Efficient batch-sequential Bayesian optimization with moments of truncated Gaussian vectors
Sébastien Marmin, Clément Chevalier, David Ginsbourger

To cite this version:
Sébastien Marmin, Clément Chevalier, David Ginsbourger. Efficient batch-sequential Bayesian optimization with moments of truncated Gaussian vectors. 2016. <hal-01361894>
Efficient batch-sequential Bayesian optimization with moments of truncated Gaussian vectors

Sébastien Marmin∗†‡, Clément Chevalier§, David Ginsbourger∗¶

September 7, 2016

Abstract
We deal with the efficient parallelization of Bayesian global optimization algorithms, and more specifically of those based on the expected improvement criterion and its variants. A closed form formula relying on multivariate Gaussian cumulative distribution functions is established for a generalized version of the multipoint expected improvement criterion. In turn, the latter relies on intermediate results that could be of independent interest concerning moments of truncated Gaussian vectors. The obtained expansion of the criterion enables studying its differentiability with respect to point batches and calculating the corresponding gradient in closed form. Furthermore, we derive fast numerical approximations of this gradient and propose efficient batch optimization strategies. Numerical experiments illustrate that the proposed approaches enable computational savings of between one and two order of magnitudes, hence enabling derivative-based batch-sequential acquisition function maximization to become a practically implementable and efficient standard. Keywords: Kriging, Expected Improvement, Parallel Optimization.

1 Introduction
Since their beginnings about half a century ago [26, 49, 31], Bayesian optimization algorithms have been increasingly used for derivative-free global minimization of expensive to evaluate functions. Typically assuming a continuous objective function $f : x \in D \subset \mathbb{R}^d \rightarrow f(x) \in \mathbb{R}$, single-objective Bayesian optimization algorithms consist in sequentially evaluating $f$ at promising points under the assumption that $f$ is a sample realization (path or trajectory) of a random field $(Y(x))_{x \in D}$. Such algorithms are especially popular in the case where evaluating $f(x)$ requires heavy high-fidelity numerical simulations (or computer experiments, see notably [33, 40, 41, 24]), where $x$ stands for some design parameters to be optimized over. Such expensive simulations are classically encountered in the resolution of partial differential equations from physical sciences, engineering and beyond [13]. In recent years, Bayesian optimization also has attracted a lot of interest from the machine learning community [27, 6, 34, 43], be it to optimize simulation-based objective functions [28, 45, 38] or even to estimate tuning parameters of machine learning algorithms themselves [3, 4, 42]. In both communities, a Gaussian random field (or Gaussian Process, GP) model is often used for $Y$, so that prior information on $f$ is taken into account through a trend function $m : D \rightarrow \mathbb{R}$ and a covariance kernel $k : (x, x') : D \times D \rightarrow \mathbb{R}$. Once $m$ and $k$ are specified, possibly up to some parameters to be inferred based on data, the considered GP model can be used as an instrument to locate the next evaluation point(s) via so-called infill sampling criteria, also referred to as acquisition functions or simply as criteria. While a number of Bayesian

∗IMSV, Department of Mathematics and Statistics, University of Bern, Switzerland
†Institut de Mathématiques de Marseille (UMR7373), École Centrale de Marseille, France
‡Institut de Radioprotection et de Sécurité Nucléaire (IRSN), PSN-RES, SEMIA, LIMAR, Cadarache, 13115 Saint-Paul-lès-Durance, France
§Institute of Statistics, University of Neuchâtel, Switzerland
¶Uncertainty Quantification and Optimal Design group, Idiap Research Institute, Martigny, Switzerland
using derivative-based deterministic search. Calculating and efficiently approximating its gradient in order to perform efficient batch optimization is on deriving Equation (1) in closed form, studying the criterion’s differentiability, and ultimately the multivariate Gaussian distribution underlying Equation (1). Our main focus here, in contrast, both what the criterion means and how to approach it by simulations, relying on three samples from where

\[
E = \text{the integral of the improvement (grey-scale function) with respect to the conditional distribution of corresponding sample path. The multipoint EI is the expectation of this length, or in other words, it is the integral of the improvement (grey-scale function) with respect to the conditional distribution of } (Y(x_{n+1}), Y(x_{n+q}))^\top \text{ knowing } A_n.
\]

optimization criteria have been proposed in the literature (see, e.g., [23, 15, 46, 43, 9] and references therein), we concentrate here essentially on the Expected Improvement (EI) criterion [30, 24] and on synchronous batch-sequential optimization. Denoting by \( x_1, \ldots, x_n \in D \) points were \( f \) is assumed to have already been evaluated and by \( x_{n+1:n+q} := (x_{n+1}, \ldots, x_{n+q}) \in D^q \) a batch of candidate points where to evaluate \( f \) next, the multipoint EI is defined as

\[
EI_n(x_{n+1:n+q}) = \mathbb{E}_n \left( \min_{i=1, \ldots, n} Y(x_i) - \min_{j=n+1, \ldots, n+q} Y(x_j) \right) \tag{1},
\]

where \( \mathbb{E}_n \) refers to the conditional expectation knowing the event \( A_n := \{ Y(x_1) = f(x_1), \ldots, Y(x_n) = f(x_n) \} \). One way of calculating such criterion is to rely on Monte Carlo simulations. Figure 1 illustrates both what the criterion means and how to approach it by simulations, relying on three samples from the multivariate Gaussian distribution underlying Equation (1). Our main focus here, in contrast, is on deriving Equation (1) in closed form, studying the criterion’s differentiability, and ultimately calculating and efficiently approximating its gradient in order to perform efficient batch optimization using derivative-based deterministic search.

Now, for \( q = 1 \), it is well known that EI can be expressed in closed form as a function of \( m_n(x) = \mathbb{E}_n(Y(x)) \) and \( s_n(x) = \sqrt{\text{var}_n(Y(x))} \) as follows

\[
EI_n(x) = s_n(x) (u_n(x) \Phi(u_n(x)) + \varphi(u_n(x))) \text{ if } s_n(x) \neq 0 \text{ and } 0 \text{ else,} \tag{2}
\]

where \( u_n(x) = (\min_{i=1, \ldots, n} f(x_i) - m_n(x))/s_n(x) \) (defined for \( s_n(x) \neq 0 \)) and \( \Phi, \varphi \) are the cumulative distribution function and probability density function of the standard Gaussian distribution, respectively.

When deriving Equation (2), Equation (1) happens (hence for \( q = 1 \)) to involve a first order moment of the truncated univariate Gaussian distribution. As shown in [7] and developed further
here, it turns out that Equation (1) can be expanded in a similar way in the multipoint case \((q \geq 2)\) relying on moments of truncated Gaussian vectors. This is essential for the open challenges tackled here of efficiently calculating and optimizing the multipoint criterion of Equation (1).

The applied motivation for having batch-sequential EI algorithms is strong, as distributing evaluations of Bayesian optimization algorithms over several computing units allows significantly reducing wall-clock time and with the fast popularization of clouds, clusters and GPUs in recent years it is becoming always more commonplace to launch several calculations in parallel. Even at a slightly inflated price and scripting effort, reducing the total time off is often a primary goal in order to deliver conclusions involving heavy experiments, be they numerical or laboratory experiments, in studies subject to hard time limitations. Obviously, given its practical importance, the question of parallelizing EI algorithms and alike by selecting \(q > 1\) points per iteration has been already tackled in a number of works from various disciplinary horizons (including notably [36, 1, 12, 8, 19]). Here we essentially focus on approaches relying on the maximization of Equation (1) and related multipoint criteria. The multipoint EI of Equation (1) has been defined in [30, 41] and first calculated in closed form for the case \(q = 2\) in [17]. For the case \(q \geq 3\), a Monte Carlo scheme and some sub-optimal batch selection strategies were proposed. Further work on Monte Carlo simulations for multipoint EI estimation can be found in [22, 18]; besides this, stochastic simulation ideas have been explored in [14] for maximizing this multipoint EI criterion via a stochastic gradient algorithm, an approach recently investigated in [47]. Meanwhile, a closed-form formula for the multipoint EI relying on combinations of \((q - 1)\)- and \(q\)-dimensional Gaussian cumulative distribution functions was obtained in [7], a formula which applicability in reasonable time is however restricted to moderate \(q\) (say \(q \leq 10\)) in the current situation. Building upon [7], [29] recently calculated the gradient of the multipoint EI criterion in closed form and obtained some first experimental results on (non-stochastic) gradient-based multipoint EI maximization.

