Optimization of Computationally Expensive Simulations with Gaussian Processes and Parameter Uncertainty

Jing Xie, Peter I. Frazier, Sethuraman Sankaran, Alison Marsden, Saleh Elmohamed

School of Operations Research & Information Engineering, Cornell University
Department of Mechanical and Aerospace Engineering, University of California San Diego
Department of Biomedical Engineering, Center for Applied Mathematics, Cornell University

October 14, 2012
INFORMS Annual Meeting 2012
Phoenix, Arizona
Shape Optimization of An Idealized Bypass Graft Model

Target anastomosis angles given to the surgeon: $x = (x_1, x_2)$.

Actual angles constructed in a surgery: $\theta = (\theta_1, \theta_2) = x + \delta$, where $\delta = (\delta_1, \delta_2)$ are the implementation errors introduced during surgery.

Stenosis radius $r$ and inflow velocity $v$ are environmental variables.

The area of low wall-shear stress (WSS) given $\theta$ and $\omega = (r, v)$ is $f(\theta, \omega)$.

$f$ can be evaluated exactly through expensive simulation.

Utility function with optional risk aversion: $U = -f$ or $U = e^{-\alpha \cdot f}$ ($\alpha > 0$).

The joint probability density of $(\delta, \omega)$ is known, denoted by $p(\delta, \omega)$.
Shape Optimization of An Idealized Bypass Graft Model

- **Target anastomosis angles** given to the surgeon: \( x = (x_1, x_2) \).
Motivation

Shape Optimization of An Idealized Bypass Graft Model

- **Target anastomosis angles** given to the surgeon: \( x = (x_1, x_2) \).
- Actual angles constructed in a surgery: \( \theta = (\theta_1, \theta_2) = x + \delta \), where \( \delta = (\delta_1, \delta_2) \) are the *implementation errors* introduced during surgery.
Motivation

Shape Optimization of An Idealized Bypass Graft Model

- **Target anastomosis angles** given to the surgeon: \( x = (x_1, x_2) \).
- Actual angles constructed in a surgery: \( \theta = (\theta_1, \theta_2) = x + \delta \), where \( \delta = (\delta_1, \delta_2) \) are the *implementation errors* introduced during surgery.
- **Stenosis radius** \( r \) and **inflow velocity** \( v \) are environmental variables.
Shape Optimization of An Idealized Bypass Graft Model

- **Target anastomosis angles** given to the surgeon: \( x = (x_1, x_2) \).
- Actual angles constructed in a surgery: \( \theta = (\theta_1, \theta_2) = x + \delta \), where \( \delta = (\delta_1, \delta_2) \) are the **implementation errors** introduced during surgery.
- **Stenosis radius** \( r \) and **inflow velocity** \( v \) are environmental variables.
- The **area of low wall-shear stress (WSS)** given \( \theta \) and \( \omega = (r, v) \) is \( f(\theta, \omega) \). \( f \) can be evaluated **exactly** through **expensive** simulation.
Shape Optimization of An Idealized Bypass Graft Model

- **Target anastomosis angles** given to the surgeon: \( x = (x_1, x_2) \).
- Actual angles constructed in a surgery: \( \theta = (\theta_1, \theta_2) = x + \delta \), where \( \delta = (\delta_1, \delta_2) \) are the **implementation errors** introduced during surgery.
- **Stenosis radius** \( r \) and **inflow velocity** \( v \) are environmental variables.
- The **area of low wall-shear stress (WSS)** given \( \theta \) and \( \omega = (r, v) \) is \( f(\theta, \omega) \). \( f \) can be evaluated **exactly** through expensive simulation.
- **Utility function** with optional **risk aversion**: \( U = -f \) or \( U = e^{-\alpha \cdot f} \) \((\alpha > 0)\).
**Motivation**

