Bayesian optimization constrained by gray-box conservation law with unknown flux functions

Han Chen, Qiqi Wang

1 Abstract

Many engineering applications can be formulated as optimizations constrained by conservation laws. Such optimizations can be solved by the adjoint method, which computes the gradient of the objective to the design variables. Traditionally, the adjoint method has not been able to be implemented in many "gray-box" conservation law simulators. In gray-box simulators, the analytical and the numerical form of the conservation law is unknown, though the full solution of relevant flow quantities is available. We have introduced the twin model method to estimate the gradient by using the gray-box simulator's space-time solution. In this paper, we develop a Bayesian optimization framework that uses both the objective and its estimated gradient to improve optimization performance. We also give theoretical results for the method's convergence. The mthod is demonstrated in the optimization of several numerical examples, showing superior optimization performance.

2 Background

Optimization problems are of great interest in the engineering community. We consider an optimization problem to be constrained by conservation laws. For example, oil reservoir simulations may employ PDEs of various flow models, in which different fluid phases and components satisfy a set of conservation laws. Such simulations can be used to facilitate the oil reservoir management, including optimal well placement [4] and optimal production control [2, 3]. Another example is the cooling of turbine airfoils. We are interested in optimizing the interior flow path of turbine airfoil coolant to minimize pressure loss [5, 6].

We are interested in the optimization problem of the following form

$$\begin{aligned} &\min_{c\in\mathcal{C}} \xi(u,c) \\ &\text{where } u \text{ satisfies } \ \mathcal{R}(u,c)=0 \\ &\text{subject to } c\in[0,1]^n \end{aligned} \tag{1}$$

where $t\in[0,T]$, $x\in\Omega$. $u(t,\mathbf{x})$ is the space-time solution of a conservation law PDE abstracted as $\mathcal{R}$, with given initial and boundary conditions. The design variable is c.

In many cases, the simulation of $\mathcal{R}(u,c)=0$ can be computationally costly, potentially due to the complex computational models involved, and the large-scale time and space discretization. Furthermore, the dimensionality of the design space, d, can be high. A tool to enable efficient high-dimensional optimization is adjoint sensitivity analysis [8-11] , which efficiently computes $\frac{d\xi}{dc}\in\mathbb{R}^d$, the gradient of the objective to the design variables.

Many conservation law simulators do not have the adjoint implemented. Besides, we are not able to implement the adjoint method when the governing PDE and its numerical implementation is unavailable: for example, when the source code is proprietary or legacy. However, many conservation law simulators can provide the space-time solution as an output. If the simulation solves for time-independent problems, the simulation can provide the steady-state spatial solution. When a simulator satisfies such conditions, we call the simulator "gray-box" [1].

To enable the adjoint computation for gray-box simulations, we proposed the twin model method [1]. The method infers the gray-box simulator's governing PDE, $\mathcal{R}(u,c)=0$, by using its space-time or spatial solution. The inferred model, $\tilde{\mathcal{R}}(\tilde{u},c)=0$, is called the "twin model". We can simulate the twin model to evaluate $\tilde{\xi}=\xi(\tilde{u},c)$ and its adjoint. The twin model's gradient, $\frac{d\tilde{\xi}}{dc}$, can be used as an estimation for $\frac{d\xi}{dc}$.

We are able to evaluate ξ from the gray-box simulation, and to estimate $\frac{\partial\xi}{\partial c}$ from the twin model's adjoint. The optimization of ξ fits naturally into a multi-fidelity optimization (MFO) framework, where the gray-box and the twin-model simulators can be viewed as two models of different fidelity. MFO methods can be categorized into local and global approaches:

Local MFO methods aim at finding a local optimum in the search domain, including the pattern-search and the trust-region MFO methods. In pattern search methods, the objective function J is evaluated at a set of trial points, known as the 'mesh', adjacent to the current best design point in the design space. If the objective improves, the current best design is updated; otherwise, the mesh size is shrinked [19]. Booker et al. [20] extends the pattern search method to MFO and proves its convergence. In trust region methods, a surrogate is constructed in the neighborhood of the current best design, known as the trust region. Then the surrogate is optimized within the trust-region to generate the next candidate design point. Depending on the availability of the gradient samples, the surrogate can be either constructed using only the function value data [36, 37, 38], or constructed also using the gradient data [39, 40].

On the other hand, global MFO methods aim at finding the global optimum in the search domain. An important class of global MFO methods is based on the Bayesian modelling of the objective function. Mockus et al. [41] assumes the objective function is sampled from a Gaussian process. A posterior distribution for the function is maintained and updated when new data is sampled. The posterior distribution is used to pick the next sample point. Popular choices include the expected improvement (EI) method [41, 36] and the upper confidence bound (UCB) method [25]. It has been shown both methods converges to the optimum under mild assumptions of the objective function [23, 24, 25]. When multi-fidelity models are available, Kennedy et al. [32] calibrates the model inadequecy between different models by GP, and constructs the posterior using samples from all the models. When gradient samples are available, the posterior can be constructed using the co-Kriging method [30, 31].

Bayesian optimization uses all available model evaluations to construct the posterior. In many cases, the conservation law simulation is computationally much more costly than the optimization algorithm.

Therefore, we aim at reducing the number of model evaluations, while neglecting the expense of running the optimization algorithm. A good choice is Bayesian optimization, which has been shown to outperform other state of art global optimization algrithms in reducing the number of model evaluations.

In this paper, we propose an adjoint-based optimization framework for gray-box conservation law simulation by combining the twin model method and the Bayesian MFO. The Bayesian optimization uses both the objective function evaluated by the gray-box simulation and the gradient estimated by the twin model method. In the remainder of the paper, Section 3 reviews the definition of the gray-box simulation and the twin model method. Section 4 discusses the Bayesian optimization, especially Gaussian-process (GP) optimization. We are going to discuss two GP optimization methods: the expected improvement (EI) method and the upper confidence bound (UCB) method. Section 5 combines the Bayesian optimization methods with the twin model method. The proposed optimization framework takes samples from both the gray-box model's objective function and the twin model's estimated gradient. We also give the convergence proof for the optimization framework. Section 6 demonstrate the optimization framework on three numerical testcases: (1) maximizing a 2-D Rosenbrock function, (2) optimizing the control of a 1-D controlled porous media flow, and (3) optimizing the return bend geometry of a 2-D Navier-Stokes flow.

3 The gray-box simulation and the twin model method

Many conservation law simulations are gray-box. By gray-box, we mean a conservation law simulation without the adjoint method implemented. Furthermore, we are not able to implement the adjoint method when the governing PDE for the conservation law and its numerical implementation is unavailable: for example, when the source code is proprietary or legacy. Another defining property of gray-box simulation is that it can provide the space-time solution of the conservation law. If the simulation solves for time-independent problems, a gray-box simulation should be able to provide the steady state solution. In contrast, we define a simulator to be a blackbox, if neither the adjoint nor the solution is available. The only output of such simulations is the value of the objective function to be optimized. If the adjoint method is implemented or is able to be implemented, we call such simulations open-box. We summarize their differences in Table 1.

Table 1: Comparison of black-box, gray-box, and open-box simulations

	PDE and implementation	Space (or space-time) solution	Adjoint
Black-box	✘	✘	✘
Gray-box	✘	✔	✘
Open-box	✔	✔	✔

For example, consider a gray-box simulation for the conservation law $\mathcal{R}(u, c) = 0$:

$$\dot{\boldsymbol{u}} + \nabla \cdot \overset{\rightharpoonup\rightharpoonup}{\boldsymbol{F}}(\boldsymbol{u}) = \boldsymbol{q}(\boldsymbol{u}, c) \tag{2}$$

where $\boldsymbol{u}$ is a vector representing flow quantities, $\dot{\boldsymbol{u}}$ is the derivative of $\boldsymbol{u}$ with respect to time, $t \in [0, T]$. c represents the design variables, and $x \in \Omega \subseteq \mathbb{R}^n$ is the spatial coordinate. Ω may depend on the design variables, c. $\overset{\rightharpoonup\rightharpoonup}{\boldsymbol{F}}$ is the flux tensor. The exact form of $\overset{\rightharpoonup\rightharpoonup}{\boldsymbol{F}}$ is unknown. $\boldsymbol{q}$ is a source vector that may also depend on c. The boundary and initial conditions are known. The discretized space-time solution of Eqn.(2) given by the simulator is written as $\hat{u}(t_i, \mathbf{x}_i; c)\,, i = 1, \cdots, N$, where $t = \{t_1, \cdots, t_N\}$ indicates the discretized time, and $\mathbf{x}_i$ indicates the spatial discretization at time t_i. Because the flux is unknown, the adjoint method is not able to be applied to Eqn.(2) directly.