Our aim in the present paper is to present a set of novel analytical and numerical results pertaining to the calculation, the computation, and the maximization of the multipoint EI criterion. As most of these novel results apply to a broader class of criteria, we first present in Section 2 a generalization of the multipoint EI that allows accounting for noise in conditioning observations and also exponentiating the improvement. This generalized criterion is calculated using moments of truncated Gaussian vectors in the flavour of [7]. The obtained formula is then revisited in the standard case (noise-free with an exponent set to 1), leading to a numerical approximation of the multipoint EI with arbitrary precision and very significantly reduced computation time. Next, the \((qd)\)-dimensional maximization of the multipoint EI criterion is discussed in Section 3, where the differentiability of the generalized criterion is studied, its analytical gradient is calculated, and further numerical approaches for fast gradient approximations with controllable accuracy are presented. Finally, Section 4 is dedicated to numerical experiments where, in particular, a multistart derivative-based multipoint EI maximization algorithm highlighting the benefits of the considered methodological principles and the proposed fast approximations is tested and compared to baseline strategies.

2 Criteria in parallel Bayesian optimization

2.1 General definition of Expected Improvement

Throughout this section the objective function \(f\) may be observed noise-free or in noise, meaning that at some arbitrary iteration \(i\) the observed value may be \(f(x_i)\) or \(f(x_i) + \varepsilon_i\), where \(\varepsilon_i\) is a realization of a zero mean Gaussian random variable with known (or estimated and plugged-in) variance. \(f\) is assumed to be one realization of a random field \(Y\), where \(Y\) has a Gaussian random field (GRF) distribution conditionally to events of the form \(A_n := \{Y(x_1) = f(x_1), \ldots, Y(x_n) = f(x_n)\}\) (with conditioning on \(Y(x_i) + \varepsilon_i\) in the noisy case, see for instance [35]). This setup naturally includes the case where \(Y\) is a GRF, but also the so-called Universal Kriging settings where \(Y\) is the sum of a trend with an improper prior and a GRF [33, 32]. Note that in noisy cases the \(\varepsilon_i\)'s are generally assumed to be
independent (although the case of $\varepsilon_i$'s forming a Gaussian vector is tractable), but more essentially they are assumed independent of $Y$.

In batch-sequential Bayesian Optimization we are interested in computing sampling criteria $J_n$ depending on $q \geq 1$ new points $\mathbf{x}_{n+1:n+q} = (\mathbf{x}_{n+1}, \ldots, \mathbf{x}_{n+q}) \in D^q$. At any step of corresponding synchronous parallel algorithms, the next batch of $q$ points $\mathbf{x}_{n+1:n+q}^*$ is then defined by globally maximizing $J_n$ over all possible batches:

$$\mathbf{x}_{n+1:n+q}^* \in \arg \max_{\mathbf{x}_{n+1:n+q} \in D^q} J_n(\mathbf{x}_{n+1:n+q}).$$

Values of such criteria typically depend on $\mathbf{x}_{n+1:n+q}$ through the conditional distribution $Y(\mathbf{x}_{1:n+q})|\mathcal{A}_n$, simplifying to $Y(\mathbf{x}_{n+1:n+q})|\mathcal{A}_n$ in the noiseless context. Conditional mean and covariance functions are analytically formulated via the so-called kriging equations, see e.g. [39]. Working out these criteria thus generally boils down to Gaussian vector calculus, which may become intricate and quite cumbersome analytically formulated via the so-called kriging equations.

2.2 Preliminaries on moments of truncated Gaussian distribution

We fix $\alpha \in \mathbb{N}\setminus\{0\}$ and $p = n + q$ in noisy settings or $p = q$ in noiseless settings.

**Definition 1.** Let $\mathbf{Z}$ be a Gaussian vector with mean $\mathbf{m} \in \mathbb{R}^p$ and covariance matrix $\Sigma \in S_{++}^p$, where $S_{++}^p$ is the cone of positive definite matrices of $\mathbb{R}^{p \times p}$. For all positive integer $k \leq p$, we define the function $M_{k,\alpha}$ on $\mathbb{R}^p \times S_{++}^p$ by

$$M_{k,\alpha} : (\mathbf{m}, \Sigma) \mapsto M_{k,\alpha}(\mathbf{m}, \Sigma) = \mathbb{E}_n \left( \mathbf{Z}_k^\top \mathbf{1}(\mathbf{Z} \leq \mathbf{0}) \right),$$

where the inequality $\mathbf{Z} \leq \mathbf{0}$ is to be interpreted component-wise.

If $\mathbf{Z}$ is composed of values of a GRF at a batch of $q$ locations $\mathbf{x}_{n+1:n+q}$, we use the notation $M_{k,\alpha}(\mathbf{Z}(\mathbf{x}_{n+1:n+q})) := M_{k,\alpha}(\mathbf{m}(\mathbf{x}_{n+1:n+q}), \Sigma(\mathbf{x}_{n+1:n+q}))$. We obtain the moments $M_{k,\alpha}(\mathbf{m}, \Sigma)$ of a truncated Gaussian distribution by an extension of Tallis' technique [44] to any order, presented in the following proposition:

**Proposition 1.** The function $G : \mathbb{R}^p \times \mathbb{R}^p \times S_{++}^p \rightarrow \mathbb{R}$ defined by

$$G(t, \mathbf{m}, \Sigma) = e^{\frac{1}{2}(t + \Sigma^{-1}\mathbf{m})^\top \Sigma(t + \Sigma^{-1}\mathbf{m}) - \mathbf{m}^\top \Sigma^{-1}\mathbf{m})} \Phi_p,\Sigma(-\mathbf{m} - \Sigma t),$$

where $\Phi_p,\Sigma$ is the cone of positive definite matrices of $\mathbb{R}^{p \times p}$, is the cone of positive definite matrices of $\mathbb{R}^{p \times p}$, is the cone of positive definite matrices of $\mathbb{R}^{p \times p}$, is the cone of positive definite matrices of $\mathbb{R}^{p \times p}$, is the cone of positive definite matrices of $\mathbb{R}^{p \times p}$, is the cone of positive definite matrices of $\mathbb{R}^{p \times p}$, is the cone of positive definite matrices of $\mathbb{R}^{p \times p}$.
where $\Phi_{p, \Sigma}(\cdot)$ is the cumulative distribution function of the centered $p$-variate normal distribution, is infinitely differentiable, and the moments $M_{k, \alpha}$ are given by:

$$M_{k, \alpha}(m, \Sigma) = \left. \frac{\partial^\alpha G(\cdot, m, \Sigma)}{\partial t_k^\alpha} \right|_{t=0}.$$  

(7)

The proof of this Proposition is given in appendix B.1 and relies on calculating the moment generating function $t \to \mathbb{E} \left[ \exp \left( t^\top Z \right) 1(Z \leq 0) \right]$. Even if an analytical formula can be obtained at any order of differentiation $\alpha$, the complexity of derivatives in equation (7) increases rapidly. We give below the results for $\alpha$ equals 1 and 2.

**Case $\alpha = 1$**

Differentiating $G$ with respect to $t$ yields:

$$\frac{\partial G}{\partial t}(t, m, \Sigma) = \exp \left( \frac{1}{2} \left( (t + \Sigma^{-1} m)^\top \Sigma (t + \Sigma^{-1} m) - m^\top \Sigma^{-1} m \right) \right) \times$$

$$\left( (\Sigma (t + \Sigma^{-1} m) \Phi_{p, \Sigma}(-m - \Sigma t) - \Sigma \nabla \Phi_{p, \Sigma}(-m - \Sigma t) \right)$$

where $\nabla \Phi_{p, \Sigma}$ is the gradient of $\Phi_{p, \Sigma}$ (see appendix A.1 for an analytical derivation). Taking $t = 0$ in the previous equation gives

$$M_{k, 1}(m, \Sigma) = m_k \Phi_{p, \Sigma}(-m) - \Sigma_k^\top \nabla \Phi_{p, \Sigma}(-m)$$

(8)

where $\Sigma_k$ is the $k$th column of $\Sigma$. It is shown in appendix A.1 that computing each of the $p$ components of $\nabla \Phi_{p, \Sigma}$ requires to compute a multivariate CDF of the normal distribution in dimension $p - 1$. The number of calls to this function for computing the first moment of the truncated Gaussian distribution is thus of $O(p)$.

