Tutorial on

The Design of Computer Experiments for Optimization

Thomas Santner
Department of Statistics
The Ohio State University

The A-C-NW Optimization Tutorials
June 13, 2005

Outline

1. Three Types of Experiments

1. **Physical Experiments**
   - **Gold** standard for establishing cause and effect relationships
   - Mainstay of Engineering, Agriculture, Medicine
   - Principles of randomization, blocking, choice of sample size, and stochastic modeling of response variables all developed in response to needs of physical experiments

2. **Simulation Experiments** Complex physical system each of whose parts behave stochastically and interact in a known manner but whose ensemble stochastic behavior is not understood analytically
   - Used extensively in IE/OR--compare hospital emergency room setups

3. **Computer Experiments** past 15 years

2. What are Computer Experiments?

Idea: Many physical processes can not be studied by conventional experiments
Why?
(1) technically, the physical process is too difficult (expensive) to study experimentally
(2) ethical considerations
(3) number of input variables is too large

If either
(1) the components of the process of interest and their interactions are adequately understood so it can simulated (with negligable MC error) **or**
(2) the physics of the process of interest is

- sufficiently well understood so it can be described by a **mathematical model** relating the **response** to the **potential factors** that affect the output inputs,
- Numerical methods exist for solving the mathematical model
- The numerical methods can be implemented with computer code

Then the computer code can serve as a **proxy** for the physical process. As in a physical experiment,

$$\boldsymbol{x} \in \boldsymbol{\mathcal{X}} \subset \mathbb{R}^d \longrightarrow \boxed{\textbf{Code}} \longrightarrow y(\boldsymbol{x})$$

$$\boldsymbol{x} \in \boldsymbol{\mathcal{X}} \subset \mathbb{R}^d \longrightarrow \boxed{\textbf{Code}} \longrightarrow y(\boldsymbol{x})$$

Features of Computer Experiments

- $y(\boldsymbol{x})$ is deterministic
- Our interest is in settings where very few computer runs are possible due to
    1. Complex codes
    2. High--dimensional input $\boldsymbol{x}$
- Traditional principles used in designing physical experimentals (eg randomization, blocking, . . .) are irrelevant.

3. Examples

(1) Design of VLSI circuits

(2) Modeling weather or climate

(3) Design of automobile (components)

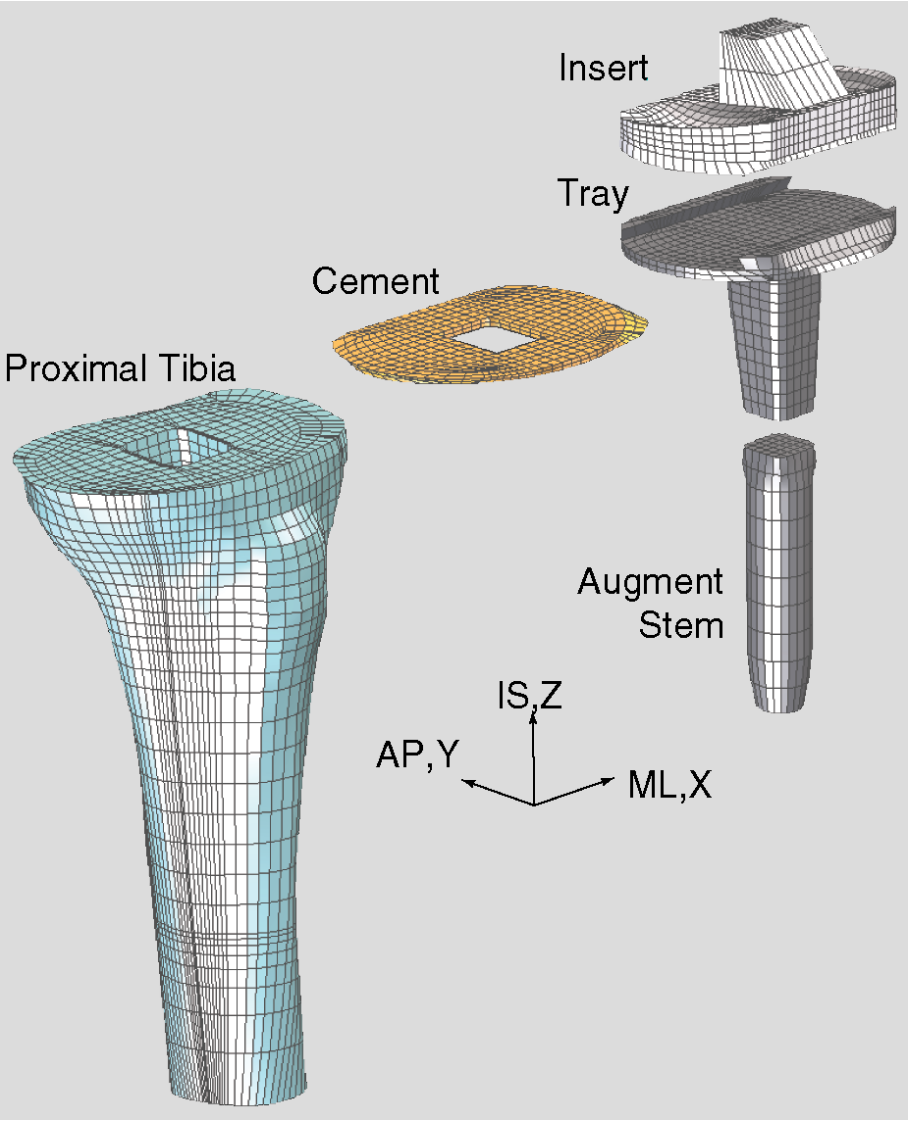
(4) Determine the performance of controlled nuclear fusion devices