To enable the adjoint method for the gray-box simulation, we developed the twin model method to infer the PDE. Firstly we proposed a parameterized PDE, the "twin model",

$$\dot{\tilde{\boldsymbol{u}}} + \nabla \cdot G(\tilde{\boldsymbol{u}}, \eta) = \boldsymbol{q}(\tilde{\boldsymbol{u}}, c) \tag{3}$$

with a flux G parameterized by $\eta \in \mathbb{R}^s$. Given the same inputs (design variables, initial conditions, and boundary conditions), a twin model should yield a space-time solution $\tilde{\boldsymbol{u}}$ close to $\boldsymbol{u}$. Suppose that the twin model and the primal model use the same discretization; we use the following expression to quantify the solution mismatch:

$$\mathcal{M} = \frac{1}{T} \sum_{i=1}^{N} \sum_{k=1}^{M} \left(\tilde{\boldsymbol{u}}_{ik} - \boldsymbol{u}_{ik}\right)^2 \Delta t_k \left|\Delta \mathbf{x}_i\right| \tag{4}$$

where Δt_k indicates the kth time step, and $|\Delta \mathbf{x}_i|$ indicates the lengths (1-D), areas (2-D), or volumes (3-D) of the grid.

We obtain the optimal η by

$$\eta^* = \arg\min_{\eta} \{\mathcal{M} + \lambda \|\eta\|^p\} \tag{5}$$

where $\tilde{\boldsymbol{u}}$ is the discretized space-time solution of the twin model. $\lambda\|\eta\|^p$ is an L_p norm regularization, and $\lambda > 0$.

With η^* inferred, we obtain a twin model simulating

$$\dot{\tilde{\boldsymbol{u}}} + \nabla \cdot G(\tilde{\boldsymbol{u}}, \eta^*) = \boldsymbol{q}(\tilde{\boldsymbol{u}}, c) \tag{6}$$

Given a design c, we can obtain $\tilde{u}(c)$ and $\tilde{\xi} \equiv \xi(\tilde{u}, c)$. Because the twin model is an open box, we are able to evaluate $\frac{d\tilde{\xi}}{dc}$ by the adjoint method, which gives an estimate for $\frac{d\xi}{dc}$.

4 Bayesian and Gaussian process optimization

Bayesian optimization works by firstly assuming the objective function to be sampled from a stochastic process. Let $\xi : \mathcal{C} \to \mathbb{R}$ be the objective function defined on the search space $\mathcal{C}$, and let Ω be a sample space, Σ be a σ-algebra over Ω, and $\mathbb{P}$ be a probability measure. A stochastic process is a function

$$\begin{aligned} \xi \ &: \mathcal{C} \times \Omega \to \mathbb{R} \\ (c, \omega) &\to \xi(c, \omega) \equiv \omega(c) \end{aligned} \tag{7}$$

that for any $c \in \mathcal{C}$, the function $\xi(c, \cdot)$ is a random variable on $(\Omega, \Sigma, \mathbb{P})$. To simplify the notation, we denote $\xi(c, \omega)$ by $\omega(c)$.

In Bayesian optimization, ξ is considered as a sample function $\xi(\cdot, \omega)$ from the stochastic process. The prior distribution of the objective function is taken into account by a probability measure P of the sample paths. The prior is update at each search step when when new data is sampled, through the computation of the posterior. Let $\mathcal{F}_n$ be the σ-algebra generated by $\xi(c_1), \cdots, \xi(c_N)$. We denote the posterior of ξ to be $P[\cdot|\mathcal{F}_n]$. The posterior is used to choose the next sample point. Let $\mathbb{R}^{\mathcal{C}}$ be the power set of functions mapping from $\mathcal{C}$ to $\mathbb{R}$. Bayesian optimization maps $\mathbb{R}^{\mathcal{C}}$ to a search sequence in $\mathcal{C}$:

$$\underline{C}(\xi) := (c_1(\xi), c_2(\xi), \cdots) \tag{8}$$

with the Markov property that, $c_{N+1}(\xi)$ depends only on the previous samples $\xi(c_1), \cdots, \xi(c_N)$.

In order to choose c_{N+1} from the previous samples, Bayesian optimization introduces an acquisition function, $\rho : \mathcal{C} \to \mathbb{R}^+$, which evaluates the expected utility of investing the next sample at c given the posterior $P[\cdot|\mathcal{F}_n]$. The next sample point should be chosen to maximize the acquisition function.

$$c_{N+1} \in \arg\max_{c \in \mathcal{C}} \rho(c) \tag{9}$$

There are several popular criterions for the acquisition function, such as the expected improvement (EI) [29] and the upper confidence bound (UCB) [25]. Let c_N^* be a best design of the existing sample points, i.e.

$$c_N^* \in \arg\max_{c \in \underline{c}_N} \xi(c) \tag{10}$$

EI uses the acquisition function:

$$\rho_{\mathrm{EI}}(c) = \mathbb{E}\left[\max\left(\xi(c) - \xi(c_N^*), 0\right)|\mathcal{F}_n\right] , \tag{11}$$

where the expectation is taken on the posterior. UCB uses the acquisition function

$$\rho_{UCB}(c) = \mathbb{E}[\xi(c)|\mathcal{F}_n] + \kappa\sigma[c|\mathcal{F}_n] , \tag{12}$$

where σ indicates the standard deviation. κ is a tunable parameter to balance exploitation against exploration. In this paper, we will focus on both the GP-EI and GP-UCB methods.

Gaussian process is a special case of stochastic processes. For any design points $c_1, \cdots, c_N$, the random variables defined by

$$\begin{aligned} g_i &= \xi(c_i, \cdot) : \omega \in \Omega \to \mathbb{R} \\ \mathbf{g} &= (g_1, \cdots, g_N) \end{aligned} \tag{13}$$

has a joint Gaussian distribution. GP is solely determined by its mean function $m(c)$ and its covariance function $K(c, c')$.

$$\begin{aligned} m(c) &= \mathbb{E}[\xi(c)] \\ K(c, c') &= \mathbb{E}[(\xi(c) - m(c))(\xi(c') - m(c'))] , \end{aligned} \tag{14}$$

for $\forall\, c, c' \in \mathcal{C}$. Such a GP can be written as $\xi \sim \mathcal{N}(m(c), K(c, c'))$. GP allows us to compute conditionals in closed form. Assume we have evaluated ξ at $\underline{c}_n = \{c_1, \cdots, c_n\}$, then $P[\xi(c)|\mathcal{F}_n]$ can be constructed from the joint distribution

$$\begin{pmatrix} \xi(c) \\ \xi(\underline{c}_n) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} m(c) \\ m(\underline{c}_n) \end{pmatrix}, \begin{pmatrix} K(c,c) & K(c,\underline{c}_n) \\ K(\underline{c}_n, c) & K(\underline{c}_n, \underline{c}_n) \end{pmatrix} \right) \tag{15}$$

Conditioned on the samples $\xi(\underline{c}_n)$, the posterior is still a GP, with the mean and the variance being

$$\begin{aligned} \mathbb{E}\left[\xi(c)|\, \xi(\underline{c}_n)\right] &= m(c) + K(c,\underline{c}_n) K(\underline{c}_n,\underline{c}_n)^{-1} \left(\xi(\underline{c}_n) - m(\underline{c}_n)\right) \\ Var\left[\xi(c)|\, \xi(\underline{c}_n)\right] &= K(c,c) - K(c,\underline{c}_n) K(\underline{c}_n,\underline{c}_n)^{-1} K(\underline{c}_n, c) \end{aligned} \tag{16}$$

The closed form posterior of GP is very useful computationally, and we will use GP to model the objective function in this paper. Without loss of generality, we model the prior to have zero mean and to be stationary, i.e. $m(c) = 0$, and $K(c, c') = K(c - c', 0)$.

GP is able to express a rich distribution of functions, depending on the choice of the covariance K. Example covariance functions include the squared exponential kernel and the Matern kernels. For a detailed review of such covariances, see [28].

For GP optimization, the acquisition function is computed solely from the posterior mean μ and variance σ. Specifically, Eqn.(11) has an analytical form

$$\rho_{EI}(c) = \sigma(c) \left[\left(\frac{\mu(c) - f(c_N^*)}{\sigma(c)} \right) \Phi\left(\frac{\mu(c) - f(c_N^*)}{\sigma(c)} \right) + \phi\left(\frac{\mu(c) - f(c_N^*)}{\sigma(c)} \right) \right] \tag{17}$$

where ϕ is the PDF, Φ is the CDF of standard normal distribution.

5 GP optimization with the estimated gradient

Assume the gray-box simulator evaluates the objective function ξ accurately, and assume ξ to be differentiable. We consider optimizing ξ by sampling the gray-box simulator's ξ and the twin model's estimated gradient $\frac{d\tilde{\xi}}{dc}$. The type of error introduced by the gray-box model's gradient estimation is treated as model inadequacy. The notion of model inadequacy is introduced by Kennedy and O'Hagan

[32]. as the "difference between the true value and the code output". In our problem, we assume the gray-box simulator provides a deterministic space-time solution given an input c. In other words, the twin model and its adjoint is solely determined by the space-time solution, thus determined by c. Review: Kenedy Ohagan bayesian calibration, DACE model jones,

The deterministic error ϵ can be treated as the model inadequacy. The model inadequacy can be modeled as an additive term [jones 1998],

$$\nabla\tilde{\xi}(c) = \nabla\xi(c) + \epsilon(c)\,, \tag{18}$$

ϵ is a d-dimension vector representing the error of estimating $\nabla\xi(c)$ by the twin model. The components of ϵ are modelled as a realization of a GP with zero mean:

$$\begin{pmatrix} \epsilon_i(c) \\ \epsilon_j(c') \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, G_{ij}(c,c')\right)\,, \tag{19}$$

for $i, j = 1, \cdots, d$. Let $\mathbf{G}$ be the d-by-d matrix with the i, jth components being G_{ij}. For simplicity, we assume no correlation between different components. In other words, $\mathbf{G}$ is modeled as a diagonal matrix.

$$\mathbf{G}(c,c') = \begin{pmatrix} G_1(c,c') & & \\ & \ddots & \\ & & G_d(c,c') \end{pmatrix}\,, \tag{20}$$

This assumption reduces the modeling cost of the model discrepancy.