**Case $\alpha = 2$**

Similarly, differentiating $G$ twice with respect to $t$ yields

$$M_{k, 2}(m, \Sigma) = (\Sigma_{kk} + m_k^2) \Phi_{p, \Sigma}(-m) + \Sigma_k^\top \nabla \nabla \Phi_{p, \Sigma}(m) \Sigma_k$$

$$+ 2m_k M_{k, 1}(m, \Sigma).$$

(9)

For readability, the detailed formula of $\nabla \nabla \Phi_{p, \Sigma}$, the Hessian matrix of $\Phi_{p, \Sigma}$, is sent to Appendix A.2. The number of calls to the multivariate normal CDF is of $O(p^2)$.

### 2.3 Analytic formulas for generalized $q$-EI

The previous results obtained for the moments of the truncated normal distribution turn out to be of interest for computing the generalized $q$-EI introduced in Equation (4), as shown by the following proposition.

**Proposition 2.** For $x_{n+1:n+q} \in \mathcal{D}^q$, the criterion $EI_n$ defined by (4) exists for all $\alpha$ and can be written as a sum of moments of truncated normal distributions

$$EI_n(x_{n+1:n+q}) = \sum_{\ell=1}^{n} \sum_{k=1}^{q} M_{n+k-1, \alpha}(z^{(\ell,k)}(x_{n+1:n+q})),$$

(10)
with $Z^{(\ell,k)}(x_{n+1:n+q})$ a vector of size $n+q-1$ defined, noting $Y_i := Y(x_i)$, by

$$
Z_i^{(\ell,k)} = \begin{cases}
Y_i - Y_i & \text{if } 1 \leq i \leq \ell - 1, \\
Y_i - Y_{i+1} & \text{if } \ell \leq i \leq n - 1, \\
Y_k - Y_{i+1} & \text{if } n \leq i \leq n + q - 1 \text{ and } i \neq n + k - 1, \\
Y_k - Y_{\ell} & \text{if } i = n + k - 1.
\end{cases}
$$

Moreover, in the noiseless case the random vector $(Y(x_1), \ldots, Y(x_n))$ becomes deterministic conditionally to $\mathcal{A}_n$. Denoting by $l_0$ the (smallest) index of the minimal observation, i.e. $Y_{l_0} = \min_{i=1,\ldots,n} Y_i$, and writing $Z^{(k)}(x_{n+1:n+q})$ the vector of the $q$ last components of $Z^{(l_0,k)}(x_{n+1:n+q})$, Equation (10) is simplified to:

$$
EI_n(x_{n+1:n+q}) = \sum_{k=1}^{q} \mathcal{M}_{k,\alpha} \left( Z^{(k)}(x_{n+1:n+q}) \right).
$$

**Remark 1.** In the rest of the article we also use the following compact notation for the $(n+q-1)$-dimensional vector $Z^{(\ell,k)}(x_{n+1:n+q})$:

$$
Z^{(\ell,k)}(x_{n+1:n+q}) = A^{(\ell,k)} \left( Y(x_1), \ldots, Y(x_{n+q}) \right)^\top,
$$

where $A^{(\ell,k)}$ is a matrix implicitly defined by $Z_i^{(\ell,k)}$ of proposition 2.

The proof of Proposition 2 is relegated to Appendix C for conciseness. Equation (10) highlights that the computation of the generalized $q$-EI in noisy settings is challenging since it involves computing $nq$ different moments, each requiring $(n+q)^\alpha$ calls to the multivariate normal CDF in a dimension close to $n + q$. Even for $\alpha = 1$ and moderate $q$, the linear dependence in the number of observations $n$ makes the use of this criterion challenging in application. Regarding the noiseless criterion, the computation of $q$ moments is more affordable, at least for moderate $q$, but one has to keep in mind that the ultimate goal here is to perform global maximization of the considered criteria. It is thus important to bring further calculation speed-ups in order to perform this optimization in a reasonable time compared to the evaluation time of the objective function $f$, assumed expensive to evaluate.

The next section discusses these matters and proposes faster formulas to compute both $q$-EI and its gradient.

### 3 Computing and optimizing the criteria

#### 3.1 Generalities

Maximizing the $EI_n$ expressions given in Equation (10) (noisy settings) or (11) (noiseless settings) is difficult. These maximizations are performed with respect to a batch of $q$ points $x_{n+1:n+q} \in (\mathbb{R}^d)^q$, and are thus optimization problems in dimension $dq$. In this space, the objective function to be maximized, is not convex in general and has the interesting property that the $q$ points in the batch can be permuted without changing the value of $EI_n$; i.e. $EI_n((x_{n+1}, \ldots, x_{n+q})) = EI_n((x_{n+\sigma(1)}, \ldots, x_{n+\sigma(q)}))$ for any permutation $\sigma$ of \{1, \ldots, $q$\}. With this property, one can reduce the measure of the search domain by $q!$, e.g. by imposing that the first coordinate of the $q$ points in the batch are in ascending order. We will restrict our attention here to the use of multi-start gradient based local optimization algorithms acting on the whole input domain $D^n \subset \mathbb{R}^d$, that do not exploit the structure of the problem but do not seem to be affected by this, at least with the chosen settings regarding the starting designs. Our contribution here will be to propose a faster formula for computing the first moments $\mathcal{M}_{k,1}$ previously presented, as well as their derivatives. This will yield an easier computation of both the generalized EI and its $dq$-dimensional gradient. Besides, a second approximate but faster formula to further reduce the calculation time of the gradient will be introduced.
3.2 Gradient of the generalized q-EI

In this section, we extend the analytical gradient calculation of the q-EI performed in [29] to the case of the generalized noisy and noise-free q-EI, and provide in turn a more concise formula. Again, the presented formulas rely on results on moments of truncated Gaussian distributions.

**Proposition 3.** Let \( x_{n+1:n+q} \in D^q \) be a batch such that the conditional covariance matrix \( \text{cov}(Y(x_{n+1})), Y(x_{n+j})|A_n) \) is positive definite and the functions \( \mathbb{E}(Y(\cdot)|A_n) \) and \( \text{cov}(Y(\cdot), Y(x_{n+j})|A_n) \) are differentiable at each point \( x_{n+i} \) (1 \( \leq i \leq q \)). These derivatives are written \( m^{(i)} \in \mathbb{R}^d \) and \( \Sigma^{(i)} \in \mathbb{R}^{q \times d} \) respectively. In this setup, the EI\(_n\) function of Equation (10) is differentiable and its derivative with respect to the \( j \)th coordinate of the point \( x_{n+i} \) is

\[
\frac{\partial EI}{\partial x_{ij}}(x_{n+1:n+q}) = \sum_{\ell=1}^{n} \sum_{k=1}^{q} m^{(i)}_{\ell,k} A_{\ell,k}^T \frac{\partial M_{n+k-1,1}}{\partial m} (Z^{(\ell,k)}) + \text{tr} \left( A^{(\ell,k)} \Gamma^{(i,j)} A^{(\ell,k)}^T \frac{\partial M_{n+k-1,1}}{\partial \Sigma} (Z^{(\ell,k)}) \right),
\]

where \( \Gamma^{(i,j)} = (\Sigma^{(i)}_{u,\delta} \delta_{i,u} + \Sigma^{(i)}_{\delta,\delta} \delta_{i,u}) \in \mathbb{R}^{q \times q} \), and \( \delta \) is the Kronecker symbol. The derivatives \( \frac{\partial M_{n+k-1,1}}{\partial m} \) and \( \frac{\partial M_{n+k-1,1}}{\partial \Sigma} \) are calculated in Appendix B.2.

This new expansion of the gradient of the generalized EI as a sum of derivatives of first order moments will prove to be very useful thanks to formulas presented next.