(5) Temporal evolution of contained and wild fires

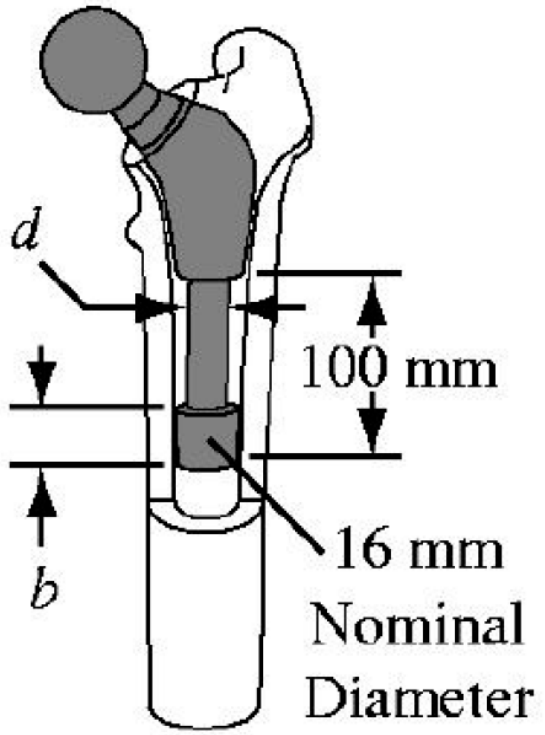
(6) Design of helicopter rotor blades

(7) Biomechanics — Design of prosthetic devices

Example (7) Designing hip and knee implants



Biomechanics I Design a hip implant



Goal To determine the design of a hip implant, i.e, (b, d) that minimizes femoral stress shielding while providing adequate resistance to implant toggling.

Inputs

1. **Engineering Variables** (manufacturing design)

   - Prosthesis geometry (length, cross-section, width, etc)
   - Prosthesis material
   - Nominal insertion parameters

2. **Environmental Variables (Patient** & **Surgical Variables)**

   - Bone material properties, weight (and other patient variables)
   - Deviation from nominal insertion parameters
     (and other surgical variables)

In the above figure, the **Engineering Variables** are

- b = bullet-tip length
- d = midstem diameter

the **Environmental Variables** are

- E = trabecular bone elastic modulus
- θ = joint force angle
- f = Implant-bone interface friction

Computed Response constructed from:

- $S = S(b, d, E, \theta, f) =$ normalized measure of bone stress shielding
- $D = D(b, d, E, \theta, f) =$ normalized measure of implant toggling

(competing objectives!!)

Formulation #1 Combine S & D because they represent competing objectives. Goal is to **minimize**

$$y(b, d, E, \theta, f) = wS(b, d, E, \theta, f) + (1 - w)D(b, d, E, \theta, f)$$

where w measures the relative importance of the two objectives.

Formulation #2 Goal is to **minimize**

$$S(b, d, E, \theta, f)$$

subject to

$$D(b, d, E, \theta, f) \leq B$$

where B is a given bound (a constrained optimization problem)