Furthurmore, we assume the model inadequacy to be uncorrelated with the gray-box model, i.e.

$$\begin{aligned} cov(\nabla\xi(c_1), \epsilon(c_2)) = 0 \\ cov(\xi(c_1), \epsilon(c_2)) = 0\,, \end{aligned} \tag{21}$$

for any c_1, $c_2 \in \mathcal{C}$.

Let

$$\xi(\underline{c}_n) = (\xi(c_1), \cdots, \xi(c_N)) \tag{22}$$

be the sampled function value,

$$\xi_\nabla(\underline{c}_n) = (\nabla\xi(c_1), \cdots, \nabla\xi(c_N)) \tag{23}$$

be the true function gradient, and

$$\xi_{\tilde{\nabla}}(\underline{c}_n) = \Big(\nabla\tilde{\xi}(c_1), \cdots, \nabla\tilde{\xi}(c_N)\Big) \tag{24}$$

be the sampled function gradient. From Eqn.(18) and (21), the joint distribution of $\xi(c)$ and the sampled data is given by

$$\begin{pmatrix} \xi(c) \\ \xi(\underline{c}_n) \\ \xi_{\tilde{\nabla}}(\underline{c}_n) \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} 0 \\ \mathbf{0} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} K(c,c) & \mathbf{v} & \mathbf{w} \\ \mathbf{v}^T & \mathbf{D} & \mathbf{H} \\ \mathbf{w}^T & \mathbf{H}^T & \mathbf{E} + \bar{\mathbf{G}} \end{pmatrix} \right), \tag{25}$$

where

$$\begin{aligned}
\mathbf{v} &= (K(c,c_1), \cdots, K(c,c_N)) \\
\mathbf{w} &= (\nabla_{c_1} K(c,c_1), \cdots, \nabla_{c_N} K(c,c_N)) \\
\mathbf{D} &= \begin{pmatrix} K(c_1,c_1) & \cdots & K(c_1,c_N) \\ \vdots & \ddots & \vdots \\ K(c_N,c_1) & \cdots & K(c_N,c_N) \end{pmatrix} \\
\mathbf{H} &= \begin{pmatrix} \nabla_{c'_1} K(c_1,c'_1) & \cdots & \nabla_{c'_N} K(c_1,c'_N) \\ \vdots & \ddots & \vdots \\ \nabla_{c'_1} K(c_N,c'_1) & \cdots & \nabla_{c'_N} K(c_N,c'_N) \end{pmatrix} \\
\mathbf{E} &= \begin{pmatrix} \nabla_{c_1}\nabla_{c'_1} K(c_1,c'_1) & \cdots & \nabla_{c_1}\nabla_{c'_N} K(c_1,c'_N) \\ \vdots & \ddots & \vdots \\ \nabla_{c_1}\nabla_{c'_N} K(c_N,c'_1) & \cdots & \nabla_{c_N}\nabla_{c'_N} K(c_N,c'_N) \end{pmatrix} \\
\bar{\mathbf{G}} &= \begin{pmatrix} \mathbf{G}(c_1,c'_1) & & \\ & \ddots & \\ & & \mathbf{G}(c_N,c'_N) \end{pmatrix}
\end{aligned} \tag{26}$$

v is a row vector with length N. w is a vector with length Nd, $\mathbf{D}$ is a N-by-N matrix, $\mathbf{H}$ is a N-by-Nd matrix, and $\mathbf{E}, \bar{\mathbf{G}}$ are a Nd-by-Nd matrices. $\underline{c}_n = \underline{c}'_n$. Using Eqn.(25), we can obtain the posterior distribution of $\xi(c)$.

GP depends on the form of the covariance functions K and G. These functions imply assumptions on the continuity and the smoothness for the realizations of the GP. A popular choice for the covariance functions is the Matern kernel K_ν, where ν is a parameter controlling the smoothness of the GP realizations. For example, a popular choice is $\nu = \infty$, which corresponds to the square exponential kernel. The square exponential kernel is

$$K(c,c') = \sigma^2 \exp\left(-\frac{\|c-c'\|^2}{2L^2}\right) , \tag{27}$$

where σ^2 is the variance, L is the characteristic length scale, $\|\cdot\|$ indicates the L-2 norm. The realizations of GP with the square exponential kernel is infinitely differentiable. Another choice, $\nu = 5/2$, corresponds to the Matern 5/2 kernel:

$$K(c,c') = \sigma^2 \left(1 + \frac{\sqrt{5}\|c-c'\|}{L} + \frac{5\|c-c'\|^2}{3L^2}\right) \exp\left(-\frac{\sqrt{5}\|c-c'\|}{L}\right) , \tag{28}$$

The realizations of the GP with the Matern 5/2 kernel is once differentiable. In this paper, we will consider both kernels.

The kernels K and $G_1, \cdots, G_d$ depend on their parameters such as the variance and the correlation length, also known as the hyperparameters. Denote these hyperparameters by θ. These hyperparameters can be selected by the maximum marginal likelihood method [36]. The method uses a point estimate of the hyperparameters in order to maximize the likelihood of observing the sampled data. Given θ, the likelihood of observing $\xi(\underline{c}_n)$ and $\xi_{\tilde{\nabla}}(\underline{c}_n)$ is :

$$\begin{aligned} p\left(\xi(\underline{c}_n), \xi_{\tilde{\nabla}}(\underline{c}_n)|\theta\right) &= \int p(\xi(\underline{c}_n), \xi_{\tilde{\nabla}}(\underline{c}_n), \xi_{\nabla}(\underline{c}_n)|\theta) dY_{\nabla} \\ &= \int p\left(\xi(\underline{c}_n), \xi_{\nabla}(\underline{c}_n)|\theta\right) p\left(\xi_{\tilde{\nabla}}(\underline{c}_n)|\xi(\underline{c}_n), \xi_{\nabla}(\underline{c}_n); \theta\right) d\xi_{\nabla}(\underline{c}_n) , \end{aligned} \tag{29}$$

The marginalization is done over $\xi_{\nabla}(\underline{c}_n)$. Under the GP assumption, we have

$$\xi(\underline{c}_n), \xi_{\nabla}(\underline{c}_n)\Big|\theta \sim \mathcal{N}\left(\begin{pmatrix}\mathbf{0}\\ \mathbf{0}\end{pmatrix}, \begin{pmatrix}\mathbf{D} & \mathbf{H}\\ \mathbf{H}^T & \mathbf{E}\end{pmatrix}\right) \tag{30}$$

and

$$\xi_{\tilde{\nabla}}(\underline{c}_n)|\xi(\underline{c}_n), \xi_{\nabla}(\underline{c}_n); \theta \sim \mathcal{N}\left(\xi_{\nabla}(\underline{c}_n), \bar{\mathbf{G}}\right) \tag{31}$$

Eqn.(30) and (31) yields a closed form of the log marginal likelihood:

$$\begin{aligned} &\log p(\xi(\underline{c}_n), \xi_{\tilde{\nabla}}(\underline{c}_n)|\theta) = \\ &-\frac{1}{2}\begin{pmatrix}\xi(\underline{c}_n)\\ \xi_{\tilde{\nabla}}(\underline{c}_n)\end{pmatrix}^T \begin{pmatrix}\mathbf{D} & \mathbf{H}\\ \mathbf{H}^T & \mathbf{E}+\bar{\mathbf{G}}\end{pmatrix}^{-1} \begin{pmatrix}\xi(\underline{c}_n)\\ \xi_{\tilde{\nabla}}(\underline{c}_n)\end{pmatrix} - \frac{1}{2}\log\left(\det\begin{pmatrix}\mathbf{D} & \mathbf{H}\\ \mathbf{H}^T & \mathbf{E}+\bar{\mathbf{G}}\end{pmatrix}\right) - \frac{N(d+1)}{2}\log(2\pi) \end{aligned} \tag{32}$$

where the matrix inversion can be performed by Cholesky decomposition. The closed form of the marginal likelihood enables efficient optimization of the hyperparameters. In this paper, we will use this method to select the hyperparameters. Using the formulation discussed above, we can evaluate the posterior of $\xi(c)$. The posterior is then used to evaluate the acquisition function, and to obtain the next optimal design point. In addition, we include a stopping rule to our optimization algorithm. If the maximum expected improvement is less than a threshold value of the current best function value, we stop [36].