3.3 Fast numerical estimation of first order moments and their derivatives

Let us now focus on the practical implementation of the closed-form formula of Equation (10). We take \( a = 1 \) and note \( p = n + q \) in noisy settings and \( p = q \) in noiseless settings. As mentioned before, the computation of the noisy or noiseless q-EI (see, Eqs. (10),(11)) requires calls to the CDF of the \( p \) and \( (p - 1) \)-variate normal distribution, \( \Phi_p \) and \( \Phi_{p-1} \). These CDFs are here computed using the Fortran algorithms of [16] wrapped in the mnormt R package [2]. A quick look at Eqs. (8),(10) suggests that the noisy q-EI requires \( nq \) evaluations of \( \Phi_p \) and \( nq^2 \) evaluations of \( \Phi_{p-1} \). For the noiseless case (see, Equation (11)), the number of calls are divided by \( n \). In both cases, a slight improvement can be obtained by noticing a symmetry which reduces the number of \( \Phi_{p-1} \) calls from \( nq^2 \) (resp. \( q^2 \) in the noiseless case) to \( nq(q+1)/2 \) (resp. \( q(q+1)/2 \)). This symmetry is justified in Appendix E.

Despite this improvement, and even in the classical noiseless case, the number of \( \Phi_{p-1} \) calls is still proportional to \( q^2 \). We now give new efficient and trustworthy expansion that enables a fast and reliable approximation of first order moments of truncated Gaussian vectors \( \mathcal{M}_{k,1} \) by reducing this number of calls to \( O(q) \).

**Proposition 4.** Let \( \varepsilon > 0 \), and let \( Z \) be a Gaussian random vector with mean vector and covariance matrix \( (m, \Sigma) \in \mathbb{R}^p \times S_{++}^p \). Then we have

\[
\mathcal{M}_{k,1}(m, \Sigma) = \frac{1}{\varepsilon} (e^{m+\varepsilon \Phi_p,\Sigma(-\Sigma_k - m)} - \Phi_p,\Sigma(-m)) + O(\varepsilon^2).
\]

**Proof.** Let us consider the function \( g_k : t \in \mathbb{R} \rightarrow e^{m_k t} \Phi_p,\Sigma(-\Sigma_k t - m) \). This function \( g_k \) is tangent at \( t = 0 \) with the function \( t \in \mathbb{R} \rightarrow \mathcal{G}(te_k) \), where the function \( \mathcal{G} \) is introduced in Proposition 1 and \( e_k \) is the \( k \)th vector of the canonical basis. It follows from Proposition 1 that

\[
\left. \mathcal{M}_{k,1}(m, \Sigma) = \frac{\partial \mathcal{G}}{\partial t_k}(t, m, \Sigma) \right|_{t=0},
\]

and we obtain the announced result by Taylor expansion of \( g_k \). \( \square \)
The obtained formula simply uses the approximation of a moment with finite differences of the moment generating function. We showed here that instead of fully computing the moment generating function, we can expand the simpler tangent function \( g_k \). For conciseness, we name here the use of this formula “tangent moment method”. This formula thus enables approximating the first order moment \( M_{k,1} \) at the cost of only two calls to \( \Phi_p \). Hence, from Equation (11), computing a noiseless \( q \)-EI can be performed at the cost of \( 2q \) calls to \( \Phi \). Besides, a similar approach can be applied to approximate the gradient of \( q \)-EI through faster computations of \( \frac{\partial M_{k,1}}{\partial m} \) and \( \frac{\partial M_{k,1}}{\partial \Sigma} \), as shown next:

**Proposition 5.** The following equations hold:

\[
\frac{\partial M_{k,1}}{\partial m} = \Phi_{p,\Sigma}(-m) e_k - \frac{1}{\varepsilon} (e_m e \nabla \Phi_{p,\Sigma}(- \Sigma k e - m) - \nabla \Phi_{p,\Sigma}(-m)) + O(\varepsilon^2) \tag{15}
\]

\[
\frac{\partial M_{k,1}}{\partial \Sigma} = - \left( \frac{\partial \Phi_{p,\Sigma}}{\partial x_v} (-m) \delta_{u,k} + \frac{\partial \Phi_{p,\Sigma}}{\partial x_u} (-m) \delta_{v,k} \right)_{u,v \leq p} + \frac{1}{\varepsilon} (e_m e \nabla \Sigma \Phi_{p,\Sigma}(- \Sigma k e - m) - \nabla \Sigma \Phi_{p,\Sigma}(-m)) + O(\varepsilon^2) \tag{16}
\]

where \( \nabla \nabla^\top \Phi_{p,\Sigma} \) is the Hessian matrix of \( \Phi_{p,\Sigma} \) (see Appendix A.2 for details).

As before, these formulas enable reducing the number of calls to the multivariate CDF by an order \( q \). For the computation of \( q \)-EI this number goes from \( O(q^2) \) to \( O(q) \). For computing its \( dq \)-dimensional gradient, it goes from \( O(q^4) \) to \( O(q^3) \). The latter complexity suggests restricting to moderate values of \( q \) in applications. In the next section we present further results that enable further reducing the complexity for numerically estimating the gradient.

### 3.4 A slightly biased but fast proxy of the gradient

The key idea to obtain further computational savings is summarized in this section. We first strategically decompose the gradient of moments as a sum of two terms.

**Proposition 6.** Let us consider a Gaussian multivariate random field \( \mathbf{Z} \) from \( \mathbb{R}^d \) to \( \mathbb{R}^p \). For \( \mathbf{x} \in \mathbb{R}^d \), let us denote by \( \mathbf{m}(\mathbf{x}) \) and \( \Sigma(\mathbf{x}) \) the mean and the covariance matrix of \( \mathbf{Z}(\mathbf{x}) \). Let \( \mathbf{x}_a \in \mathbb{R}^d \) and assume that \( \Sigma(\mathbf{x}_a) \) is positive definite. Also, assume that the functions \( \mathbf{x} \rightarrow \mathbf{m}(\mathbf{x}) \), \( \mathbf{x} \rightarrow \Sigma(\mathbf{x}) \) and \( \mathbf{x} \rightarrow (\text{cov}(\mathbf{Z}_i(\mathbf{x}), \mathbf{Z}_j(\mathbf{x}))_{i,j \leq p}) \) are differentiable at \( \mathbf{x} = \mathbf{x}_a \). Then the following decomposition holds for \( k = 1, \ldots, p \):

\[
\nabla \mathbf{x}_a \left[ M_{k,\alpha}(\mathbf{m}(\mathbf{x}), \Sigma(\mathbf{x})) \right]_{\mathbf{x} = \mathbf{x}_a} := \nabla \mathbf{x}_a \left[ \mathbb{E} \left( Z^\alpha_k(\mathbf{x}) \mathbf{1}_{\{\mathbf{Z}(\mathbf{x}) \leq 0\}} \right) \right]_{\mathbf{x} = \mathbf{x}_a} \\
= \nabla \mathbf{x}_a \left[ \mathbb{E} \left( Z^\alpha_k(\mathbf{x}) \mathbf{1}_{\{\mathbf{Z}(\mathbf{x}) \leq 0\}} \right) \right]_{\mathbf{x} = \mathbf{x}_a} + \nabla \mathbf{x}_a \left[ \mathbb{E} \left( Z^\alpha_k(\mathbf{x}_a) \mathbf{1}_{\{\mathbf{Z}(\mathbf{x}) \leq 0\}} \right) \right]_{\mathbf{x} = \mathbf{x}_a}. \tag{17}
\]

**Proof.** \( \Sigma(\cdot) \) is continuous at \( \mathbf{x}_a \), so there exists a neighborhood \( V_{\mathbf{x}_a} \) of \( \mathbf{x}_a \) such that for all \( \mathbf{x} \in V_{\mathbf{x}_a} \), \( \Sigma(\mathbf{x}) \) is positive definite. Let us define on \( V_{\mathbf{x}_a} \times V_{\mathbf{x}_a} \):

\[
g(\mathbf{u}, \mathbf{v}) = \mathbb{E} \left( Z^\alpha_k(\mathbf{u}) \mathbf{1}_{\{\mathbf{Z}(\mathbf{v}) \leq 0\}} \right).
\]