Biomechanics II A Computer Model of a Knee Simulator

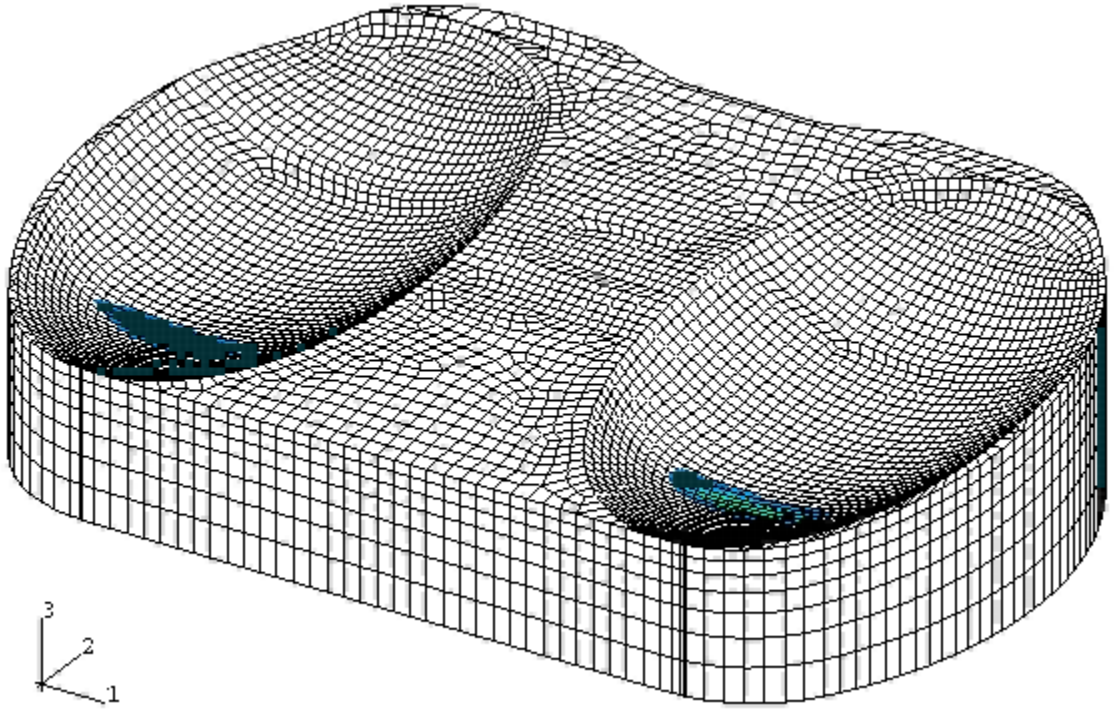
Three data sources

1. Knee Simulator (a machine)
2. Computer code that emulates the Knee Simulator

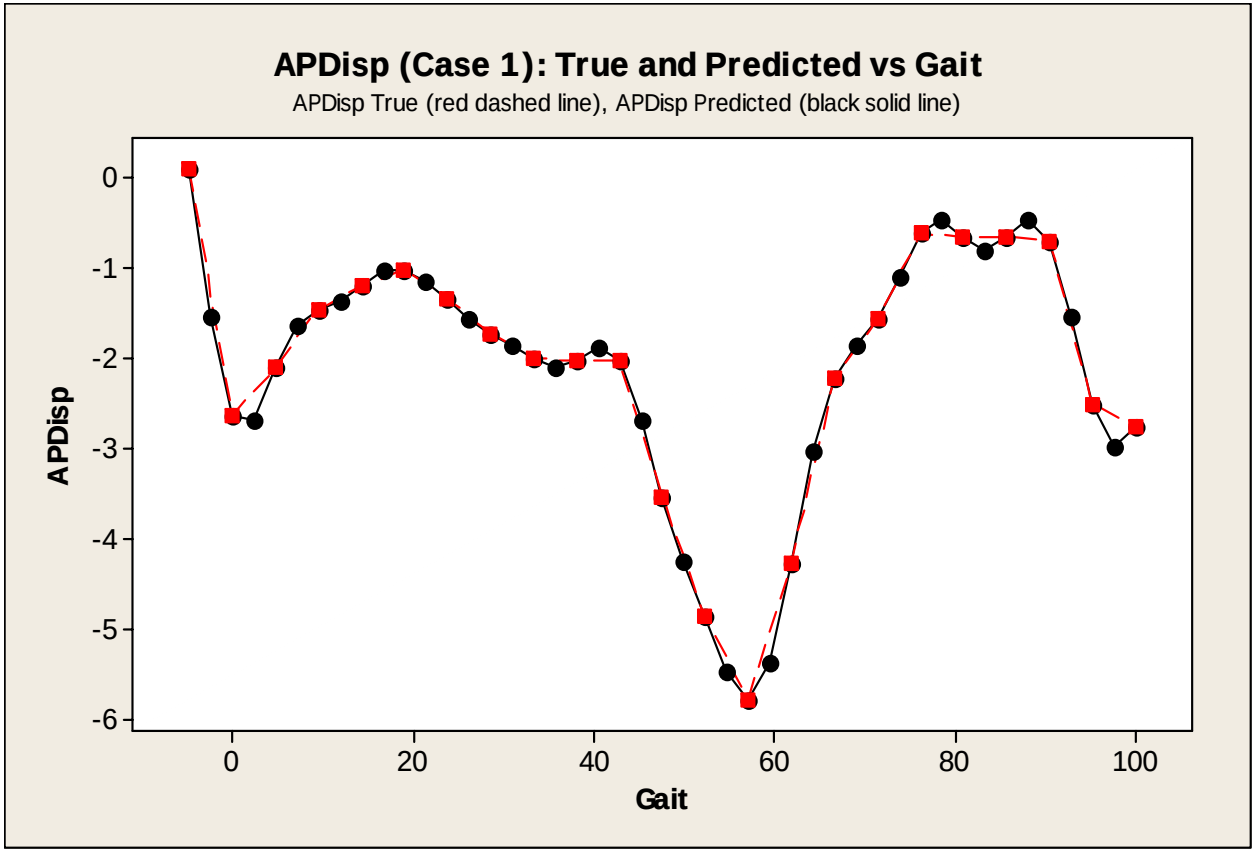
Inputs (7-10)

- Loading pattern (Flexion angle, Axial Force, AP Force, IE Torque)
- Knee design (stem lengths, constrained or not, etc)
- Frequency with which the loading pattern is applied (running/walking)
- Elastic modulus of the polyethylene in the tibial tray
- Polyethylene Irradiated or not
- Friction between knee and femoral component
- Surface type (Elemental vs Analytic – in finite element code)
- Mesh Density (in finite element code)

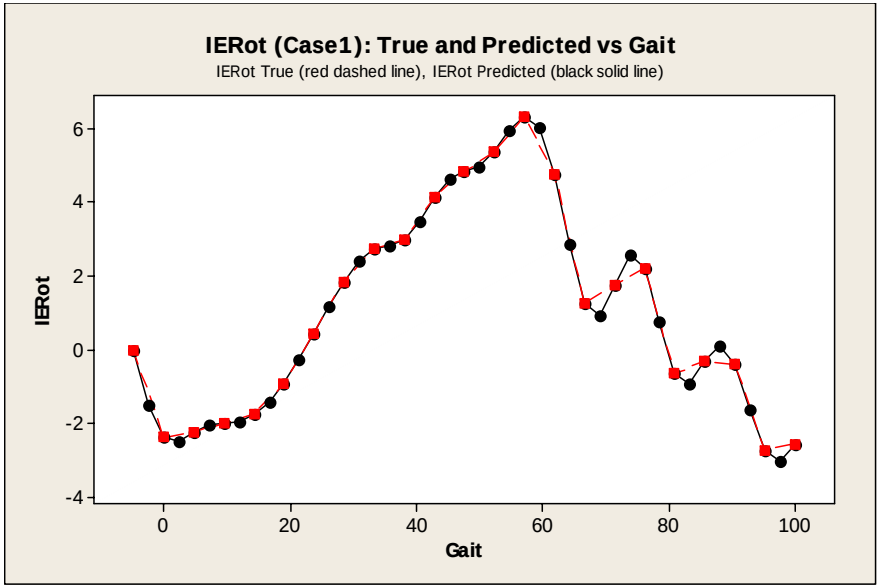
Mathematical Model (finite element model 12 hours/per run)



Output data from Knee simulator & computer code



and



Project Goals

1. "Calibrate" computer code to mimic knee simulator
2. Use calibrated computer code to produce effects seen in retreived knees
3. Explain the biomechanics of prosthetic joint failure

4. Nomenclature and a Taxonomy of Problems for Computer Exeriments

Inputs $\boldsymbol{x} = (\boldsymbol{x}_c, \boldsymbol{x}_e)$ where

$\boldsymbol{x}_c$ = **Engineering Design variables** (each choice of $\boldsymbol{x}_c$ is Engineering Design)

$\boldsymbol{x}_e$ = **Environmental Variables** (field, noise) , eg, patient bone densities.