**Shape Optimization of An Idealized Bypass Graft Model**

- **Target anastomosis angles** given to the surgeon: \( x = (x_1, x_2) \).
- Actual angles constructed in a surgery: \( \theta = (\theta_1, \theta_2) = x + \delta \), where \( \delta = (\delta_1, \delta_2) \) are the *implementation errors* introduced during surgery.
- **Stenosis radius** \( r \) and **inflow velocity** \( v \) are environmental variables.
- The *area of low wall-shear stress (WSS)* given \( \theta \) and \( \omega = (r, v) \) is \( f(\theta, \omega) \). \( f \) can be evaluated exactly through expensive simulation.
- **Utility function** with optional *risk aversion*: \( U = -f \) or \( U = e^{-\alpha \cdot f} \) \( (\alpha > 0) \).
- The *joint probability density* of \((\delta, \omega)\) is known, denoted by \( p(\delta, \omega) \).
Shape Optimization of An Idealized Bypass Graft Model

\[ \theta = (30, 45) \]
Shape Optimization of An Idealized Bypass Graft Model

\[ \theta = (30, 45) \]

\[ \theta = (70, 45) \]
Shape Optimization of An Idealized Bypass Graft Model

Sections from:

- bottom of the artery
- top of the graft

- bottom of the graft
Our overarching goal is to find the target anastomosis angles $x$ that maximize the expected value of $U(\cdot, \cdot)$,
Our overarching goal is to find the target anastomosis angles $x$ that maximize the expected value of $U(\cdot, \cdot)$, i.e., we want to solve

$$\max_x g(x),$$

where

$$g(x) := \int \int U(x + \delta, \omega) p(\delta, \omega) \, d\delta \, d\omega$$

is the expected utility that results from using target values $x$. 
Simulation Optimization under Input Uncertainties

In this problem:

- The random output variable (area of low WSS) is a deterministic function of a low-dimensional random vector (implementation error, inflow velocity and stenosis radius).
In this problem:

- The random output variable (area of low WSS) is a deterministic function of a low-dimensional random vector (implementation error, inflow velocity and stenosis radius).

- Evaluation of the deterministic function is achieved through simulations of blood-flow in the graft, which are expensive. Derivative information of the function is unavailable.
In this problem:

- The random output variable (area of low WSS) is a deterministic function of a low-dimensional random vector (implementation error, inflow velocity and stenosis radius).

- Evaluation of the deterministic function is achieved through simulations of blood-flow in the graft, which are expensive. Derivative information of the function is unavailable.

- We want to optimize the expectation of this output variable (or its variant) by allocating simulation effort efficiently across different values of the random vector.
This problem is studied in Sankaran & Marsden 2010, which

- uses the **stochastic collocation** technique (Sankaran & Marsden 2011) to incorporate and study the effects of input uncertainties;
This problem is studied in Sankaran & Marsden 2010, which

- uses the **stochastic collocation** technique (Sankaran & Marsden 2011) to incorporate and study the effects of input uncertainties;

- applies a derivative-free **surrogate management framework (SMF)** optimization method (Marsden et al. 2008) to perform robust shape design of cardiovascular simulations;
This problem is studied in Sankaran & Marsden 2010, which

- uses the stochastic collocation technique (Sankaran & Marsden 2011) to incorporate and study the effects of input uncertainties;

- applies a derivative-free surrogate management framework (SMF) optimization method (Marsden et al. 2008) to perform robust shape design of cardiovascular simulations;

- demonstrates that accounting for implementation and measurement uncertainties affects the optimal graft attachment angle.
In this work, we employ a **Bayesian** approach, where we use
In this work, we employ a **Bayesian** approach, where we use

- inference based on a **Gaussian process prior** (Rasmussen & Williams 2006) to learn the behavior of the expensive simulator across the input space, and track our uncertainty about values at unevaluated points;
In this work, we employ a **Bayesian** approach, where we use

- inference based on a **Gaussian process prior** (Rasmussen & Williams 2006) to learn the behavior of the expensive simulator across the input space, and track our uncertainty about values at unevaluated points;

