_k \le L$). The following are available from the previous stage: all precomputed matrices for dataset size N_k, and the stored denominator densities $\pi(\mathbf{g}'(\mathbf{x}_i^*) = \mathbf{0} \mid \mathcal{D}_{N_k}, \mathbf{x}_i^*)$ for all stored samples $\mathbf{x}_i^*$. Then the computational cost of stage k is

$$\begin{aligned} C_k = O\Big(& N\Big(d^3 + d^2 N_k + d N_k^2\Big) \quad \text{(density evaluations for weights)} \\ & + N \quad \text{(weight multiplication and storage)} \\ & + (N + M) \quad \text{(resampling)} \\ & + M d^3 \quad \text{(gradient/Hessian checks)} \\ & + \mathbf{1}_{\{n_k > 0\}}\Big(N_{k+1}^3 + d^3 + d^2 N_{k+1} + n_k C_f\Big) \quad \text{(augmentation)}\Big), \end{aligned}$$

where N is the number of stored samples, M the resample size, and C_f the cost of evaluating the objective function f at a single point.

Proof: The work is broken down as follows.

(1) *Weight update (density evaluation).* For each of the N stored samples we need the new density $\pi(\mathbf{g}'(\mathbf{x}_i^*) = \mathbf{0} \mid \mathcal{D}_{N_k}, \mathbf{x}_i^*)$ (numerator). By Lemma 2 this costs $N \cdot C_{\text{eval}}(N_k) = O(N(d^3 + d^2 N_k + d N_k^2))$.

(2) *Weight update (multiplication).* Each new density is multiplied by the stored previous weight $w_{k-1}(\mathbf{x}_i^*)$ to obtain $w_k(\mathbf{x}_i^*)$, which is then stored. This costs $O(N)$.

(3) *Resampling.* To draw M distinct indices without replacement with probabilities proportional to the normalized weights $w_k(\boldsymbol{x}_i^*)$, we can use the alias method (Walker (1974, 1977); Vose (1991)) or systematic resampling (Kitagawa (1996); Carpenter *et al.* (1999)). Both methods require $O(N)$ preprocessing: building the alias table for the alias method or computing cumulative sums for systematic resampling. After preprocessing, each sample can be drawn in $O(1)$ time using the alias method, while systematic resampling draws all M samples in $O(M)$ time. Consequently, the total cost of resampling is $O(N + M)$.

(4) *Gradient and Hessian checks.* For each of the M resampled points, we compute $\|\mathbf{f}'(\mathbf{x})\|_d$ ($O(d)$), assemble the Hessian $\boldsymbol{\Sigma}''(\mathbf{x})$ ($O(d^2)$), and perform Cholesky decomposition to test positive definiteness ($O(d^3)$). Total $O(Md^3)$.

(5) *Augmentation (if $n_k > 0$).* We evaluate $f(\mathbf{x})$ for the n_k points that satisfy the gradient condition. The cost per evaluation is C_f, so total $O(n_k C_f)$. Then we recompute all precomputed quantities for the new dataset size N_{k+1}: forming $\boldsymbol{\Sigma}_{22}$ ($O(N_{k+1}^2)$) and its inverse ($O(N_{k+1}^3)$); recomputing $\mathbf{P} = (\mathbf{H}^T\boldsymbol{\Sigma}_{22}^{-1}\mathbf{H} + \boldsymbol{\Sigma}_0^{-1})^{-1}$ ($O(d^3 + d^2 N_{k+1})$); and other auxiliary matrices ($O(d^2 N_{k+1})$). The dominant cost is $O(n_k C_f + N_{k+1}^3 + d^3 + d^2 N_{k+1})$.