Philosophy We often regard **environmental variables** as random variables with a distribution that represents target field conditions, i.e., $\boldsymbol{X}_e \sim \mathrm{F}(\bullet)$

Outputs

Real-valued : $y(\bullet)$

or

Multivariate: $(y_1(\bullet),\ y_2(\bullet), \ldots, y_k(\bullet))$

or

Functional: $(t, y(t, \bullet))$

- Multivariate data: single or multiple codes, e.g., code computes $y(\bullet)$ and all first partial derivatives of $y(\bullet)$)
- Functional Data: APD or IER gait profile

Special Features of (Biomechanics) Problems

• Codes for $y(\boldsymbol{x})$ are often **long-running**

• Sometimes associated **Physical Experiments** are available with output, $Y^{\mathrm{P}}(\bullet)$. Usual philosophy is that $Y^{p}(\boldsymbol{x})$ is a **noisy** measurements of the true input-output relationship, which we denote $\mu^{T}(\boldsymbol{x})$. In detail,

$$Y^{p}(\boldsymbol{x}) = \mu^{T}(\boldsymbol{x}) + \epsilon(\boldsymbol{x})$$

where the $\{\epsilon(\boldsymbol{x})\}_{\boldsymbol{x}}$ are independent measurement errors having mean zero and unknown variance σ_{ϵ}^{2} and we regard $\mu^{T}(\boldsymbol{x})$ as the **true, unknown** i-o relationship.

Caveats Sometimes only physical experiments are available for **components** of the ensemble process -- nuclear reactor simulator, code that emulates auto crash test. In other cases, only experiments that **approximate** reality are available--knee simulator

• When there are field variables, $y(\boldsymbol{x}_{\mathrm{c}}, \boldsymbol{X}_{e})$ has a distribution. We might typically be interested in one of several summary quantities associated with the distribution of $y(\boldsymbol{x}_{c}, \boldsymbol{X}_{e})$. For example,

- $\mu(\boldsymbol{x}_{c}) = E\{y(\boldsymbol{x}_{c}, \boldsymbol{X}_{e})\}$ (that the quantity around which the computer output varies)
- $\sigma^{2}(\boldsymbol{x}_{c})$ = Var$(y(\boldsymbol{x}_{c}, \boldsymbol{X}_{e}))$ (one measure of the very ability of the computer output due to variation in field inputs)

Taxonomy of Problems

• **Interpolation/Emulation** – Given computer model output at a set of inputs (training data), predict the computer simulation output at a new, untried input settings

• **Experimental design** – Determine input settings in which to carry out the sequence of simulation designs (a "good" design of a physical or computer experiment depends on the **scientific objective** of the research)

- Exploratory Designs ("space-filling")
- Prediction-based Designs
- **Optimization-based Designs (e.g., find $x^* = \text{argmin}\, y(x)$)**

• **Uncertainty/Output Analysis** – Determine the distribution of the computer model output when (some or all of) the inputs are random, i.e., determine the distribution of $y(\boldsymbol{x_d}, X_e)$. Examples of randomly varying inputs are patient specific variables (patient weight or patient bone material properties) or surgeon specific variables (measuring surgical skill)

Example In his Cornell PhD thesis, Kevin Ong studied the effect of **Surgical, Patient, and Fluid Effects** on the Stability of Uncemented Acetabular Components

• **Sensitivity Analysis** – Determine how variation in $y(\boldsymbol{x})$ can be apportioned to the different computer model inputs $\boldsymbol{x}$
(**Philosophy** Inputs that have relatively little effect on the output can be set to some nominal value and additional investigation restricted to determining how the output depends on the active inputs)

• **Calibration** – use physical experimental data and computer simulation runs to best estimate the computer code calibration variable (or to update the uncertainty regarding these parameters)

Example Set Mesh Density $= ?$, Load Discretization $= ?$, etc

• **Prediction** – Using the calibrated simulator to give predictions (with uncertainty bounds) for an associated physical system.

• **Find Robust Inputs** – In experiments with engineering design and patient-specific, enviromental variables, determine robust choices of the engineering design variables. If

$$\mu(\boldsymbol{x}_{\mathrm{c}}) = E_F\{y(\boldsymbol{x}_{\mathrm{c}}, \boldsymbol{X}_e)\}$$

then a robust set of inputs $\boldsymbol{x}_{\mathrm{c}}$ is an engineering "design" whose output is **minimally sensitive** to uncertainty in the distribution $F(\bullet)$ of $\boldsymbol{X}_e$

Many of the problems above have "natural" solutions obtained by approximating the computer model by an even "faster" predictor, a **metamodel**. Statistical issues choosing the best possible surrogate for the code and devising valid methods to accomplish calibration etc.

5. Sequential Design of Computer Experiments for Global Optimization

Recall $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^d \longrightarrow$ **Code** $\longrightarrow y(\boldsymbol{x})$

• Minimizing the number of function evaluations (computer runs) is critical in many computer experiments (Many methods, eg, direct search algorithms such as Nelder Mead "simplex" algorithm, or gradient-based algorithms can require "too many" function evaluations to be useful in the computer experiment settings)

• Some specific optimization problems

Case 1 find $\boldsymbol{x}_{opt} \equiv \text{argmin}\, y(\boldsymbol{x})$

Case 2 Suppose $\boldsymbol{x} = (\boldsymbol{x_c}, \boldsymbol{x_e})$ where

$\boldsymbol{x_c}$ = **control (manufacturing, engineering design)** variables

$\boldsymbol{x_e}$ = **noise (field, enviromental)** variables

and $\mathbf{X}_e \sim F(\bullet)$ ("target field conditions"). $y(\boldsymbol{x_c}, \boldsymbol{X_e})$ is a random variable with distribution induced by $\boldsymbol{X_e}$. Find

$$\boldsymbol{x}_{c,\,opt} = \text{argmin}\, \mu_F(\boldsymbol{x_c})$$

where $\mu_F(\boldsymbol{x_c}) = E_F\{y(\boldsymbol{x_c}, \boldsymbol{X_e})\}$

Case 3 Suppose output is $(y_0(\bullet), y_1(\bullet), y_2(\bullet), \ldots, y_k(\bullet))$. Find

$$x^* \equiv \text{argmin}\, y_0(\boldsymbol{x})$$

subject to

$$y_i(\bullet) \leq B_i \; (1 \leq i \leq k)$$

Goal **Describe Efficient Global Optimization (EGO) Algorithms**

Idea EGO is a *direct search algorithm* that uses a $y(\boldsymbol{x})$ predictor, $\widehat{y}(\boldsymbol{x})$, to explore $y(\boldsymbol{x})$ surface and also *accounts for uncertainty* in the predictor

Part 1 The $y(x)$ Predictor!