- **Bayesian quadrature** or **Bayesian Monte Carlo** techniques (O’Hagan 1991, Rasmussen & Ghahramani 2003) to evaluate the integral (expectation), by taking advantage of the analytical convenience of the GP models;
In this work, we employ a **Bayesian** approach, where we use

- **inference based on a Gaussian process prior** (Rasmussen & Williams 2006) to learn the behavior of the expensive simulator across the input space, and track our uncertainty about values at unevaluated points;

- **Bayesian quadrature or Bayesian Monte Carlo** techniques (O’Hagan 1991, Rasmussen & Ghahramani 2003) to evaluate the integral (expectation), by taking advantage of the analytical convenience of the GP models;

- **value of information** calculations (Frazier et al. 2008, Chick & Gans 2009) to decide at which inputs it would be most valuable to run the simulator next.
We use Bayesian statistics to provide

- estimates of \( U(\theta, \omega) \) across all points \((\theta, \omega)\), as well as
- uncertainties associated with these estimates,

based on those points at which \( U \) has actually been evaluated.
Step 1: Evaluating the Utility Function $U$

We use Bayesian statistics to provide

- estimates of $U(\theta, \omega)$ across all points $(\theta, \omega)$, as well as
- uncertainties associated with these estimates,

based on those points at which $U$ has actually been evaluated.

We place a Gaussian process prior distribution over $U$: $U(\cdot, \cdot) \sim \text{GP}(\mu_0, \Sigma_0)$
Step 1: Evaluating the Utility Function $U$

We use Bayesian statistics to provide

- estimates of $U(\theta, \omega)$ across all points $(\theta, \omega)$, as well as
- uncertainties associated with these estimates,

based on those points at which $U$ has actually been evaluated.

We place a Gaussian process prior distribution over $U$: $U(\cdot, \cdot) \sim \text{GP}(\mu_0, \Sigma_0)$

- GP priors are frequently used in the Bayesian global optimization (Mockus 1989, Jones et al. 1998), to model our belief about an implicit continuous function over $\mathbb{R}^d$ that closer inputs are more likely to cause similar outputs.
Step 1: Evaluating the Utility Function $U$

We use Bayesian statistics to provide

- **estimates** of $U(\theta, \omega)$ across *all* points $(\theta, \omega)$, as well as
- **uncertainties** associated with these estimates,

based on those points at which $U$ has actually been evaluated.

We place a **Gaussian process** prior distribution over $U$: $U(\cdot, \cdot) \sim \text{GP}(\mu_0, \Sigma_0)$

- **Correlations** in a GP prior are extremely important for reducing the number of samples needed to evaluate an expensive function, since they allow us to learn about areas that have *not* been measured from those that have.
Step 1: Evaluating the Utility Function $U$

- Is a GP prior appropriate for our bypass graft application?

*leave-one-out cross-validation*
Step 1: Evaluating the Utility Function $U$

We use Bayesian statistics to provide

- estimates of $U(\theta, \omega)$ across all points $(\theta, \omega)$, as well as
- uncertainties associated with these estimates,

based on those points at which $U$ has actually been evaluated.

We place a Gaussian process (GP) prior distribution over $U$: $U(\cdot, \cdot) \sim \text{GP}(\mu_0, \Sigma_0)$

The posterior distribution on $U$ at time $n$ (after observing $n \geq 1$ samples) is then $\text{GP}(\mu_n, \Sigma_n)$, where $\mu_n$ and $\Sigma_n$ can be computed recursively.
A Bayesian Quadrature VOI-based Design

Step 1: Evaluating the Utility Function $U$

We use Bayesian statistics to provide

- estimates of $U(\theta, \omega)$ across all points $(\theta, \omega)$, as well as
- uncertainties associated with these estimates,

based on those points at which $U$ has actually been evaluated.

We place a Gaussian process (GP) prior distribution over $U$: $U(\cdot, \cdot) \sim \text{GP}(\mu_0, \Sigma_0)$

The posterior distribution on $U$ at time $n$ (after observing $n \geq 1$ samples) is then $\text{GP}(\mu_n, \Sigma_n)$, where $\mu_n$ and $\Sigma_n$ can be computed recursively.