6 Convergence properties

In this section, we give some results on the convergence properties of the GP optimization framework discussed in the previous section. Researchers have explored the convergence properties of GP optimization using only the objective function sampling. M. Locatelli [33] proves that the search sequence produced by the GP-EI method is dense in $\mathcal{C}$ for $n \to \infty$ for the 1-D optimization problem $c^* = \max_{c \in [0,1]} \xi(c)$, if ξ is a realization of the Wienner process. E. Vazquez [23] generalizes the results, by showing that the sequence is still dense for higher dimensional space and for a general class of stochastic processes. Further, A. Bull [24] provides a convergence rate at $\mathcal{O}(n^{-\nu/d})$ for the GP-EI method, where $\nu > 0$ is a constant parameter controlling the smoothness of the RKHS. Similar results are also given for GP-UCB. N. Srinivas [25] bounds the convergence rate from above at $n^{-\frac{\nu}{2\nu+d(d+1)}}$ for GP-UCB, and also establish the relationship between the convergence rate and the information gain due to sampling. In this section, we will analyze the convergence properties of GP-EI when the estimated gradient sampling is also available. Under the assumptions in section5, our main result is: 1. GP-EI optimization using the estimated gradient produces a dense search sequence when $n \to \infty$. (2. GP-UCB optimization using the estimated gradient converges no slower than without using the estimated gradient.)

We first reiterate some notations and assumptions. Without loss of generality [21] , we assume the objective function to be a realization of a Gaussian process with zero mean. Specifically, the objective function belongs to the reproducing kernel Hilbert space $\mathcal{H}_K$ generated by the kernel $K : \mathcal{C} \times \mathcal{C} \to \mathbb{R}^+ \bigcup 0$. Assume K to be differentiable, then the gradient of all functions in $\mathcal{H}_K$ forms a reproducing kernel Hilbert space $\mathcal{H}_{K_\nabla}$ with the kernel $K_\nabla(c_1, c_2) \equiv \nabla_{c_1} \nabla_{c_2} K(c_1, c_2)$ for all $c_1, c_2 \in \mathcal{C}$ [22]. The gradient estimation error is defined as in Eqn.(19) and Eqn.(20). In other words, the ith component ($i = 1, \cdots, d$) of ϵ belongs to the reproducing kernel Hibert space $\mathcal{H}_G^i$ generated by the kernel $G_i : \mathcal{C} \times \mathcal{C} \to \mathbb{R}^+ \bigcup 0$, and the components are mutually independent. Denote $\mathcal{H}_G \equiv \mathcal{H}_G^1 \otimes \cdots \otimes \mathcal{H}_G^d$.

Denote the stochastic dependence of ξ by ω_ξ, and the stochastic dependence of ϵ_i by ω_ϵ^i for $i = 1, \cdots, d$. Let $(\Omega_\xi, \Sigma_\xi, \mathbb{P}_\xi)$ be the probability space for ω_ξ, and let $(\Omega_\epsilon^i, \Sigma_\epsilon^i, \mathbb{P}_\epsilon^i)$ be the probability space for ω_ϵ^i for $i = 1, \cdots, d$. We have

$$\begin{aligned} \xi \ &: \mathcal{C} \times \Omega_\xi \to \mathbb{R} \\ &(c, \omega_\xi) \to \xi(c; \omega_\xi) \end{aligned} \tag{33}$$

$$\begin{aligned} \epsilon \ &: \mathcal{C} \times \Omega_\epsilon^i \to \mathbb{R}^d \\ &(c, \omega_\epsilon^i) \to \epsilon(c; \omega_\epsilon^i) \end{aligned} \qquad \text{for } i = 1, \cdots, d \tag{34}$$

Let $\omega_\epsilon = (\omega_\epsilon^1, \cdots, \omega_\epsilon^d)$ and $\Omega_\epsilon = \Omega_\epsilon^1 \otimes \cdots \otimes \Omega_\epsilon^d$. The true objective function is modelled as $\xi(c; \omega_\xi^*)$ for $\omega_\xi^* \in \Omega_\xi$, and the true estimated gradient error is modelled as $\epsilon(c; \omega_\epsilon^*)$ for $\omega_\epsilon^* \in \Omega_\epsilon$. In other words $\xi(c; \omega_\xi^*) = \xi(c)$ and $\epsilon(c; \omega_\epsilon^*) = \epsilon(c)$. Conditioned on the existing samples, GP optimization generates the next search point deterministically. We denote the search sequence $(c_n)_{n \geq 1}$ generated from the optimization strategy by $\underline{c}_n$. We also denote the function value, the estimated gradient sample, the true gradient, and the gradient estimation error on $\underline{c}_n$ by $\xi(\underline{c}_n)$, $\xi_{\tilde{\nabla}}(\underline{c}_n)$, $\xi_\nabla(\underline{c}_n)$, and $\epsilon(\underline{c}_n)$. Given the initial design c_0, the search sequence can be seen as a mapping

$$\underline{C}(\omega_\xi, \omega_\epsilon) = (C_1(\omega_\xi, \omega_\epsilon), C_2(\omega_\xi, \omega_\epsilon), \cdots) \tag{35}$$

The search strategy $\underline{C}$ generates a random search sequence $C_1, C_2, \cdots$ in $\mathcal{C}$, with the property that C_{n+1} is $\mathcal{F}_n$-measurable, where $\mathcal{F}_n$ is the σ-algebra generated by $\xi(\underline{c}_n)$ and $\xi_{\tilde{\nabla}}(\underline{c}_n)$.

At the n-th search step in the optimization algorithm, the posterior mean and variance of $\xi(c)$ conditioned on $\xi(\underline{c}_n)$ and $\xi_{\tilde{\nabla}}(\underline{c}_n)$ are written as

$$\hat{\xi}_n(c; \underline{c}_n) = \mathbb{E}_{\omega_\xi, \omega_\epsilon} \left[\xi(c, \omega_\xi) \middle| \underline{c}_n, \xi(\underline{c}_n), \xi_{\tilde{\nabla}}(\underline{c}_n) \right] , \tag{36}$$

and

$$\sigma_n^2(c; \underline{c}_n) = \mathbb{E}_{\omega_\xi, \omega_\epsilon} \left[\left(\xi(c) - \hat{\xi}_n(c) \right)^2 \middle| \underline{c}_n, \xi(\underline{c}_n), \xi_{\tilde{\nabla}}(\underline{c}_n) \right] . \tag{37}$$

Notice $\sigma_n^2(c;\underline{c}_n)$ only depends on $\underline{c}_n$, and is independent of $\xi(\underline{c}_n), \xi_{\tilde{\nabla}}(\underline{c}_n)$ because of the GP assumption.

6.1 GP-EI generates a dense search sequence

In this section, we proves that the GP-EI algorithm generates a dense search sequence in the design space. The proof is similar to the proof given by E. Vazquez [23]. The contribution of this paper is to extend the proof to the case where the estimated gradient sample is also available.

Firstly, we have (Chapter 1, Theorem 4.1, [27]).

Lemma 1 *Let* K_1, K_2 be the reproducing kernels of functions on $\mathcal{C}$ with norms $\|\cdot\|_{\mathcal{H}_1}$ and $\|\cdot\|_{\mathcal{H}_2}$ respectively. Then $K = K_1 + K_2$ is the reproducing kernel of the space

$$\mathcal{H} = \mathcal{H}_1 \oplus \mathcal{H}_2 = \{f = f_1 + f_2,\ f_1 \in \mathcal{H}_1,\ f_2 \in \mathcal{H}_2\}$$

with norm $\|\cdot\|_{\mathcal{H}}$ defined by

$$\forall f \in \mathcal{H} \quad \|f\|_{\mathcal{H}}^2 = \min_{f=f_1+f_2,\, f_1\in\mathcal{H}_1, f_2\in\mathcal{H}_2} \left(\|f_1\|_{\mathcal{H}_1^2} + \|f_2\|_{\mathcal{H}_2}^2\right)$$

Using Lemma 1, We prove the following inequality:

Theorem 1

$$\left|\xi(c,\omega_\xi) - \hat{\xi}(c;\underline{c}_n)\right|^2 \le \left(\left(1+\frac{4d}{3}\right)\|\xi(c;\omega_\xi)\|_{\mathcal{H}_K} + \frac{4d}{3}\|\nabla_c\xi(c;\omega_\xi)\|_{\mathcal{H}_{K_\nabla}} + \frac{4}{3}\sum_{i=1}^{d}\left\|\epsilon_i(c;\omega_\epsilon^i)\right\|_{\mathcal{H}_G^i}\right)\sigma^2(c;\underline{c}_n)$$

Proof of Theorem 1 Define a vector

$$u = (u_1, \cdots, u_d)^T\ , \tag{38}$$

where $u_1 \in [0,1], \cdots, u_d \in [0,1]$. We write the domain for u as $\mathcal{U}$. Define an auxiliary function

$$F(c,u;\omega_\xi,\omega_\epsilon) = \left(1 - \sum_{i=1}^{d} u_i\right)\xi(c,\omega_\xi) + u^T\left[\nabla_c\xi(c,\omega_\xi) + \epsilon(c;\omega_\epsilon)\right]\ , \tag{39}$$