Problem Given training data

$$\left(\boldsymbol{x}_1^{tr},\, y(\boldsymbol{x}_n^{tr})\right),\, \ldots,\, \left(\boldsymbol{x}_n^{tr},\, y(\boldsymbol{x}_1^{tr})\right)$$

predict $y(\boldsymbol{x}_0),$ where $\boldsymbol{x}_0$ is an untried new input

Idea Regard the function $y(\boldsymbol{x})$ as a realization, a "draw," from a random function $Y(\boldsymbol{x})$.

Philosophically different than regression, MARS, and other prediction methods that assume (complicated mean structure + simple correlation structure) and versus the methodology (below) which is based the assumptions of (simple mean + complicated correlation structure).

Issues

• What model for $Y(\boldsymbol{x})$? One that permits great flexibility in the form of $y(\boldsymbol{x})$!

• The simplest possible model for $Y(\boldsymbol{x})$ is

$$Y(\boldsymbol{x}) = \underbrace{\sum_j \beta_j f_j(\boldsymbol{x})}_{\text{"large scale trends"}} + \underbrace{Z(\boldsymbol{x})}_{\text{"smooth deviations"}}$$

$$= \beta^\top f(\boldsymbol{x}) + Z(\boldsymbol{x})$$

where

$f_1(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})$ are known regression functions,

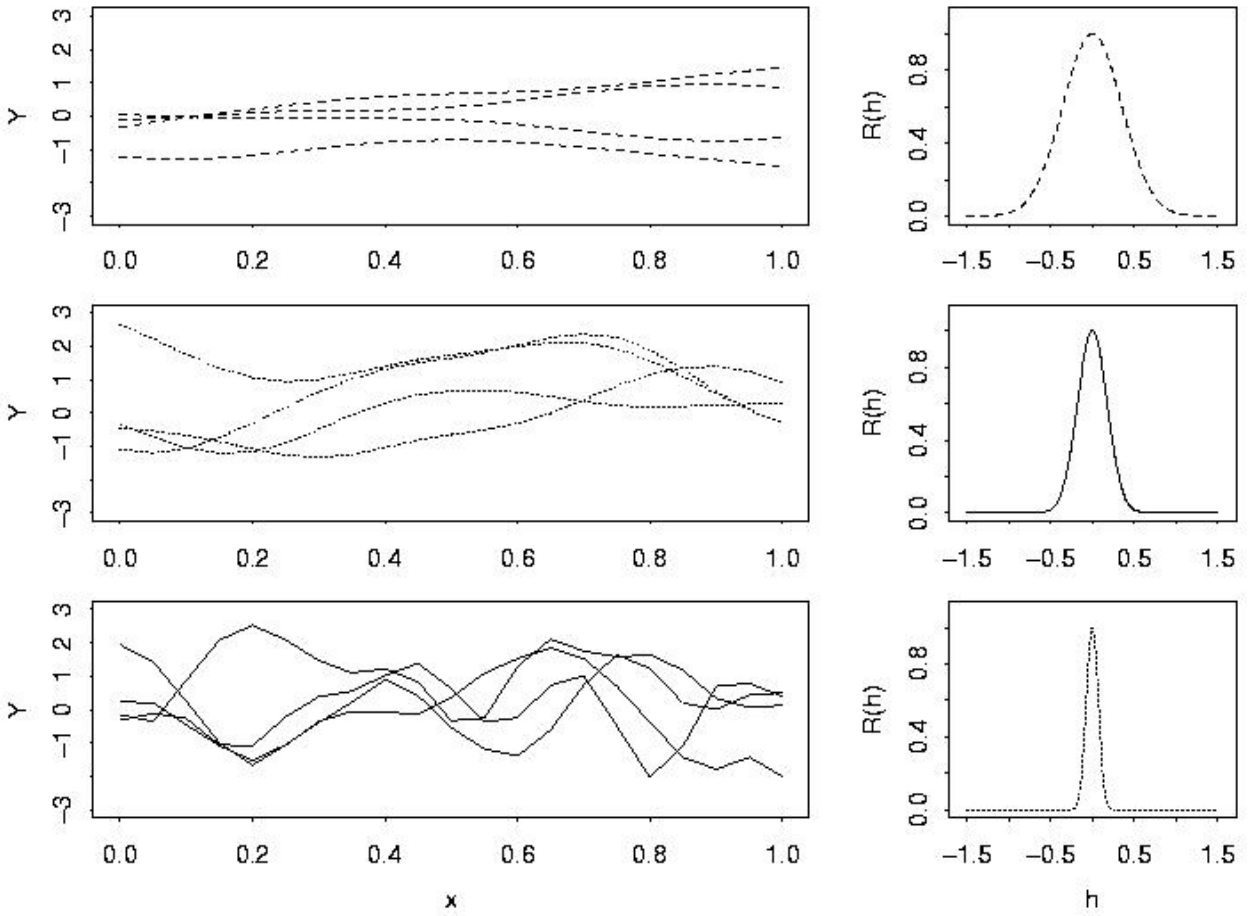
$\boldsymbol{\beta}$ is an unknown regression vector, and

$Z(\boldsymbol{x})$ is a "stationary Gaussian Stochastic Process" (GSP), a random function