These estimates and associated uncertainties then imply estimates and uncertainties of \( g(x) = \int \int U(x + \delta, \omega) \ p(\delta, \omega) \ \text{d}\delta \text{d}\omega \) across the domain of $x$. 
Step 2: Evaluating the Objective Function $g$

At time $n$, the **posterior mean** of the function $g$ at an arbitrary point $x$, and the **posterior covariance** between $g(x)$ and $g(x')$ at two arbitrary points $x$ and $x'$, are

\[
\mathbb{E}_n [g(x)] = \int \int \mu_n (x + \delta, \omega) p(\delta, \omega) d\delta d\omega,
\]

\[
\text{Cov}_n [g(x), g(x')] = \int \int \int \Sigma_n (x + \delta, \omega, x' + \delta', \omega') p(\delta, \omega) p(\delta', \omega') d\delta d\omega d\delta' d\omega',
\]

and can be computed analytically when e.g., $\Sigma_0$ is a square exponential covariance function, and $\delta, \omega$ are uniformly or (truncated) normally distributed.
Step 2: Evaluating the Objective Function $g$

At time $n$, the posterior mean of the function $g$ at an arbitrary point $x$, and the posterior covariance between $g(x)$ and $g(x')$ at two arbitrary points $x$ and $x'$, are

$$\mathbb{E}_n[g(x)] = \int \int \mu_n(x + \delta, \omega) p(\delta, \omega) \, d\delta \, d\omega,$$

$$\text{Cov}_n[g(x), g(x')] = \int \int \int \int \Sigma_n(x + \delta, \omega, x' + \delta', \omega') p(\delta, \omega) p(\delta', \omega') \, d\delta \, d\omega \, d\delta' \, d\omega',$$

and can be computed analytically when e.g., $\Sigma_0$ is a square exponential covariance function, and $\delta$, $\omega$ are uniformly or (truncated) normally distributed.
Step 2: Evaluating the Objective Function $g$

At time $n$, the posterior mean of the function $g$ at an arbitrary point $x$, and the posterior covariance between $g(x)$ and $g(x')$ at two arbitrary points $x$ and $x'$, are

$$
\mathbb{E}_n[g(x)] = \int \int \mu_n(x + \delta, \omega) p(\delta, \omega) \, d\delta \, d\omega,
$$

$$
\text{Cov}_n[g(x), g(x')] = \int \int \int \int \Sigma_n(x + \delta, \omega, x' + \delta', \omega') p(\delta, \omega) p(\delta', \omega') \, d\delta \, d\omega \, d\delta' \, d\omega',
$$

and can be computed analytically when e.g., $\Sigma_0$ is a square exponential covariance function, and $\delta, \omega$ are uniformly or (truncated) normally distributed.

Then, if we were to stop after $n$ evaluations of the simulator, we would choose

$$
x_n^* = \arg\max_x \mathbb{E}_n[g(x)],
$$

which is the Bayes-optimal solution.
Step 3: Optimizing the Objective Function $g$

At each time $n \geq 0$, we apply a *value of information* analysis to determine $(\theta_{n+1}, \omega_{n+1})$, which is the point to evaluate at time $n+1$. 
Step 3: Optimizing the Objective Function $g$

At each time $n \geq 0$, we apply a value of information analysis to determine $(\theta_{n+1}, \omega_{n+1})$, which is the point to evaluate at time $n+1$.

What is the value of evaluating $U$ at an arbitrary point $(\theta, \omega)$ at time $n+1$?
Step 3: Optimizing the Objective Function $g$

At each time $n \geq 0$, we apply a *value of information* analysis to determine $(\theta_{n+1}, \omega_{n+1})$, which is the point to evaluate at time $n+1$.

What is the value of evaluating $U$ at an arbitrary point $(\theta, \omega)$ at time $n+1$?