Applying equation (22) of appendix B.1, for all \( \mathbf{u} \) and \( \mathbf{v} \), \( g(\mathbf{u}, \mathbf{v}) \) is a moment generated by differentiation of the following function:

\[
M_{\mathbf{u}, \mathbf{v}} : t \rightarrow e^{\frac{1}{2}(2 \Sigma k(\mathbf{u}) e + 2 \mathbf{m}(\mathbf{u})) \Phi_{p, \Sigma(\mathbf{v})}(- (\mathbf{m}(\mathbf{v}) - t (\text{cov}(\mathbf{Z}(\mathbf{u}), \mathbf{Z}(\mathbf{v}))))^\top) + \frac{1}{2} \text{tr}(\Sigma k(\mathbf{u}) \Sigma(\mathbf{v}))}.
\tag{18}
\]
The analytical form of equation (18) and the assumed differentiability of \( \mathbf{x}_a \) ensure existence of partial derivatives of \( g = (\mathbf{u}, \mathbf{v}) \rightarrow \frac{d^a M_{\mathbf{u}, \mathbf{v}}}{dx_0}(0) \) at \((\mathbf{x}_a, \mathbf{x}_a)\). So to conclude,

\[
\nabla_{\mathbf{x}} [M_{k, a} (\mathbf{m}(\mathbf{x}), \Sigma(\mathbf{x}))]_{\mathbf{x}=\mathbf{x}_a} = \nabla_{\mathbf{x}} [g(\mathbf{x}, \mathbf{x})]_{\mathbf{x}=\mathbf{x}_a} = \frac{\partial}{\partial \mathbf{u}} [g(\mathbf{u}, \mathbf{x}_a)]_{\mathbf{u}=\mathbf{x}_a} + \frac{\partial}{\partial \mathbf{v}} [g(\mathbf{x}_a, \mathbf{v})]_{\mathbf{v}=\mathbf{x}_a}
\]

\[
\]

The latter decomposition can be interpreted as follows: infinitesimal variations of \((\mathbf{m}(\mathbf{x}), \Sigma(\mathbf{x}))\) around \((\mathbf{m}(\mathbf{x}_a), \Sigma(\mathbf{x}_a))\) modify the moments \(M_{k, a} (\mathbf{m}(\mathbf{x}), \Sigma(\mathbf{x}))\) in two ways. First, it modifies the distribution of \(Z^a_{\mathbf{k}}(\mathbf{x})\), second it changes the distribution of the truncation \(1_{\{ \mathbf{Z}(\mathbf{x}) \leq 0 \}}\). For the particular case of \(q\)-EI, we propose to neglect this second variation. Applying this approximation to (11) gives for \(X_0 \in D^q\),

\[
\nabla_{\mathbf{x}_{n+j}} EI(\mathbf{x}_{n+1:n+q}=X_0) = \sum_{k=1}^{q} \nabla_{\mathbf{x}_{n+j}} \mathbb{E} \left( (T - Y(\mathbf{x}_{n+k}))^\alpha 1_{\{A^{(k)}Y(\mathbf{x}_{n+1:n+q}) \leq 0\}} \right)_{\mathbf{x}_{n+1:n+q}=X_0} \\
\approx \sum_{k=1}^{q} \nabla_{\mathbf{x}_{n+j}} \mathbb{E} \left( (T - Y(\mathbf{x}_{n+k}))^\alpha 1_{\{A^{(k)}Y(X_0) \leq 0\}} \right)_{\mathbf{x}_{n+1:n+q}=X_0} \\
= -\nabla_{\mathbf{x}_{n+j}} \mathbb{E} \left( Y(\mathbf{x}_{n+j})^\alpha 1_{\{A(0)Y(X_0) \leq 0\}} \right)_{\mathbf{x}_{n+1:n+q}=X_0} \\
= -\mathbb{E} \left( \nabla_{\mathbf{x}_{n+j}} Y(\mathbf{x}_{n+j})^\alpha \right)_{\mathbf{x}_{n+1:n+q}=X_0} 1_{\{A(0)Y(X_0) \leq 0\}}, \tag{19}
\]

where the last step is obtained by mean square differentiability of the process \(\mathbf{x} \rightarrow Y(\mathbf{x})^\alpha 1_{B}\), with \(B\) an event constant with respect to \(\mathbf{x}\), see Appendix D. We can observe that this approximation makes a summation term disappear. The computation of this formula requires \((d + 1)\) evaluations of \(q\)-variate Gaussian CDF. Indeed, Equation (19) indicates that each component of the gradient vector can be considered as a moment of a truncated Gaussian vector, so we can apply the results of section 2. In particular, when \(\alpha = 1\), applying Proposition 4, two Gaussian CDF calls are needed for each of the \(d\) components, leading to \(2d\) evaluations. Besides, from Equation (14), the second CDF call does not depend on \(k\), which implies that this term is common for every dimension. Thus the gradient of Equation (19) finally comes with \(d+1\) CDF evaluations instead of \(2d\). For a full gradient with respect to all \(q\) points of the batch, we then need \(q(d+1)\) CDF evaluations – a substantial improvement compared to the \(O(q^2)\) obtained in [29] and the \(O(q^3)\) obtained in the previous section. The complexities for computing moments, \(q\)-EI and its gradients, expressed in terms of number of calls to the \(\Phi\) function, are summarized in Table 1. These new computational savings come at the price of a non-exact gradient calculation. A first numerical validation is represented in Figure 2. On this example, we observe small \((1 \times 10^{-2})\) relative errors between the exact and approximate gradient of dimension \(q \times d = 4\) (the biggest difference vector has a norm of 0.13, compared to an exact gradient norm of 13.1). We also observe that the relative error appears to be typically smaller with higher \(q\)-EI, which is promising for \(q\)-EI maximisations. However, this apparently trustful but non-exact calculation naturally raises the question of the impact of such an approximation on the performances of gradient-based \(q\)-EI maximization algorithms. As we will see in the next section, this proxy gradient turned out to enable quite competitive \(q\)-EI maximization performances based on numerical experiments.

4 Application

The goal of this section is to illustrate the usability of the proposed gradient-based \(q\)-EI maximization schemes and in particular the improvements brought by the fast formulas detailed in the previous
Figure 2: Numerical validation of the approximation from equation (19), with $\alpha = 1$, $q = 2$, $d = 2$. From left to right: 1) Norm of the $q$-EI gradient, with respect to the first batch point (the other point is fixed in the center of $[0, 1]^d$) ; 2) Norm of the difference vector between the analytical gradient and its approximation ; 3) Relative error (norm of the difference divided by the real norm) computed on 3000 random batches sampled uniformly in $[0, 1]^{d \times q}$, with respect to their $q$-EI.

Table 1: In noiseless settings, total number of calls to the CDF of the multivariate Gaussian distribution for computing $\mathcal{M}_{k,1}$, $q$-EI, their gradients and their approximations, depending on $q$ and $d$. For $q$-EI in noisy setting, replace $q$ by $p = n + q$ and multiply each number of calls by $n$. 

<table>
<thead>
<tr>
<th></th>
<th>$\Phi_{q-3}$</th>
<th>$\Phi_{q-2}$</th>
<th>$\Phi_{q-1}$</th>
<th>$\Phi_q$</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{M}_{k,1}$</td>
<td>analytic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>tangent moment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q$-EI</td>
<td></td>
<td>$q$</td>
<td></td>
<td></td>
<td>$O(q)$</td>
</tr>
<tr>
<td></td>
<td>analytic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>tangent moment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$O(q^2)$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$O(q^2)$</td>
</tr>
<tr>
<td>$\nabla \mathcal{M}_{k,1}$</td>
<td>analytic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>tangent moment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>proxy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\nabla q$-EI</td>
<td>analytic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>tangent moment</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>proxy</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$O(qd)$</td>
</tr>
</tbody>
</table>

$O(q)$, $O(q^2)$, $O(q^3)$, $O(q^4)$, $O(q^d)$. 
The relevance of using sequential sampling strategies based on the \( q \)-EI maximization has already been investigated before (see, [7, 47, 29]) and all these articles pointed out the importance of calculation speed which often limits the use of \( q \)-EI based strategies to moderate \( q \). We do not aim again at proving the performance of \( q \)-EI based sequential strategies. Instead we aim at illustrating the gain, in computation time, brought by the fast formulas and show that using the approximate gradient obtained in Equation (19) does not impair the ability to find batches with (close to) maximal \( q \)-EI.