$u_1, \cdots, u_d$ can be viewed as realizations of a reproducing kernel Hilbert space $\mathcal{H}_u$ on $\mathcal{U} = [0,1]$. For example, $\mathcal{H}_u$ can be the Sobolev space $W^{1,2}$ defined on $\mathcal{U}$, equipped with the inner product

$$\langle\phi,\psi\rangle = \int_{\mathcal{U}} \phi\psi + (\nabla\phi)^T(\nabla\psi)\,du \tag{40}$$

and the reproducing kernel

$$K_u(\phi, \psi) = \frac{1}{2} \exp\left(- |\phi - \psi|\right) \tag{41}$$

Given ω_ξ and ω_ϵ, $F(c, u; \omega_\xi, \omega_\epsilon)$ can be viewed as a realization from a reproducing kernel Hilbert space, $\mathcal{H}_F$, defined on $\mathcal{C} \times \mathcal{U}$. Let the kernel function of $\mathcal{H}_F$ be

$$\begin{aligned} K_F \; &: \mathcal{C} \times \mathcal{U}, \mathcal{C} \times \mathcal{U} \to \mathbb{R} \\ &(c_1, u_1), (c_2, u_2) \to K_F((c_1, u_1), (c_2, u_2)) \end{aligned} \tag{42}$$

Notice

$$F(c, \mathbf{0}; \omega_\xi, \omega_\epsilon) = \xi(c, \omega_\xi) \tag{43}$$

is the objective function, and

$$(F(c, e_1; \omega_\xi, \omega_\epsilon), \cdots, F(c, e_d; \omega_\xi, \omega_\epsilon)) = \nabla_c \xi(c; \omega_\xi) + \epsilon(c; \omega_\epsilon) \tag{44}$$

is the estimated gradient, where $e_i, i = 1, \cdots, d$ indicates the ith unit Cartesian basis vector in $\mathbb{R}^d$. Conditioned on the sampling $\xi(\underline{c}_n)$ and $\xi_{\tilde{\nabla}}(\underline{c}_n)$, we can bound the error of the estimation of $F(c, \mathbf{0}; \omega_\xi, \omega_\epsilon)$ by the Cauchy-Scharz inequality [27] in $\mathcal{H}_F$,

$$\left| F(c, \mathbf{0}; \omega_\xi, \omega_\epsilon) - \hat{F}(c, \mathbf{0}; \underline{c}_n) \right| = \left| \xi(c; \omega_\xi) - \hat{\xi}_n(c; \underline{c}_n) \right| \leq \sigma(c; \underline{c}_n) \|F\|_{\mathcal{H}_F} \tag{45}$$

Besides,

$$\begin{aligned} \|F\|_{\mathcal{H}_F} &= \left\| \left(1 - \sum_{i=1}^{d} u_i\right) \xi(c; \omega_\xi) + u^T \left[\nabla_c \xi(c; \omega_\xi) + \epsilon(c; \omega_\epsilon)\right] \right\|_{\mathcal{H}_F} \\ &\leq \|\xi(c; \omega_\xi)\|_{\mathcal{H}_K} + \left(\sum_{i=d}^{d} \|u_i\|_{\mathcal{H}_u}\right) \|\xi(c; \omega_\xi)\|_{\mathcal{H}_K} + \left(\sum_{i=d}^{d} \|u_i\|_{\mathcal{H}_u}\right) \|\nabla_c \xi(c; \omega_\xi)\|_{\mathcal{H}_{K_\nabla}} + \sum_{i=1}^{d} \left\| u_i \epsilon_i(c; \omega_\epsilon^i) \right\|_{\mathcal{H}_u \bigotimes \mathcal{H}_G^i} \\ &= \|\xi(c, \omega)\|_{\mathcal{H}_K} + \frac{4d}{3} \|\xi(c, \omega)\|_{\mathcal{H}_K} + \frac{4d}{3} \|\nabla_c \xi(c; \omega_\xi)\|_{\mathcal{H}_{K_\nabla}} + \frac{4}{3} \sum_{i=1}^{d} \left\| \epsilon_i(c; \omega_\epsilon^i) \right\|_{\mathcal{H}_G^i} \end{aligned} \tag{46}$$

The second line of Eqn.(46) uses Lemma 1, and the third line uses the fact that $\mathcal{H}_u$ can be chosen as $W^{1,2}$. Thus we proved Theorem 1.

Using theorem 1, we can prove

Theorem 2 Let $(\underline{c}_n)_{n \geq 1}$ and $(\underline{a}_n)_{n \geq 1}$ be two sequences in $\mathcal{C}$. Assume that the sequence (a_n) is convergent, and denote by a^* its limit. Then each of the following conditions implies the next one:

1. a^* is an adherent point of $\underline{c}_n$ (there exists a subsequence in $\underline{c}_n$ that converges to a^*) ,

2. $\sigma^2(a_n; \underline{c}_n) \to 0$ when $n \to \infty$,

3. $\hat{\xi}(a_n; \underline{c}_n) \to \xi(a^*, \omega)$ when $n \to \infty$, for all $\xi \in \mathcal{H}_K$, $\epsilon \in \mathcal{H}_G$.

The proof of theorem 2 is the same as the proposition 8 in [23], except that the posterior is defined using the estimated gradient sampling. Theorem 2 can be proven by replacing the Cauchy-Schwarz inequality used in [23] by Theorem 1.

For the stationary process $\xi \in \mathcal{H}_K$ defined on the domain $\mathbb{R}^d$, define $\Phi(c) \equiv K(c, 0)$. Let the Fourier transform of $\Phi(c)$ be $\hat{\Phi}(\eta)$. We assume $\hat{\Phi}$ satisfies the property:

Assumption There exist $C \geq 0$ and $k \in \mathbb{N}^+$, such that $(1 + |\eta|^2)^k \big|\hat{\Phi}(\eta)\big| \geq C$ for all $\eta \in \mathbb{R}^d$.

For any $\xi \in \mathcal{H}_K$ and its Fourier transform $\hat{\xi}$, we have [24]

$$\big\|\xi\big\|_{W^{k,2}} = \int (1 + |\eta|^2)^k \big|\hat{\xi}\big|^2 \, d\eta \geq C \int \big|\hat{\Phi}(\eta)\big|^{-1} \big|\hat{\xi}(\eta)\big|^2 \, d\eta = C\sqrt{(2\pi)^d} \big\|\xi\big\|_{\mathcal{H}_K} , \tag{47}$$

where $W^{k,2}$ is the Sobolev space whose weak derivatives up to order k have a finite L^2 norm. Therefore, $W^{k,2} \subseteq \mathcal{H}_K$. The result can be extended to $\xi \in \mathcal{H}_K(\mathcal{C})$ defined on the domain $\mathcal{C} \in \mathbb{R}^d$, because $\mathcal{H}_K(\mathcal{C})$ embeds isometrically into $\mathcal{H}_K(\mathbb{R}^d)$ [34]. Besides, we have that C_c^∞ is dense in $W^{k,2}$ (Chapter 2, Lemma 5.1 [35]), where C_c^∞ is the C^∞ functions with compact support on $\mathcal{C}$. As a consequence, $\mathcal{C}_c^\infty \subseteq \mathcal{H}_K$ [23].

It can be shown that (3) results in (1) if only $\xi(\underline{c}_n)$ are available [23]. Indeed, if (1) is false, then there exist a neighborhood U of a^* that does not intersect $\underline{c}_n$. There exist $\xi \in \mathcal{H}_K$ that is compactly supported in U. It follows that $\hat{\xi}(a^*; \underline{c}_n) = 0$ whereas $\xi(a^*) \neq 0$. The argument can be extended to the case when both $\xi(\underline{c}_n)$ and $\xi_{\tilde{\nabla}}(\underline{c}_n)$ are available: there exist $\xi = 0$ and $\epsilon = \mathbf{0}$, such that $\hat{\xi}(a^*; \underline{c}_n) = 0$ whereas $\xi(a^*) \neq 0$. Thus (1), (2), and (3) in Theorem 2 are equivalent.

Finally, we have the theorem:

Theorem 3 (E. Vazquez, Theorem 5 [23]) If the 3 conditions in Theorem 2 are equivalent, then for all $c_{init} \in \mathcal{C}$ and all $\omega \in \mathcal{H}$, the sequence $\underline{c}_n$ generated by the GP-EI algorithm is dense in $\mathcal{C}$.

7 Numerical examples

In this section, we demonstrate the optimization method on several numerical examples. The first example is to minimize n-dimensional Rosenbrock functions, the second example is to optimize the control of a 1-D porous media flow problem, and the third example is to optimize the boundary geometry of a 2-D return bend channel.

7.1 Optimize generalized Rosenbrock functions

The inclusion of estimated gradient in addition to the function evaluation can potentially accerlerate the Bayesian optimization. In this numerical example, we apply the framework proposed in section 5 to the minimization of d-D generalized Rosenbrock functions. Generalized Rosenbrock functions are non-convex test functions to check optimization algorithms' performance [42], and it has been used to test Bayesian optimization algorithms [43]. The n-D Rosenbrock function is defined by

$$\xi(\mathbf{x}) = \sum_{i=1}^{N-1} b(x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \,, \tag{48}$$

where $\mathbf{x} = (x_1, \cdots, x_N) \in \mathbb{R}^N$. By convention we set $a = 1, b = 100$. For all $d > 2$, $\xi(\mathbf{x})$ has the global minimum 0 at $\mathbf{x} = (1, 1, \cdots, 1)$. We apply the optimization framework to minimize the d-D Rosenbrock functions. Because no PDE simulation is involved in the evaluation of $\xi(\mathbf{x})$ thus no twin model is involved, we simulate the gradient estimation by Eqn.(18). The covariance of ϵ is sampled by Eqn.(19) and (20), while $G_i, i = 1, \cdots, d$ are assumed to be i.i.d. Gaussian processes with Matern 5/2 kernel (28) with correlation length $L = 1$.