- The quality of the best solution that we have right now is $\max_x \mathbb{E}_n [g(x)]$. 
Step 3: Optimizing the Objective Function $g$

At each time $n \geq 0$, we apply a value of information analysis to determine $(\theta_{n+1}, \omega_{n+1})$, which is the point to evaluate at time $n + 1$.

What is the value of evaluating $U$ at an arbitrary point $(\theta, \omega)$ at time $n + 1$?

- The quality of the best solution that we have right now is $\max_x E_n [g(x)]$.
- The quality of the best solution we can obtain after we observe the sample $y_{n+1} = U(\theta, \omega)$ at time $n + 1$ is $Q = \max_x E_{n+1} [g(x)]$. 
Step 3: Optimizing the Objective Function $g$

At each time $n \geq 0$, we apply a value of information analysis to determine $(\theta_{n+1}, \omega_{n+1})$, which is the point to evaluate at time $n + 1$.

What is the value of evaluating $U$ at an arbitrary point $(\theta, \omega)$ at time $n + 1$?

- The quality of the best solution that we have right now is $\max_x \mathbb{E}_n [g(x)]$.
- The quality of the best solution we can obtain after we observe the sample $y_{n+1} = U(\theta, \omega)$ at time $n + 1$ is $Q = \max_x \mathbb{E}_{n+1} [g(x)]$.
- This quantity is unknown at time $n$, as it depends on $y_{n+1}$. Yet we can calculate its expected value at time $n$, i.e., $\mathbb{E}_n [Q | \theta_{n+1} = \theta, \omega_{n+1} = \omega]$. 
Step 3: Optimizing the Objective Function $g$

At each time $n \geq 0$, we apply a *value of information* analysis to determine $(\theta_{n+1}, \omega_{n+1})$, which is the point to evaluate at time $n+1$.

What is the value of evaluating $U$ at an arbitrary point $(\theta, \omega)$ at time $n+1$?

- The quality of the best solution that we have right now is $\max_x \mathbb{E}_n [g(x)]$.
- The quality of the best solution we can obtain *after* we observe the sample $y_{n+1} = U(\theta, \omega)$ at time $n + 1$ is $Q = \max_x \mathbb{E}_{n+1} [g(x)]$.
- This quantity is unknown at time $n$, as it depends on $y_{n+1}$. Yet we can calculate its expected value at time $n$, i.e., $\mathbb{E}_n [Q \mid \theta_{n+1} = \theta, \omega_{n+1} = \omega]$.
- The *expected improvement* in solution quality from time $n$ to time $n + 1$, is the value of the information achieved from measuring $(\theta, \omega)$ at time $n+1$:

$$V_n(\theta, \omega) = \mathbb{E}_n \left[ \max_x \mathbb{E}_{n+1} [g(x)] \mid \theta_{n+1} = \theta, \omega_{n+1} = \omega \right] - \max_x \mathbb{E}_n [g(x)].$$
Step 3: Optimizing the Objective Function $g$

The value of evaluating $U$ at point $(\theta, \omega)$ at time $n+1$ is

$$V_n(\theta, \omega) = \mathbb{E}_n \left[ \max_x \mathbb{E}_{n+1}[g(x)] \mid \theta_{n+1} = \theta, \omega_{n+1} = \omega \right] - \max_x \mathbb{E}_n [g(x)].$$
Step 3: Optimizing the Objective Function $g$

The value of evaluating $U$ at point $(\theta, \omega)$ at time $n+1$ is

$$V_n(\theta, \omega) = \mathbb{E}_n \left[ \max_x \mathbb{E}_{n+1}[g(x)] \mid \theta_{n+1} = \theta, \omega_{n+1} = \omega \right] - \max_x \mathbb{E}_n[g(x)].$$

We seek to evaluate the simulator at the point maximizing the value of information. That is, we want to evaluate at time $n+1$

$$(\theta_{n+1}, \omega_{n+1}) = \arg\max_{\theta, \omega} V_n(\theta, \omega).$$