### 4.1 Objective function and pure calculation speed

The objective function is the so-called Borehole function [21]. It has been previously used for testing methods using a surrogate model [48, 20]. The function computes a rate of water flow, \( \phi \), through a borehole. The problem is described by \( d = 8 \) input variables, \( r_w \in [0.05, 0.15], r \in [100, 50000], T_u \in [63070, 115600], H_u \in [990, 1110], T_i \in [63.1, 116], H_i \in [700, 820], L \in [1120, 1680], K_w \in [1500, 15000] \) and is given below

\[
\phi = \frac{2\pi T_u(H_u - H_i)}{\ln \left( \frac{r}{r_w} \right) \left( 1 + \frac{2L T_u}{\ln \left( \frac{r^2}{r_w^2} \right) K_w} + \frac{T_u}{T_i} \right)}.
\]

(20)

Here, the objective function \( f \) is obtained by rescaling \( \phi \) on the input domain \( D = [0, 1]^8 \). An analytical study of variations shows that there is a unique global minimum at \( \mathbf{x}^* = (0, 1, 0, 0, 0, 1, 1, 0)^T \), with \( f(\mathbf{x}^*) \approx 1.1918 \).

Before using sequential strategies to minimize \( f \), we look at empirical computation times for evaluating \( q \)-EI and its gradient as a function of the batch size \( q \). For the computations, the so-called “analytic” method relies on the state of the art formulas of [7, 29] with a number of calls to the multivariate normal cdf of respectively \( O(q^2) \) and \( O(q^4) \). The “tangent moment” method uses our formula for moment calculation to yield \( q \)-EI and its gradient (see, Equations (14) and (15),(16)). Finally, for computing the gradient only, the “proxy” method relies on Equation (19).
Figure 3 exhibits computation times averaged over 1000 batches drawn uniformly. The Gaussian process model is based on an initial design of \( n_0 = 10d = 80 \) points drawn from an optimum-LHS procedure \[25\]. We use the Matérn \((\nu = 3/2)\) tensor product covariance function and estimate the hyperparameters by maximum likelihood using the DiceKriging R package \[11\]. Figure 3 shows significant computational savings. For instance with \( q = 8 \), one gradient computation takes respectively 0.04s, 0.33s and 1.33s using respectively the proxy, the tangent moment and analytic methods. Since the complexity for computing a gradient with the proxy is of \( O(qd) \) against \( O(q^2) \) and \( O(q^4) \) for the two other methods, the computational savings of the proxy tend to increase with \( q \). It should also be noted that these savings will be larger with decreasing domain dimension \( d \). If we look at \( q \)-EI computations, the tangent moment method is 3.3 times faster than the analytic one when \( q = 8 \) and 6.5 times faster when \( q = 20 \); thanks to an \( O(q) \) complexity against \( O(q^2) \).

4.2 Experimental setup: sequential minimization strategies

We now perform a total of 50 minimizations of \( f \), each using an initial design of experiments of \( n_0 = 80 \) points drawn from an optimum-LHS procedure with a different seed. Three different batch-sequential strategies are investigated.

The first one – serving as a benchmark – is a variation of the “Constant Liar Mix” heuristic \[7, 47\] where, at each iteration, the batch of size \( q \) is chosen among several batches obtained from the Constant Liar heuristic \[17\] with different lie levels. We use 7 lie levels fixed to the current maximum observation the current minimum observation, and the 2.5%, 10%, 50%, 90%, 97.5% quantiles of the conditional distribution of the point selected in the batch. A total of 7 batches are proposed at each iteration and the CL-mix heuristic picks the one with maximum \( \text{q-EI} \).

The two other strategies considered here rely on pure \( \text{q-EI} \) maximization using a multistart BFGS algorithm with a stopping criterion of precision \( 2.2 \times 10^{-7} \) (parameter \texttt{control$\$factr} of the R function \texttt{optim} \[37\]). The gradients involved in the optimization are computed either with the tangent moment formula or the proxy. For the gradient-based \( \text{q-EI} \) maximization, we use a total of 10 starting batches obtained, again, using a Constant Liar heuristic with random lies sampled from the conditional distribution at the selected point. Finally we use two different batch sizes. When \( q = 8 \) we run a total of 10 iterations and when \( q = 4 \) we run 20 iterations. The hyperparameters of the GP model are re-estimated at each iteration after having incorporated the new observations.

4.3 First \( \text{q-EI} \) maximization

We first compare the performances, in terms of \( \text{q-EI} \), of the multistart BFGS algorithm when the proxy gradient and the tangent moment methods are used. Table 2 compares the results at iteration 1 for these two methods and the CL-mix strategy. The results are averaged over the 50 initial designs.

<table>
<thead>
<tr>
<th></th>
<th>( q = 4 )</th>
<th>( q = 8 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>tangent moment</td>
<td>12.45 (22.6 s)</td>
<td>15.35 (700.2 s)</td>
</tr>
<tr>
<td>proxy</td>
<td>12.46 (14.3 s)</td>
<td>15.35 (127.0 s)</td>
</tr>
<tr>
<td>CL-mix</td>
<td>11.80 (7.7 s)</td>
<td>14.34 (15.6 s)</td>
</tr>
</tbody>
</table>

Table 2: Average \( \text{q-EI} \) value of the optimal batches found for each of the 50 initial designs. The numbers between brackets are the average computation times.

As expected, the CL-mix heuristic yields batches with lower \( \text{q-EI} \) than the strategies directly maximizing \( \text{q-EI} \). Also, for both \( q = 4 \) and \( q = 8 \), the two \( \text{q-EI} \) based methods have the same performance, which stresses out the relevance of the proxy method since the latter is about 1.6 times faster when \( q = 4 \) and 5.5 faster when \( q = 8 \).
4.4 Several $q$-EI maximization steps

We now compare the performances of the different $q$-EI maximization approaches after multiple batch evaluations. Figure 4 displays the average regret as a function of the iteration number (first row) and the total computation time (i.e. the time to evaluate $f$ and find the next batch to evaluate) assuming respectively that the computation time of $f$ is 0 seconds (i.e. instantaneous), two minutes and one hour (rows 2, 3, 4 respectively). Looking at the performances as a function of the iteration number (first row on Figure 4), the CL-mix heuristic, which samples a batch with lower $q$-EI at each step, leads in average to a slower convergence than the two other methods, for both $q = 4$ and $q = 8$. In contrast, the two strategies based on $q$-EI maximization have similar performances.

However, these conclusions do not hold when the regret is plotted as a function of the total computation time (rows 2, 3, 4 on Figure 4). First, when the computation time $t_{\text{eval}}$ of $f$ is null (row 2) it is clear that $q$-EI-based sequential strategies are not adapted since they are too expensive. In this case, the CL-mix heuristic performs better and some other optimization strategies which are not metamodel-based would probably be more relevant. Second, when $f$ is moderately expensive (i.e. $t_{\text{eval}} = 2$ minutes), the proxy method and CL-mix have comparable performances when $q = 8$, but the proxy outperforms when $q = 4$. Besides, the proxy shows a much faster convergence than the tangent moment method when $q = 8$. The use of $q$-EI based strategies thus becomes relevant when $t_{\text{eval}}$ is larger than a few minutes, if the proxy is used. Finally, when $t_{\text{eval}}$ is equal to one hour, the use of $q$-EI based strategies is particularly recommended. In that case the relative improvement of the proxy compared to the tangent moment method tends to naturally vanish because of the long computation time of $f$. When $f$ is extremely expensive to compute, using the proxy is thus not essential. However, since it does not impair the ability to find a batch with large $q$-EI we still recommend to use it, especially when $q$ is large.