Firstly, we minimize the 2-D Rosenbrock function in $\mathbf{x} \in [-3, 3]^d$, while using various σ^2. Thus we evaluate how the optimization performance is affected by the choise of σ^2. We set σ to be (1) $\sigma = 10$, (2) $\sigma = 100$, and (3) $\sigma = \infty$ respectively. (1) corresponds to a noisy gradient, (2) corresponds to more noisy gradient, and (3) corresponds to no gradient evaluation. Because of the randomness involved in

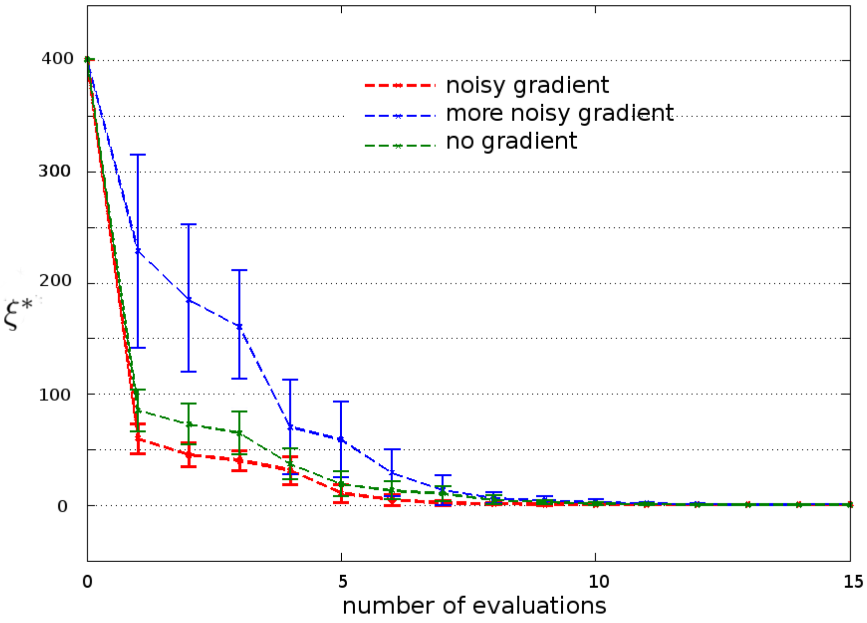

Figure 1: The current best ξ_n^* after n function evaluations. The minimization is performed under 3 scenarios: noisy gradient ($\sigma = 10$), more noisy gradient ($\sigma = 100$), and no gradient ($\sigma = \infty$). The minimization is performed 100 times. This plot shows the averaged ξ^* and the one-sigma error bar for the 100 minimizations.

the optimization of the acquisition function, we optimize ξ for 100 times, and compute the averaged, current best ξ_n^* after n function evaluations. The result is shown in Fig.1. The Bayesian optimization performance improves as the noise in the gradient reduces.

Secondly, we minimize the d-D Rosenbrock functions with the 3 choices of σ, for $d = 2, 3, 4, 5, 6$. We report the average number of function evaluations required in order to obtain $\xi_n^* < 5$ for 100 minimizations for each d. As the dimension d increases, the inclusion of the noisy gradient reduces the number of function evaluations.

In conclusion, the inclusion of noisy gradient data in the Bayesian optimization reduces the number of function evaluations required to achieve the desired accuracy for the d-D Rosenbrock functions, especially for larger d.

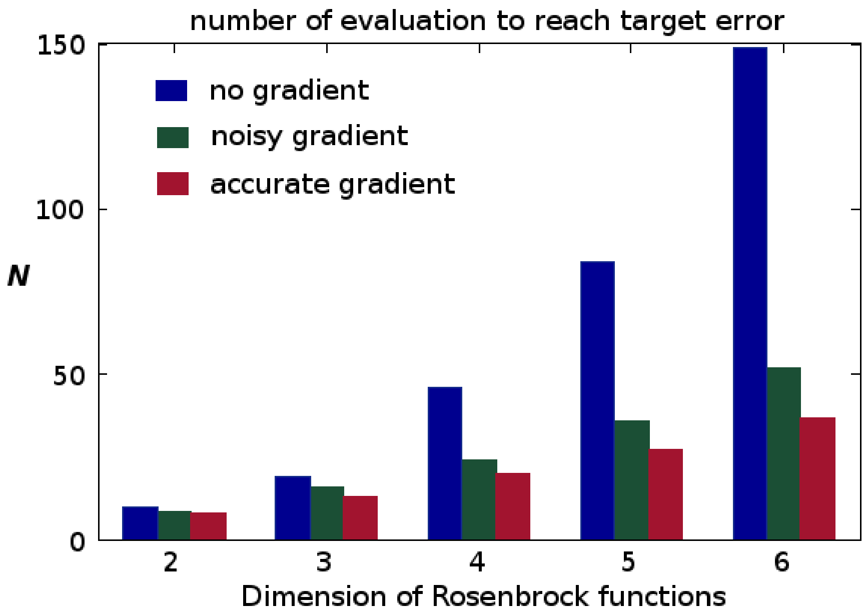

Figure 2: The number of function evaluations required to reduce ξ_n^* to a target threshold ($\xi_n^* < 5$). The minimization is performed under the 3 scenarios $\sigma = 10, 20$, and ∞.

7.2 Optimize a 1-D porous media flow

Consider a PDE

$$\begin{aligned} &\frac{\partial u}{\partial t} + \frac{\partial F(u)}{\partial x} = c(t,x) \quad x \in [0,1] \; t \in [0,1] \\ &u(t=0,x) = u_0(x), \; u(t,x=0) = u(t,x=1) \end{aligned} \tag{49}$$

that models a 1-D, two-phase, porous media flow with periodic boundary condition. u denotes the saturation of one phase, and $1-u$ denotes the saturation of the other phase. $0 \le u \le 1$. $c(t,x)$ is a space-time dependent control which models the phase-1 injection rate. The flux function $F(u)$ depends on the properties of the porous media and the fluids. We optimize $c(t,x)$ such that $u(t=1,x)$ is close to a target function $u^*(x)$.

We parameterize the space-time-dependent control $c(t,x)$ by $\mathbf{c} = (c_{ij})_{i=1,\cdots,m\;,j=1,\cdots,n}$:

$$\begin{aligned} &c(t,x) = \sum_{i=1}^{m} \sum_{j=n}^{s} c_{ij} \exp\left(-(t-t_i)^2/l_t^2\right) \exp\left(-(x-x_j)^2/l_x^2\right) \\ &t \in [0,1], x \in [0,1] \end{aligned} \tag{50}$$

The objective function is given by

$$\xi(c_{ij}) = \int_{x=0}^{1} |u(t=1,x) - u^*(x)|^2 + \lambda \sum_{ij} c_{ij}^2 \,, \tag{51}$$

We set Eqn.(49) to be Buckley-Leverett equation which models the flow driven by capillary pressure and Darcy's law [7], whose flux is

$$F(u) = \frac{u^2}{1 + A(1-u)^2} \,. \tag{52}$$

In the optimization of ξ we assume F is unknown, thus Eqn.(49) is a gray-box model. We have demonstrated that twin model can give accurate gradient estimation for ξ [1].

Specifically, we set $m = s = 5$, $l_t = l_x = 0.25$, $A = 2$, $u^*(x) = 0.5$. $\lambda = 0.01$. t_i's and x_j's are equally spaced in [0,1]. We minimize Eqn.(51) by either using or not using the estimated gradient provided by the twin model, and compare their current best ξ^*'s after the same number of iterations. Fig.4 and 6 show that using the twin model improves the optimization result when the number of gray-box simulations is limited.

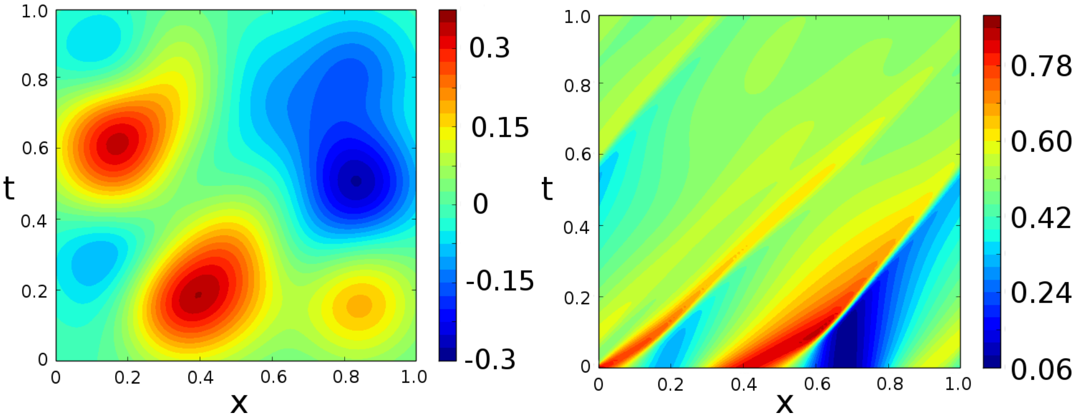

Figure 3: Left: the control $c(t,x)$ optimized using the twin model after 100 gray-box simulations. Right: $u(t,x)$ obtained by the gray-box simulation using the $c(t,x)$ on the left.