Summing all contributions yields the stated bound. □

Total complexity of Algorithm 1

Theorem 7: Let Algorithm 1 be run with initial dataset size n_0, N_{TMCMC} TMCMC iterations, N stored samples, resample size M, at most K refinement stages, and at most L points augmented per stage. Let C_f denote the cost of evaluating f at a single point. Then the total time complexity is

$$\begin{aligned} T_{\text{total}} = O\Big(& n_0^3 + d^3 + d^2 n_0 \quad \text{(precomputation)} \\ & + N_{\text{TMCMC}}(d^3 + d^2 n_0 + d n_0^2) \quad \text{(TMCMC)} \\ & + \sum_{k=1}^{K} N(d^3 + d^2 N_k + d N_k^2) \quad \text{(density evaluations for weights)} \\ & + KN \quad \text{(weight multiplication)} \\ & + K(N + M) \quad \text{(resampling)} \\ & + KMd^3 \quad \text{(gradient/Hessian checks)} \\ & + \sum_{k:n_k>0} (N_{k+1}^3 + d^3 + d^2 N_{k+1} + n_k C_f) \quad \text{(augmentation)} \Big), \end{aligned}$$

where $N_1 = n_0$, $N_{k+1} = N_k + n_k$, and $0 \le n_k \le L$.

Since the maximum dataset size $N_{\max} = \max_k N_k \le n_0 + LK$, the total complexity simplifies to

$$\begin{aligned} T_{\text{total}} = O\Big(& N_{\text{TMCMC}}(d^3 + d^2 n_0 + d n_0^2) + KN(d^3 + d^2 N_{\max} + d N_{\max}^2) \\ & + K(N + Md^3) + K_{\text{aug}}(N_{\max}^3 + LC_f) \Big), \end{aligned}$$

where $K_{\text{aug}} = \#\{k : n_k > 0\} \le K$.

Proof: The total cost is the sum of:

(1) *Precomputation (once).* Form $\boldsymbol{\Sigma}_{22}$ ($O(n_0^2)$), its inverse ($O(n_0^3)$), $\mathbf{P}$ ($O(d^3 + d^2 n_0)$), and other constant-time operations. Hence $O(n_0^3 + d^3 + d^2 n_0)$.

(2) *Step 1 (TMCMC)*: N_{TMCMC} iterations, each costing $O(d^3 + d^2 n_0 + d n_0^2)$ by Lemma 3.

(3) *Step 2*: sum of the costs of K stages given by Lemma 4. Summing over $k = 1, \ldots, K$ yields the terms shown. The term $\sum_k N(d^3 + d^2 N_k + d N_k^2)$ appears because each stage requires N density evaluations with dataset size N_k. The additional KN accounts for the multiplications. The $K(N + M)$ and KMd^3 come from resampling and gradient/Hessian checks. The last sum accounts for the recomputation after augmentation.

Using $N_k \leq N_{\max}$, we obtain the simplified bound. The cost C_f appears only in stages where $n_k > 0$, and each such stage contributes at most LC_f, so the total contribution from function evaluations is $O(K_{\text{aug}} L C_f)$. □

Remark 9: The cost C_f is problem-dependent. In all examples of this paper, f is a sum of elementary functions (polynomials, exponentials, etc.) evaluated at $O(md)$ terms, where m is the number of data points. Because m is at most 200 when $d = 100$, we have $C_f = O(d^2)$ (polynomial in d). The dominant term in the total complexity is then $N_{\text{TMCMC}} d^3$, since N_{TMCMC} is large (for example, 5×10^6). This matches the empirical runtimes reported in Section 7.

Remark 10: If the objective function f were such that C_f grows faster than any polynomial (for example, exponentially) in d, the total complexity could be dominated by $K_{\text{aug}} L C_f$. However, for the functions considered in this paper (and for most practical optimization problems where derivatives are available), C_f is at most polynomial in d, and the TMCMC term remains the primary bottleneck for moderate to large d.

References

Adler, R. J. (1981). *The Geometry of Random Fields*. John Wiley & Sons Ltd., New York.

Adler, R. J. and Taylor, J. E. (2007). *Random Fields and Geometry*. Springer, New York.

Bhattacharya, D., Maitra, T., Roy, S., and Bhattacharya, S. (2025). Bayesian nonparametrics with random normalizing flows. *ResearchGate preprint*, Available at `https://www.researchgate.net/publication/395337211_Bayesian_Nonparametrics_With_Random_Normalizing_Flows`.

Bhattacharya, S. (2021a). IID sampling from doubly intractable distributions. *arXiv preprint*, arXiv:2112.07939.