Conclusion

In this article we provide a closed-form expression of generalized $q$-points Expected Improvement criterion for batch-sequential Bayesian global optimization. An interpretation based on moments of truncated Gaussian vectors yields fast $q$-EI formulas with arbitrary precision. Furthermore an new approximation for the gradient is shown to be even faster while preserving ability to find batches close to maximal $q$-EI. As the use of these strategies was previously considered cumbersome from a dozen of batch points, these formulas happen to be of particular interest to run $q$-EI based batch-sequential strategies for larger batch sizes. We show that these methods are implementable and efficient on a classic 8-dimensional test case. Additionally, some of the intermediate results established here might be of interest for other research questions involving moments of truncated Gaussian vectors and their gradients. Perspectives include deriving second order derivatives of $q$-EI and fast numerical estimates thereof. Also, we aim at improving the sampling of initial batches in multistart derivative-based $q$-EI maximization.

Acknowledgements: Part of this work has been conducted within the frame of the ReDice consortium, gathering industrial (CEA, EDF, IFPEN, IRSN, Renault) and academic (École des Mines de Saint-Étienne, INRIA, Universität Bern) partners around advanced methods for computer experiments.

A Differentiating multivariate Gaussian CDF

We consider the CDF dimension $p \geq 2$. We use the convention $\Phi_0 = 1$. 
Figure 4: Log-scaled average regret of the three considered optimization strategies as a function of the iteration number (row 1) and the total computation time (rows 2, 3, 4) assuming that the computation times of $f$, $t_{\text{eval}}$, are respectively 0 seconds, 2 minutes and 1 hour. Experiments are performed with $q = 4$ (left column) and $q = 8$ (right column).
A.1 Gradient

Using the following identity, derived from conditional distributions of a Gaussian vector,
\[
\forall i, 1, \ldots, p, \quad \varphi_{p, \Sigma}(x) = \varphi_{1, \Sigma_{i,i}}(x_i) \varphi_{p-1, \Sigma_{i,i}}(x_i - m_{i,i,x_i}),
\]
with \(m_{i,i,u} = \frac{\varphi_{i} - \Sigma_{i,i} \varphi_{p-1, \Sigma_{i,i}}}{\Sigma_{i,i}}\), and \(\Sigma_{i} = \Sigma_{i,i} - \frac{1}{\Sigma_{i,i}} \Sigma_{i,i} \Sigma_{-i,i}^\top\), we reformulate the integral of the Gaussian CDF:
\[
\forall i = 1, \ldots, p, \quad \Phi_{p, \Sigma}(x) = \int_{-\infty}^{x_i} \varphi_{1, \Sigma_{i,i}}(u_i) \Phi_{p-1, \Sigma_{i,i}}(x_i - u_i - m_{i,i,u}) \, du_i.
\]

Here indexed minus symbols, e.g. in \(\Sigma_{-i,i}\), refer to exclusions of a line or a column.

Finally we have
\[
\nabla \Phi_{p, \Sigma}(x) = \left( \varphi_{1, \Sigma_{i,i}}(x_i) \Phi_{p-1, \Sigma_{i,i}}(x_i - m_{i,i,x_i}) \right)_{i=1, \ldots, p}. \tag{21}
\]

A.2 Hessian

As for the computation of the gradient, we write
\[
\forall i, j = 1, \ldots, p, \quad i \neq j,
\]
\[
\Phi_{p, \Sigma}(x) = \int_{-\infty}^{x_i} \int_{-\infty}^{x_j} \varphi_{2, \Sigma_{i,j,i}}(u_{i,j}) \Phi_{2, \Sigma_{i,j}}(x_{-i,j} - m_{i,j}(u_{i,j})) \, du_i \, du_j,
\]
with \(m_{i,j}(u_{i,j}) = \Sigma_{-ij} \Sigma_{ij}^{-1} \left[ u_{i,j} \right]\) and \(\Sigma_{ij} = \Sigma_{-ij} \Sigma_{ij}^{-1} \Sigma_{-ij}^\top\).

So \(\forall i, j = 1, \ldots, p, \quad i \neq j\),
\[
\frac{\partial^2 \Phi_{p, \Sigma}}{\partial x_i \partial x_j}(x) = \varphi_{2, \Sigma_{i,j,i}}(x_{i,j}) \Phi_{2, \Sigma_{i,j}}(x_{-i,j} - m_{i,j}(x_{i,j})).
\]

When \(i = j\), the differentiation of equation (21) gives,
\[
\frac{\partial^2 \Phi_{p, \Sigma}}{\partial x_i^2}(x) = -\frac{1}{\Sigma_{ii}} \left( \frac{\partial \Phi_{p, \Sigma}}{\partial x_i}(x) + \sum_{j=1}^{p} \Sigma_{ij} \frac{\partial^2 \Phi_{p, \Sigma}}{\partial x_i \partial x_j}(x) \right).
\]

B Moments of truncated multivariate Gaussian distribution

B.1 Analytical formula (propositions 1 and 6)

We see here why we can derive an analytical formula of \(M_{k,\alpha}(m, \Sigma)\), with \(k \leq s \in \mathbb{N} \setminus \{0\}, \quad m \in \mathbb{R}^s\) and \(\Sigma \in S_{++}^s\), by differentiating \(G\), defined in equation (6). It is known, see e.g. [10], that moments can be obtained differentiating the moment generating function \(G_{m, \Sigma, s}\):
\[
M_{k,\alpha}(m, \Sigma) = \frac{\partial^\alpha G_{m, \Sigma, s}}{\partial t_k^\alpha}(0),
\]
with, for \(r \in \{1, \ldots, s\}\), \(G_{m, \Sigma, r} : t \to \mathbb{E} \left( \exp \left( t \Sigma \right) \mathbf{1}_{((x_1, \ldots, x_s) \leq 0)} \right)\), \(Z \sim \mathcal{N}(m, \Sigma)\). We derive now an analytical formula for \(G_{m, \Sigma, r}\). As needed in Proposition 6, we derive an analytical formula for any
We differentiate here the equation (8) with respect to $B$. Differentiation with respect to mean and covariance

With respect to the mean $m$

$$G_{m,\Sigma,r}(t) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp(t^T z)\varphi_{\Sigma}(z - m) \, dz_1 \ldots \, dz_r$$

$$= \varphi_{\Sigma}(0) \int_{-\infty}^{0} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}((z - m)^T \Sigma^{-1}(z - m) - 2t^T z)\right) \, dz$$

$$= e^{-\frac{1}{2}(t + \Sigma^{-1}m)^T \Sigma(t + \Sigma^{-1}m) + m^T \Sigma^{-1}m}$$

$$\varphi_{\Sigma}(0) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-\frac{1}{2}((z - m - \Sigma t)^T \Sigma^{-1}(z - m - \Sigma t))^T} \, dz$$

$$= e^{\frac{1}{2}(t + \Sigma^{-1}m)^T \Sigma(t + \Sigma^{-1}m) - m^T \Sigma^{-1}m} \Phi_{r,(\Sigma_{ij})_{i,j\leq r}}\left(-m - (\Sigma_{ij})_{i\leq r\leq s} t\right). \quad (22)$$

In the frame of the proof of Proposition 6,

- if $\Sigma_k(u, v)$, the covariance matrix of $(Z(v)^T, Z_k(u)^T)$, is positive definite, we take $M_{u,v} = t \rightarrow G_{(m(v), m_k(u), \Sigma_k(u,v), p)((0, \ldots, 0, t)^T),}$

- else, as $\Sigma(v)$ is definite positive, there exists only one index $k_0$ such as $Z_k(u) = Z_{k_0}(v)$ almost surely (for example $k = k_0$ when $u = v$), and we have $M_{u,v} = t \rightarrow G_{(m(v), \Sigma(v), p)((0, \ldots, t, \ldots, 0)^T)}^{k_0\text{th position}}.$

In both cases, equation (22) leads to equation (18).

### B.2 Differentiation with respect to mean and covariance

We differentiate here the equation (8) with respect to $m$ and $\Sigma$.