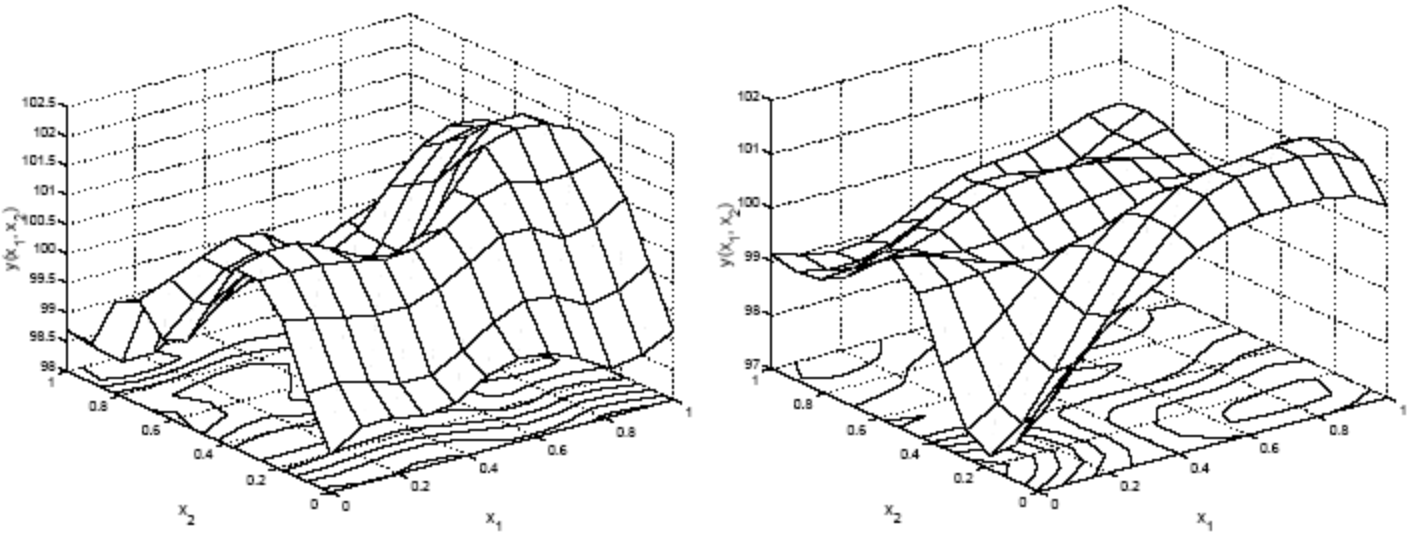
Example Four draws from a zero mean, unit variance GSP $Z(x)$ from [0,1] ➞ $\mathbb{R}$ with the "Gaussian" correlation function

$$R(h) = exp\big(-\theta h^2\big)$$

for $\theta =$ 0.5 (solid lines), $\theta =$ 1.0 (dotted lines), and $\theta =$ 10.0 (dashed lines)



- Some draws from a GSP function $Z{:}[0,1]^2 \longrightarrow \mathbb{R}$



- $Z(\boldsymbol{x}), \boldsymbol{x} \in \mathcal{X}$ has
  - $E\{Z(\boldsymbol{x})\} = 0$ (zero mean)
    $(\Rightarrow E\{Y(\boldsymbol{x})\} = \beta_0 + 0 = \beta_0)$
  - $\text{Var}(Z(\boldsymbol{x})) = \sigma_Z^2$
  - Correlation Function: $R(\cdot)$ with $R(\mathbf{0}) = 1$ and

$$\text{Cov}(Z(\boldsymbol{x}_1), Z(\boldsymbol{x}_2)) = \sigma_Z^2 R(\boldsymbol{x}_1 - \boldsymbol{x}_2)$$

  - Typically $R(\cdot) = R(\cdot|\boldsymbol{\xi})$, i.e., $R(\cdot)$ is known up to a finite vector of parameters
  - GSP: for any $\boldsymbol{x}_1, ..., \boldsymbol{x}_s$, $(Z(\boldsymbol{x}_1), ..., Z(\boldsymbol{x}_s))$ has the multivariate normal distribution

• Smoothness properties of $y(\boldsymbol{x})$ depend on smoothness of $R(\bullet)$ at the origin, $e.g$,

$$R(\boldsymbol{h}|\boldsymbol{\xi}) = \prod_{i=1}^{k} \exp\{-\xi_i |h_i|^{p_i}\}$$

• Best linear unbiased predictor (BLUP $\boldsymbol{\xi}$ known) or empirical BLUP (EBLUP-$\boldsymbol{\xi}$ estimated) can be calculated to predict $y(\bullet)$ at any $\boldsymbol{x}$