(1)
Step 3: Optimizing the Objective Function $g$

The value of evaluating $U$ at point $(\theta, \omega)$ at time $n+1$ is

$$V_n(\theta, \omega) = \mathbb{E}_n \left[ \max_x \mathbb{E}_{n+1}[g(x)] \mid \theta_{n+1} = \theta, \omega_{n+1} = \omega \right] - \max_x \mathbb{E}_n[g(x)].$$

We seek to evaluate the simulator at the point maximizing the value of information. That is, we want to evaluate at time $n+1$

$$(\theta_{n+1}, \omega_{n+1}) = \arg\max_{\theta, \omega} V_n(\theta, \omega).$$

(1)

Applying more analysis, we can compute $V_n(\cdot, \cdot)$, $\nabla_{\theta} V_n(\theta, \omega)$ and $\nabla_{\omega} V_n(\theta, \omega)$ analytically. We can then solve (1) using multi-start gradient ascent.
A Simple Test Problem

- $x$, $\delta$ and $\omega$ are 1-d
- $\delta \equiv 0$ ($\theta \equiv x$) and $\omega \sim \mathcal{N}(1, 1/9)$
- $U(\theta, \omega) = -100(\omega - \theta^2)^2 - (1 - \theta)^2$
A Simple Test Problem

- $x$, $\delta$ and $\omega$ are 1-d
- $\delta \equiv 0$ ($\theta \equiv x$) and $\omega \sim \mathcal{N}(1, 1/9)$
- $U(\theta, \omega) = -100(\omega - \theta^2)^2 - (1 - \theta)^2$

We measure performance by expected opportunity cost $\mathbb{E}[\max_x g(x) - g(x_n^*)]$ at each time $n$, where

- $g(x) = (1 - x)^2 + 100 \left[(1 - x^2)^2 + \frac{1}{9}\right]$. 
A Simple Test Problem

- $x, \delta$ and $\omega$ are 1-d
- $\delta \equiv 0$ ($\theta \equiv x$) and $\omega \sim \mathcal{N}(1, 1/9)$
- $U(\theta, \omega) = -100(\omega - \theta^2)^2 - (1 - \theta)^2$

We measure performance by expected opportunity cost $\mathbb{E}[\max_x g(x) - g(x^*_n)]$ at each time $n$, where

- $g(x) = (1 - x)^2 + 100 \left[(1 - x^2)^2 + \frac{1}{9}\right]$. 

![Graph showing comparison between random sampling and VOI-based sampling](image)
A Bayesian Quadrature VOI-based Design

Illustration of Our Sampling Algorithm
Illustration of Our Sampling Algorithm

posterior on $U$ at time 1

$\mu_1(\theta, \omega)$

posterior on $g$ at time 1

$a_1(x)$
Illustration of Our Sampling Algorithm

posterior on $U$ at time 2

posterior on $g$ at time 2
Illustration of Our Sampling Algorithm

posterior on $U$ at time 3

posterior on $g$ at time 3

$\mu_3(\theta, \omega)$

$a_3(x)$
Illustration of Our Sampling Algorithm

-posterior on $U$ at time 4

-posterior on $g$ at time 4
Illustration of Our Sampling Algorithm

posterior on $U$ at time 5

posterior on $g$ at time 5

$$\mu_5(\theta, \omega)$$

$$a_5(x)$$
Illustration of Our Sampling Algorithm

posterior on $U$ at time 6

posterior on $g$ at time 6
Illustration of Our Sampling Algorithm

posterior on $U$ at time 7

posterior on $g$ at time 7
A Bayesian Quadrature VOI-based Design

Illustration of Our Sampling Algorithm

posterior on $U$ at time 8

posterior on $g$ at time 8
Illustration of Our Sampling Algorithm

posterior on $U$ at time 9

posterior on $g$ at time 9
Illustration of Our Sampling Algorithm

posterior on $U$ at time 10

posterior on $g$ at time 10
posterior on $U$ at time 11

posterior on $g$ at time 11
Illustration of Our Sampling Algorithm

posterior on \( U \) at time 12

\[ \mu_{12}(\theta, \omega) \]