With respect to the mean $m$

$$\frac{\partial M_{k,1}}{\partial m}(m, \Sigma) = \Phi_{p,\Sigma}(-m)e_k - m_k \nabla \Phi_{p,\Sigma}(-m) + \nabla^T \Phi_{p,\Sigma}(-m) \Sigma_k. \quad (23)$$

With respect to the covariance $\Sigma$

$$\frac{\partial M_{k,1}}{\partial \Sigma}(m, \Sigma) = m_k \frac{\partial \Phi_{p,\Sigma}(-m)}{\partial \Sigma} - \sum_{i=1}^{p} \Phi_{\Sigma,ii}(-m_i) \Phi_{p-1,\Sigma_{ii}}(-m_{\cdot i}) E^{(k,i)}$$

$$+ \Sigma_{ki} \frac{\partial \Phi_{\Sigma,ii}(-m_i) \Phi_{p-1,\Sigma_{ii}}(-m_{\cdot i}) E^{(i,i)}}{\partial \Sigma_{ii}}$$

$$+ \Sigma_{ki} \frac{\partial \Phi_{\Sigma,ii}(-m_i) \Phi_{p-1,\Sigma_{ii}}(-m_{\cdot i})}{\partial \Sigma}.$$ 

$$\quad (24)$$
with \( m_i = m_{-i} - \frac{m_i}{m_{-i}} \Sigma_{-i,i} \) and \( \Sigma_i = \Sigma_{-i,i} - \frac{1}{\Sigma_{-i,i}} \Sigma_{-i,i}^T \Sigma_{-i,i} \). Writing \( d\Sigma[m_i] d\Sigma[\Sigma_i] \) the differential of the functions \( \Sigma \rightarrow m_i \) and \( \Sigma \rightarrow \Sigma_i \), we have:

\[
d\Sigma[m_i](H) = \frac{m_i}{\Sigma_{ii}} H_{-i,i} \quad \text{(25)}
\]

\[
d\Sigma[\Sigma_i](H) = H_{-i,-i} + \frac{H_{ii}}{\Sigma_{ii}} \Sigma_{-i,i} \Sigma_{-i,i}^T - \frac{2}{\Sigma_{ii}} H_{-i,i} \Sigma_{-i,i}^T \quad \text{(26)}
\]

\[
\frac{\partial}{\partial \Sigma} \Phi_{p-1,\Sigma,i}(-m_i) = \sum_{r=1}^{p} \sum_{s=1}^{p} \left( -d\Sigma[m_i](E^{(r,s)}), \nabla \Phi_{p-1,\Sigma,i}(-m_i) \right) + tr \left( \frac{\partial}{\partial \Gamma} \Phi_{p-1,\Sigma,i}(-m_i) d\Sigma[\Sigma_i](E^{(r,s)}) \right) E^{(r,s)}
\]

with:

- \( E^{(r,s)} = (\delta_{ij})_{i,j=1,...,p} \),
- \( \frac{\partial}{\partial \Gamma} \Phi_{p-1,\Sigma,i}(-m_i) \) the derivative of \( \Gamma \rightarrow \Phi_{p-1,\Gamma}(-m_i) \) evaluated at \( \Sigma_i \). We use the Plackett's differential equation, extended by [5], to find

\[
\frac{\partial}{\partial \Gamma} \Phi_{p-1,\Sigma,i}(-m_i) = \nabla \nabla^T \Phi_{p-1,\Sigma,i}(-m_i),
\]

\( \nabla \nabla^T \Phi \) is given in appendix A.2.

**C Generalized q-EI as a sum of moments**

**Proof.** For given \( (\ell, k) \) in \( \{1, \ldots, n\} \times \{1, \ldots, q\} \), we consider \( E_{\ell,k} \) the event that the random variable inside the expectation term of equation (4) equals \( (Y(x_\ell) - Y(x_{n+k}))^n \). We have

\[
E_{\ell,k} = \{ Y(x_{n+k}) \leq Y(x_\ell) \} \cap \{ \forall i \leq n, i \neq \ell; Y(x_\ell) \leq Y(x_i) \} \cap \{ \forall j \leq q, j \neq k; Y(x_{n+k}) \leq Y(x_{n+j}) \}
\]

Considering all pairs \( (\ell, k) \), we have:

\[
EI_n(x_{n+1:n+q}) = \sum_{\ell=1}^{n} \sum_{k=1}^{q} E_n ((Y(x_\ell) - Y(x_{n+k}))^n 1_{(E_{\ell,k})}).
\]

For each term \( (\ell,k) \) of the sum, the conditioning event can be rewritten \( E_{\ell,k} = \{ Z^{(\ell,k)}(x_{n+1:n+q}) \leq 0 \} \), with \( Z^{(\ell,k)} \) a random vector of size \( n + q - 1 \), defined by the following linear transformation of \( Y = (Y(x_1), \ldots, Y(x_{n+q}))^T \):

\[
\forall i = 1, \ldots, n + q - 1, \quad Z^{(\ell,k)}_i = \begin{cases} Y_\ell - Y_i & \text{if } 1 \leq i \leq \ell - 1 \\ Y_{\ell+1} - Y_i & \text{if } \ell \leq i \leq n - 1 \\ Y_{n+k} - Y_{i+1} & \text{if } n \leq i \leq n + q - 1, i \neq n + k - 1 \\ Y_{n+k} - Y_\ell & \text{if } i = n + k - 1 \end{cases}
\]

Indeed, the first \( n - 1 \) components of \( Z^{(\ell,k)} \leq 0 \) reflect \( \{ \forall i \leq n, i \neq \ell; Y(x_\ell) \leq Y(x_i) \} \), and the last components reflect \( \{ \forall j \leq q, j \neq k; Y(x_{n+k}) \leq Y(x_{n+j}) \} \) and \( \{ Y(x_{n+k}) \leq Y(x_\ell) \} \). \( \square \)
D Mean square differentiability of $Y(x)^\alpha 1\{B\}$

Let $B$ be an event, $Y$ be a mean-squared differentiable Gaussian process and $\alpha \in \mathbb{N}$. Then we have:

$$\mathbb{E}\left( \left( \frac{(Y(x+h)^\alpha - Y(x)^\alpha}{h} 1\{B\} - \frac{dY^\alpha}{dx}(x) 1\{B\} \right)^2 \right) \leq \mathbb{E}\left( \left( \frac{(Y(x+h)^\alpha - Y(x)^\alpha}{h} - \frac{dY^\alpha}{dx}(x) \right)^2 \right) \xrightarrow{h \to 0} 0$$

by mean-squared differentiability of $Y^\alpha$.

E Symmetry argument

The term $\frac{q(q+1)}{2}$ comes from a symmetry occurring when summing terms with different index but actually equal. At fixed summation index $\ell$ in (10), we denote $\omega_{ki}$ the $i$th term in the scalar product in (8) for each $M_{m+k-1,1}$ required for $q$-EI:

$$\forall i, k = 1, \ldots, q, \; \omega_{ki} = \Sigma_{\ell i}^{(\ell, k)} \left[ \nabla \Phi_{p, \Sigma(\ell)} \left( -m_{\ell i}^{(\ell, k)} \right) \right]_i.$$ 

Then the following symmetry between indices $i$ and $k$ occurs:

$$\forall i, k = 1, \ldots, q, \; \frac{\omega_{ki}}{\Sigma_{\ell i}^{(\ell, k)} \varphi_{\Sigma(\ell)}^{i, i}(\Sigma^{(\ell, k)})} = \frac{\omega_{kj}}{\Sigma_{\ell k}^{(\ell, k)} \varphi_{\Sigma(\ell)}^{k, k}(\Sigma^{(\ell, k)})}$$

Indeed, using the formula of the derivative of CDF, (appendix A.1), leads to:

$$\frac{\omega_{ki}}{\Sigma_{\ell i}^{(\ell, k)} \varphi_{\Sigma(\ell)}^{i, i}(\Sigma^{(\ell, k)})} = \Phi_{p, \Sigma(\ell)} \left( -m_{\ell i}^{(\ell, k)} \right) = \mathbb{P} \left( Y(x_{i+n+k}) \leq Y(x_i), Y(x_{i+n+j}) \leq Y(x_{i+n+k}), \forall j = 1 \ldots q, j \neq k, j \neq i \right) \frac{Y(x_{i+n+i})}{Y(x_{i+n+k})},$$

which is clearly symmetrical between $i$ and $k$.

References