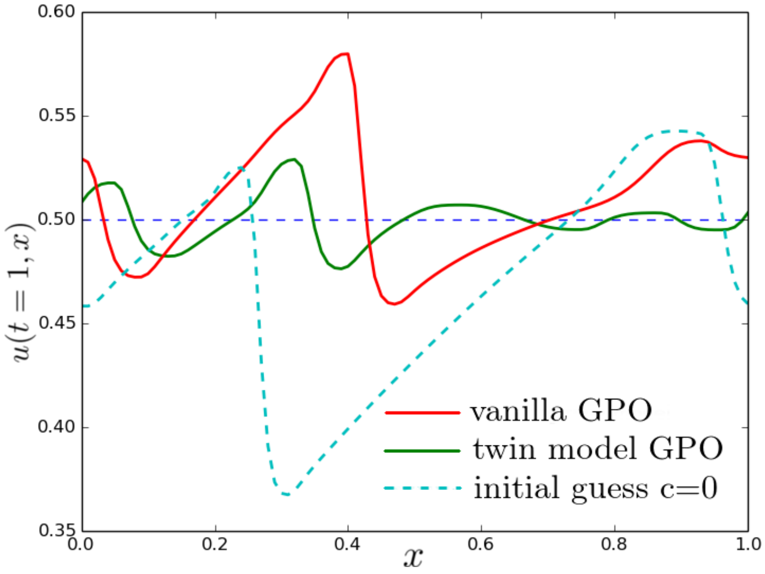

Figure 4: $u(t = 1, x)$. The dashed cyan line indicates the solution obtained by setting $\mathbf{c} = \mathbf{0}$ *(initial guess). The red line is the optimized solution after 100 gray-box simulations without using the twin model. The green line is the optimized solution after 100 gray-box simulations using the twin model. The dashed black line is the target solution $u^*(x)$.*

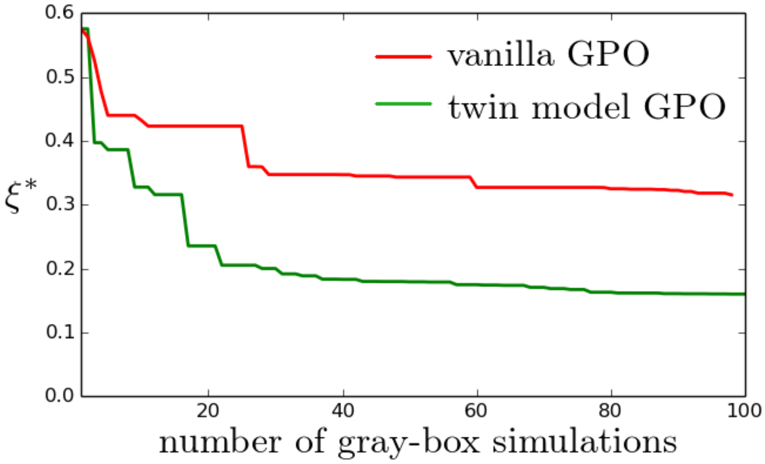

Figure 5: The current best ξ_n^* after n gray-box simulations. The blue line is obtained by using the twin model, and the green line is obtained by using the twin model.

7.3 Optimize a Navier-Stokes flow

We consider a steady-state compressible internal flow in a 2-D return bend channel governed by Navier-Stokes equations.

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho \\ \rho u \\ \rho v \\ \rho E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho u \\ \rho u^2 + p - \sigma_{xx} \\ \rho u v - \sigma_{xy} \\ u(E\rho + p) - \sigma_{xx}u - \sigma_{xy}v\end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix}\rho v \\ \rho u v - \sigma_{xy} \\ \rho v^2 + p - \sigma_{yy} \\ v(E\rho + p) - \sigma_{xy}u - \sigma_{yy}v\end{pmatrix} = \mathbf{0} \tag{53}$$

where

$$\begin{aligned}
\sigma_{xx} &= \mu\left(2\frac{\partial u}{\partial x} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right) \\
\sigma_{yy} &= \mu\left(2\frac{\partial v}{\partial y} - \frac{2}{3}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right) \\
\sigma_{xy} &= \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)
\end{aligned} \tag{54}$$

Given the boundary geometry, the flow simulation provides the steady-state solution of the density ρ, the velocity (u, v), and the energy E. The state equation $p = p(\rho, U)$ is assumed unknown, so the flow simulation is a gray-box model. The details of gray-box simulation and the twin model setup are described in [1].

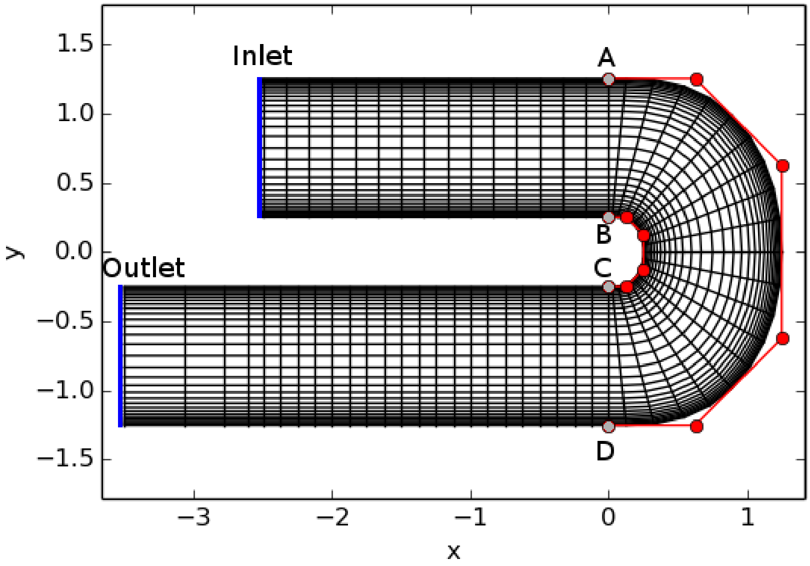

Figure 6: The return bend geometry and the mesh for simulation. The innder and outer boundaries at the bending region are parameterized by the 8 red control points using B-Spline. The area of the bending section (the curved quadrilateral whose vertices are A,B,C,D) is constrained to be a constant in the optimization.

Let the area of the bending section of the return bend to be V. We maximize the cross-sectional mass flow rate

$$\xi = -\int_{\text{outlet}} \rho_\infty u_\infty \big|_{\text{outlet}}\, dy = \int_{\text{inlet}} \rho_\infty u_\infty \big|_{\text{inlet}}\, dy \tag{55}$$

constrained by V being a constant, by tuning the B-Spline control points. To avoid ill-posedness of the mesh, we constrain the coordinates of the control points in the boxes shown in Fig.7.

The twin model has been demonstrated to provide accurate gradient estimation of ξ with respect to the coordinates of the control points [1]. Using the estimated gradient, we obtained faster objective function improvement, as shown in Fig.8 with a limited number of gray-box simulations.

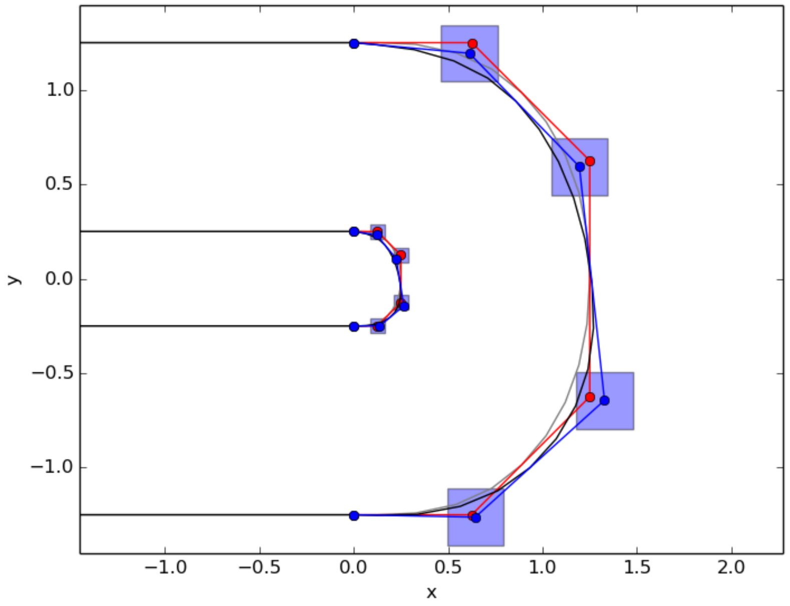

Figure 7: The initial guess and the optimized control points coordinates. The blue points are the initial guesses, and the red are the optimized. The blue boxes indicate the constraints used to bound the control points to avoid ill-posed meshing.