$$\boldsymbol{\widehat{y}(x_0) \equiv E\{Y(x_0)|\text{data}\}}$$

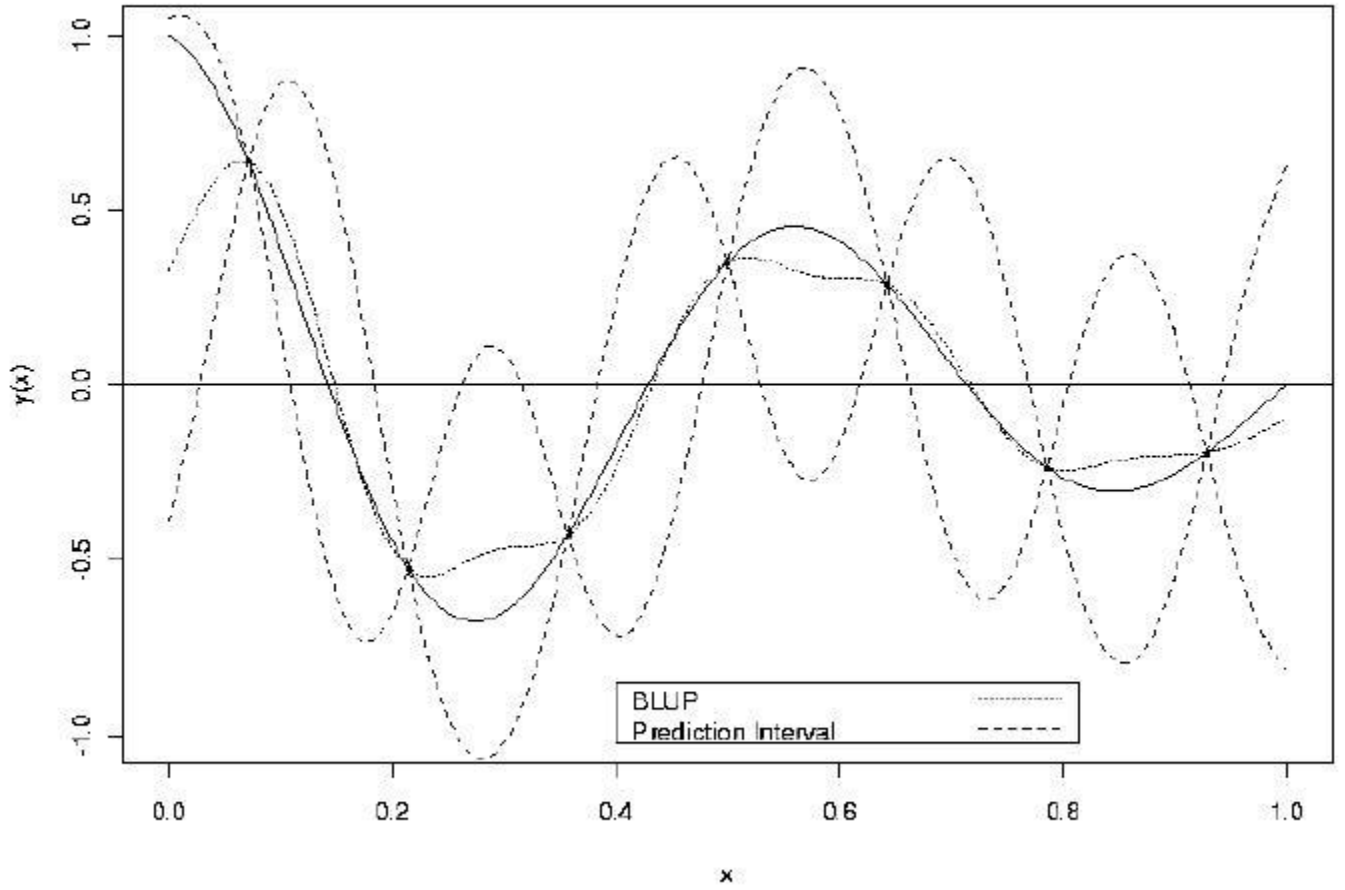
• Engineering literature often calls such a $\widehat{y}(\boldsymbol{x}) \equiv$ **metamodel**

• In addition,

$$\boldsymbol{\sigma^2(x_0) = E\Big((Y(x_0) - \widehat{y}(x_0))^2|\text{data}\Big) = \text{Var}(\widehat{y}(x_0)|\text{data})}$$

is a measure of our uncertainty about the predicted value of $y(\boldsymbol{x})$.

Example $n = 7$, 1-dim



The BLUP and corresponding pointwise 95% prediction interval limits for $y(\boldsymbol{x})$ for $n = 7$ training data observations

Properties of $\widehat{y}(\boldsymbol{x}_0)$

• Simple to compute (linear in training data $(y(\boldsymbol{x}_1^{tr}), \ldots, y(\boldsymbol{x}_n^{tr}))$, i.e.,

$$\widehat{y}(\boldsymbol{x}_0) = c_0 + \sum_{j=1}^{n} c_j \, y(\boldsymbol{x}_j^{tr})$$

• Viewed as a function of $\boldsymbol{x}_0$

$$\widehat{y}(\boldsymbol{x}_0) = d_0 + \sum_{i=1}^{n} d_j \, R(\boldsymbol{x}_0 - \boldsymbol{x}_i^{tr})$$

• $\widehat{y}(\boldsymbol{x})$ interpolates the training data, i.e.,

$$\widehat{y}(\boldsymbol{x}_i^{tr}) = y(\boldsymbol{x}_i^{tr}) \text{ for } i = 1, \ldots, n$$

• Splines, neural networks and other well-known interpolators correspond to specific choices of correlation function $R(\bullet)$

• **Software**

1. SAS Proc Mixed
2. PErK (B. J. Williams)

Part 2 Designing Computer Experiments to Find Global Optima

Expected Improvement-I

Goal Find $\boldsymbol{x}_{opt} = \text{argmin}\, y(\boldsymbol{x})$

(**Implicit** keep number of $y(\boldsymbol{x})$ evaluations "small")

Idea of Sequential Design Algorithm for the Computer Experiment

Given training data

$$\left(\boldsymbol{x}_1^{tr},\, y(\boldsymbol{x}_n^{tr})\right),\, \ldots,\, \left(\boldsymbol{x}_n^{tr},\, y(\boldsymbol{x}_1^{tr})\right)$$

choose $\boldsymbol{x}_{n+1}^{tr}$ to be that $\boldsymbol{x}$ which maximizes an **expected improvement** criterion as the "best" input $\boldsymbol{x}_{opt}$ at which to compute $y(\bullet)$

Improvement Let

$$y_n^{\min} = \min_{1 \le i \le n} y(\boldsymbol{x}_i^{tr})$$

$\equiv$ best (smallest) $y(\bullet)$ calculated through the n^{th} evaluation

Let

$$y_n^{\min} = \min_{1 \le i \le n} y(\boldsymbol{x}_i^{tr})$$

$\equiv$ best (smallest) $y(\bullet)$ calculated through the n^{th} evaluation. Consider **potential** new site $\boldsymbol{x}$. Then **define**

$$I(\boldsymbol{x}) = \begin{cases} 0, & y(\boldsymbol{x}) \ge y_n^{\min} \\ y_n^{\min} - y(\boldsymbol{x}), & y(\boldsymbol{x}) < y_n^{\min} \end{cases}$$

$$= \max\{0, y_n^{\min} - y(\boldsymbol{x})\}$$

is the improvement in using $\boldsymbol{x}$ as the $y(\boldsymbol{x})$ minimizer over current training data.

Warning In $I(\boldsymbol{x})$, $y_n^{\min}$ is **known** BUT $y(\boldsymbol{x})$ and (hence) $I(\boldsymbol{x})$ are **unknown** :--(.