Bhattacharya, S. (2021b). IID sampling from intractable multimodal and variable-dimensional distributions. *arXiv preprint*, arXiv:2109.12633.

Bhattacharya, S. (2022). IID sampling from posterior dirichlet process mixtures. *arXiv preprint*, arXiv:2206.09233.

Bhattacharya, S. (2025). IID sampling from intractable distributions. *Sankhyā A*, To appear in Professor C. R. Rao special issue.

Bissiri, P. G., Holmes, C. C., and Walker, S. G. (2016). A general framework for updating belief distributions. *Journal of the Royal Statistical Society: Series B*, **78**, 1103–1130.

Carpenter, J., Clifford, P., and Fearnhead, P. (1999). An improved particle filter for non-linear problems. *IEE Proceedings - Radar, Sonar and Navigation*, **146**, 2–7.

Dey, K. K. and Bhattacharya, S. (2016). On geometric ergodicity of additive and multiplicative transformation based Markov Chain Monte Carlo in high dimensions. *Brazilian Journal of Probability and Statistics*, **30**, 570–613.

Dey, K. K. and Bhattacharya, S. (2017). A brief tutorial on transformation based Markov Chain Monte Carlo and optimal scaling of the additive transformation. *Brazilian Journal of Probability and Statistics*, **31**, 569–617.

Dey, K. K. and Bhattacharya, S. (2019). A brief review of optimal scaling of the main MCMC approaches and optimal scaling of additive TMCMC under non-regular cases. *Brazilian Journal of Probability and Statistics*, **33**, 222–266.

Dutta, S. (2012). Multiplicative random walk Metropolis-Hastings on the real line. *Sankhya. Series B*, **74**, 315–342.

Dutta, S. and Bhattacharya, S. (2014). Markov Chain Monte Carlo based on deterministic transformations. *Statistical Methodology*, **16**, 100–116.

Duvenaud, D., Nickisch, H., and Rasmussen, C. E. (2011). Additive Gaussian processes. *Advances in Neural Information Processing Systems*, **24**, 226–234.

Ferguson, T. S. (1973). A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, **1**, 209–230.

Frazier, P. I. (2018). A tutorial on Bayesian optimization. arXiv preprint.

Giraud, C. (2015). *Introduction to High-Dimensional Statistics*. CRC Press, New York.

Hunter, D. R. and Lange, K. (2000). Quantile regression via an MM algorithm. *Journal of Computational and Graphical Statistics*, **9**, 60–77.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *Journal of Computational and Graphical Statistics*, **5**, 1–25.

Lange, K. (2010). *Numerical Analysis for Statisticians*. Springer-Verlag, New York.

Liu, J. S. and Sabatti, S. (2000). Generalized Gibbs sampler and multigrid Monte Carlo for Bayesian computation. *Biometrika*, **87**, 353–369.

McKay, M. D., Beckman, R. J., and Conover, W. J. (1979). A comparison of three Methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, **21**, 239–245.

Quiñonero-Candela, J. and Rasmussen, C. E. (2005). A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, **6**, 1939–1959.

Rahimi, A. and Recht, B. (2007). Random features for large-scale kernel machines. *Advances in Neural Information Processing Systems*, **20**, 1177–1184.

Roy, S. and Bhattacharya, S. (2020). Bayes meets Riemann – Bayesian characterization of infinite series with application to Riemann hypothesis. *International Journal of Applied Mathematics and Statistics*, **59**, 81–128.

Santner, T. J., Williams, B. J., and I.Notz, W. (2003). *The Design and Analysis of Computer Experiments*. Springer-Verlag, New York.

Sobol', I. M. (1967). On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, **7**, 86–112.

Stein, M. L. (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer-Verlag, New York.

van der Vaart, A. W. and Wellner, J. A. (1996). *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer, New York.

Vose, M. D. (1991). A linear algorithm for generating random numbers with a given distribution. *IEEE Transactions on Software Engineering*, **17**, 972–975.

Walker, A. J. (1974). New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electronics Letters*, **10**, 127–128.

Walker, A. J. (1977). An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software*, **3**, 253–256.