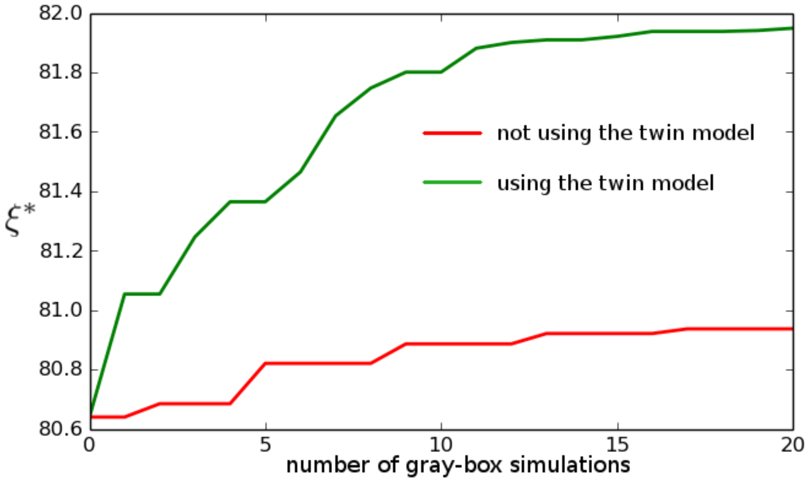

Figure 8: The current best ξ^* after n number of gray-box simulations.

8 Conclusion

This paper presented a Bayesian optimization framework for optimization problems constrained by gray-box PDE simulations that are expensive to evaluate. The gray-box simulations provides the space-time or spatial solution, which permit the twin model inference. The twin model estimates the adjoint of the gray-box simulation and the gradient of the objective function. The estimated gradient, combined with the gray-box objective function evaluation, is used in a Bayesian optimization framework. We showed that the optimization framework generates a dense search sequence under some assumptions of the estimated gradient. The proposed optimization framework is then compared with the Bayesian optimization using only the gray-box simulations. We tested the optimization of the n-D Rosenbrock functions, the 1-D Buckley-Leverett equation, and a 2-D return bend problem. Using the same number of gray-box simulations, the proposed framework outperforms the framework without using the twin model in all these test cases.

9 References

[1] Chen, Han, and Qiqi Wang. "Adjoint-based gradient estimation from gray-box solutions of unknown conservation laws." arXiv preprint arXiv:1511.04576 (2015).

[2] D. Brouwer et al., Dynamic optimization of waterflooding with smart wells using optimal control theory, SPE Journal, volume 9, number 4, 2004

[3] W. F. Ramirez, Application of optimal control theory to enhanced oil recovery, Elsevier, 1987.

[4] M. Zandvliet et al., Adjoint-based well-placement optimization under production constraints, SPE Journal, volume 13, number 4, 2008.

[5] T. Verstraete et al., Optimization of a U-bend for minimal pressure loss in internal cooling channels Part I: Numerical method, Journal of Turbomachinery, volume 135, number 5, 2013.

[6] F. Coletti et al., Optimization of a U-Bend for minimal pressure loss in internal cooling channels Part II: Experimental validation, Journal of Turbomachinery, volume 135, number 5, 2013.

[7] S. E. Buckley et al., Mechanism of fluid displacement in sands, Transactions of the AIME, volume 146, number 1, 1942.

[8] H. Chen, Blackbox stencil interpolation method for model reduction, Master thesis, Massachusetts Institute of Technology, 2012.

[9] H. Chen et al., Data-driven model inference and its application to optimal control under reservoir uncertainty, 14th European Conference of Mathematics of Oil Recovery, 2014.

[10] J. L. Lions, Optimal control of systems governed by partial differential equations, Springer-Verlag, 1971.

[11] A. Jameson, Aerodynamic design via control theory, Journal of Scientific Computing, volume 3, number 3, 1988.

[12] W. H. Chen et al., A new algorithm for automatic history matching, Society of Petroleum Engineering Journal, volume 14, number 6, 1974.

[13] W. F. Ramirez et al., Optimal injection policies for enhanced oil recovery: part 1: theory and computational strategies. Society of Petroleum Engineering Journal, volume 24, number 3, 1984.

[14] J. E. Dennis et al., Quasi-Newton methods, motivation and theory. SIAM Review, volume 19, number 1, 1977.

[15] L. M. Rios et al., Derivative-free optimization: A review of algorithms and comparison of software implementations. Journal of Global Optimization, volume 56, number 3, 2013.

[16] J. Nocedal, Updating quasi-Newton matrices with limited storage, Mathematics of Computation, volume 35, number 151, 1980.

[17] R. Tibshirani, Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society, Series B (Methodological), 1996.

[18] J. W. Murdock, Fundamental fluid mechanics for the practicing engineer, CRC Press, 1993.

[19] Torczon, Virginia. "On the convergence of pattern search algorithms." SIAM Journal on optimization 7.1 (1997): 1-25.

[20] Booker, Andrew J., et al. "A rigorous framework for optimization of expensive functions by surrogates." Structural optimization 17.1 (1999): 1-13.

[21] Rasmussen, C.E. and Williams, C.K.I. "Gaussian Process for Machine Learning." MIT Press, 2006.

[22] Zhou, Ding-Xuan. "Derivative reproducing properties for kernel methods in learning theory." Journal of computational and Applied Mathematics 220.1 (2008): 456-463.

[23] Vazquez, Emmanuel, and Julien Bect, "Convergence properties of the expected improvement algorithm with fixed mean and covariance functions." Journal of Statistical Planning and inference 140.11 (2010): 3088-3095.

[24] Bull, Adam D. "Convergence rates of efficient global optimization algorithms." The Journal of Machine Learning Research 12 (2011): 2879-2904.

[25] Srinivas, Niranjan, et al. "Gaussian process optimization in the bandit setting: No regret and experimental design." arXiv preprint arXiv:0912.3995 (2009).

[26] D.R. Jones. A taxonomy of global optimization methods based on response surfaces. Journal of Global Optimization. 21(4):345-383, 2001.

[27] Berlinet, Alain, and Christine Thomas-Agnan. ”Reproducing kernel Hilbert spaces in probability and statistics.” Springer Science and Business Media, 2011.

[28] Snoek, Jasper, Hugo Larochelle, and Ryan P. Adams. ”Practical Bayesian optimization of machine learning algorithms.” Advances in neural information processing systems. 2012.

[29] Mokus, J. ”On Bayesian methods for seeking the extremum.” Optimization Techniques IFIP Technical Conference. Springer Berlin Heidelberg, 1975.

[30] H. Chung and J. J. Alonso. Using gradients to construct cokriging approximation models for high-dimensional design optimization problems. AIAA Paper, 317:14- 17, 2002.

[31] A. I. J. Forrester, A. S6bester, and A. J. Keane. Multi-fidelity optimization via surrogate modelling. Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science, 463(2088):3251-3269, 2007.

[32] Kennedy, Marc C., and Anthony O’Hagan. ”Predicting the output from a complex computer code when fast approximations are available.” Biometrika 87.1 (2000): 1-13.

[33] Locatelli, Marco. ”Bayesian algorithms for one-dimensional global optimization.” Journal of Global Optimization 10, no. 1 (1997): 57-76.

[34] Aronszajn, Nachman. ”Theory of reproducing kernels.” Transactions of the American mathematical society (1950): 337-404.

[35] Showalter, Ralph E.. ”Hilbert space methods for partial differential equations.” Courier Corporation, 2010.

[36] Jones, Donald R., Matthias Schonlau, and William J. Welch. ”Efficient global optimization of expensive black-box functions.” Journal of Global optimization 13.4 (1998): 455-492.

[37] Conn, Andrew R., Katya Scheinberg, and Lus N. Vicente. "Global convergence of general derivative-free trust-region algorithms to first-and second-order critical points." SIAM Journal on Optimization 20.1 (2009): 387-415.

[38] Wild, Stefan M., and Christine Shoemaker. "Global convergence of radial basis function trust-region algorithms for derivative-free optimization." SIAM Review 55.2 (2013): 349-371.

[39] Alexandrov, Natalia M., et al. "A trust-region framework for managing the use of approximation models in optimization." Structural Optimization 15.1 (1998): 16-23.

[40] Carter, Richard G. "Numerical experience with a class of algorithms for nonlinear optimization using inexact function and gradient information." SIAM Journal on Scientific Computing 14.2 (1993): 368-388.

[41] J Mockus, V Tiesis, and A Zilinskas "The application of Bayesian methods for seeking the extreme." Towards Global Optimization, 2 (1978): 117-129

[42] H. H. Rosenbrock "An automated method for finding the greatest or least value of a function." The Computer Journal, 3 (1960)L: 175-184

[43] Lam, Remi, Allaire, Douglas L., and Willcox, Karen E. "Multifidelity Optimization using Statistical Surrogate Modeling for Non-Hierarchical Information Sources" 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference Kissimmee, Florida