posterior on \( g \) at time 12

\[ a_{12}(x) \]
Illustration of Our Sampling Algorithm

posterior on $U$ at time 13

posterior on $g$ at time 13
Illustration of Our Sampling Algorithm

posterior on $U$ at time 14

posterior on $g$ at time 14
Illustration of Our Sampling Algorithm

posterior on $U$ at time 15

$\mu_{15}(\theta,\omega)$

posterior on $g$ at time 15

$a_{15}(x)$
posterior on $U$ at time 16

posterior on $g$ at time 16
A Bayesian Quadrature VOI-based Design

Illustration of Our Sampling Algorithm

posterior on $U$ at time 17

posterior on $g$ at time 17
Illustration of Our Sampling Algorithm

posterior on $U$ at time 18

posterior on $g$ at time 18
Illustration of Our Sampling Algorithm

posterior on $U$ at time 19

posterior on $g$ at time 19
Illustration of Our Sampling Algorithm

posterior on $U$ at time 20

posterior on $g$ at time 20
Illustration of Our Sampling Algorithm

posterior on $U$ at time 21

posterior on $g$ at time 21
Illustration of Our Sampling Algorithm

posterior on $U$ at time 22

posterior on $g$ at time 22
Illustration of Our Sampling Algorithm

posterior on $U$ at time 23

posterior on $g$ at time 23
Illustration of Our Sampling Algorithm

posterior on $U$ at time 24

posterior on $g$ at time 24
A Bayesian Quadrature VOI-based Design

Illustration of Our Sampling Algorithm

posterior on $U$ at time 25

posterior on $g$ at time 25
Illustration of Our Sampling Algorithm

posterior on $U$ at time 26

posterior on $g$ at time 26
Illustration of Our Sampling Algorithm

posterior on $U$ at time 27

posterior on $g$ at time 27
Illustration of Our Sampling Algorithm

posterior on $U$ at time 28

posterior on $g$ at time 28
Illustration of Our Sampling Algorithm

posterior on $U$ at time 29

\[ \mu_{29}(\theta, \omega) \]

posterior on $g$ at time 29

\[ a_{29}(x) \]
Illustration of Our Sampling Algorithm

posterior on $U$ at time 30

posterior on $g$ at time 30
A Bayesian Quadrature VOI-based Design

Illustration of Our Sampling Algorithm
A Harder Test Problem

- $\theta_1 \sim \mathcal{N}(x_1, 1/9)$, $\theta_2 \sim \mathcal{N}(x_2, 1/36)$, $\nu \sim \mathcal{N}(0, 1/9)$, $r \sim \mathcal{N}(2, 4/9)$
- $U(\theta_1, \theta_2, \nu, r) = [\theta_1^2 + (\theta_1 - \nu)^2] \cdot [\theta_2^2 + (\theta_2 - r)^2]$

Bayesian quadrature designs

SMF with stochastic collocation
Conclusions & Future Work

In many applications of simulation-based optimization, the random output variable whose *expectation* is being optimized is

- a deterministic function of a low-dimensional random vector.
- expensive to compute, making simulation optimization difficult.
In many applications of simulation-based optimization, the random output variable whose expectation is being optimized is:
- a deterministic function of a low-dimensional random vector.
- expensive to compute, making simulation optimization difficult.

We design an algorithm that exploits this random vector’s low-dimensionality to improve performance, using:
- Gaussian processes (kriging),
- Bayesian quadrature techniques,
- value of information computations.
Conclusions & Future Work

In many applications of simulation-based optimization, the random output variable whose expectation is being optimized is

- a deterministic function of a low-dimensional random vector.
- expensive to compute, making simulation optimization difficult.

We design an algorithm that exploits this random vector’s low-dimensionality to improve performance, using

- Gaussian processes (kriging),
- Bayesian quadrature techniques,
- value of information computations.

<Future work> implement the algorithm in bypass grafts shape design.
THANK YOU!