Idea of the Algorithm

1. Obtain a starting design, ie., set of initial inputs, $\boldsymbol{x}_i$, at which to calculate y(•) (eg Space-filling "Latin Hypercube Design")

2. Evaluate $y(\boldsymbol{x})$ at the starting design and use the data to estimate any unknown correlation parameters $\boldsymbol{\xi}$

3. Choose $\boldsymbol{x}_{n+1}^{tr}$ to maximize the expected improvement **given** the current data (and $\boldsymbol{\xi}$), i.e.,

$$\boldsymbol{x}_{n+1}^{tr} \equiv \text{argmax}\, E\left\{ I(\boldsymbol{x}) | Y^n \equiv \left\{ \left(\boldsymbol{x}_i^{tr},\, y(\boldsymbol{x}_i^{tr}) \right) \right\}_{i=1}^{n} \right\} \tag{1}$$

(the expected value of $I(\boldsymbol{x})$ under the stochastic process model)

4. **Stop** when the maximum expected improvement is "small," and set

$$\hat{\boldsymbol{x}}_{opt} = \text{argmin}\, \hat{y}(\boldsymbol{x}),$$

where $\hat{y}(\boldsymbol{x})$ is the EBLUP of $y(\boldsymbol{x})$

Fact

$$\boldsymbol{x}_{n+1}^{tr} \equiv \underset{\mathcal{X}}{\text{argmax}} \left\{ (y_n^{\min} - \widehat{y}(\boldsymbol{x}))\Phi\left(\frac{y_n^{\min} - \widehat{y}(\boldsymbol{x})}{\sigma(\boldsymbol{x})} \right) + \sigma(\boldsymbol{x})\, \phi\left(\frac{y_n^{\min} - \widehat{y}(\boldsymbol{x})}{\sigma(\boldsymbol{x})} \right) \right\}$$

$$\boldsymbol{x}_{n+1}^{tr} \equiv \underset{\mathcal{X}}{\text{argmax}} \left\{ \underbrace{(y_n^{\min} - \widehat{y}(\boldsymbol{x}))\Phi\left(\frac{y_n^{\min} - \widehat{y}(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\right)}_{\textbf{(1)}} + \underbrace{\sigma(\boldsymbol{x})\,\phi\left(\frac{y_n^{\min} - \widehat{y}(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\right)}_{\textbf{(2)}} \right\}$$

(1) is large at $\boldsymbol{x}$ if the predicted $y(\boldsymbol{x})$, $\widehat{y}(\boldsymbol{x})$, is "much lower" than the current champion minimum, $y_n^{\min}$, i.e. both factors are large in this case

(2) is large at $\boldsymbol{x}$ if there is large uncertainty in our prediction of $y(\boldsymbol{x})$, i.e., $\sigma(\boldsymbol{x})$ is "large" and $\left|\frac{y_n^{\min} - \widehat{y}(\boldsymbol{x})}{\sigma(\boldsymbol{x})}\right|$ is "small" - remember that ϕ(h) is max at $h = 0$.

Example (Hartman Function)

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = -\sum_{i=1}^{4} c_i \exp\left\{ -\sum_{j=1}^{6} \alpha_{ij}(x_j - p_{ij})^2 \right\}$$

$0 \leq x_i \leq 1$ for $i = 1, \ldots, 6$, (α_{ij}) are given by

		j						
		1	2	3	4	5	6	c_i
i	1	10	3	17	3.5	1.7	8	1
	2	.05	10	17	.1	8	14	1.2
	3	3	3.5	1.7	10	17	8	3
	4	17	8	.05	10	.1	14	3.2

and (p_{ij}) are given by

		j					
		1	2	3	4	5	6
i	1	.1312	.1696	.5596	.0124	.8283	.5886
	2	.2329	.4135	.8307	.3736	.1004	.9991
	3	.2348	.1451	.3522	.2883	.3047	.6650
	4	.4047	.8828	.8732	.5743	.1091	.0381

Characteristics

$f(\bullet)$ has a unique has a unique minimum value of $-$ **3.322375**

EGO Algorithm

1. Initial sample $n = 51$
2. Additional observations 74 (125 total)
   (stopping when max expected improvement $\leq$ 0001

Result Minimum identified $\hat{\boldsymbol{x}}_{opt}$ $-$ **3.322309**

Expected Improvement-II

• **Setup**:

1. **Inputs** $\boldsymbol{x} = (\boldsymbol{x_c}, \boldsymbol{x_e})$ where
   $\boldsymbol{x_c}$ = **control (manufacturing, engineering design)** variables
   $\boldsymbol{x_e}$ = **noise (field, enviromental)** variables
   (model variables)

2. $\mathbf{X}_e$ has known distribution with support $\left\{\boldsymbol{x}_{e,j}^{sup}\right\}_{j=1}^{n_e}$ and probabilities $\left\{p_j\right\}_{j=1}^{n_e}$, i.e.,

$$p_j = P\left\{\mathbf{X}_e = \boldsymbol{x}_{e,i}^{sup}\right\} \text{ for } 1 \le j \le n_e$$

("target field conditions")

3. In this case $y(\boldsymbol{x_c}, \boldsymbol{X_e})$ **distribution** which we can summarize in the usual ways. The simplest summary of the performance of the design $\boldsymbol{x_c}$ is ,

$$\mu(\boldsymbol{x_c}) = E\{y(\boldsymbol{x}_c, \boldsymbol{X}_e)\} = \sum_{i=1}^{n_e} p_i \, y(\boldsymbol{x}_c, \boldsymbol{x}_i^{sup})$$

(Var($\boldsymbol{x_c}$) $= Var\{y(\boldsymbol{x}_c, \boldsymbol{X}_e)\}, \ldots)$

• **Goal** Find $\boldsymbol{x}_{c,opt} = \text{argmin}\, \mu(\boldsymbol{x_c})$

• **Goal** Find $\boldsymbol{x}_{c,opt} \equiv \text{argmin}\, \mu(\boldsymbol{x_c})$

• **Training Data** $\boldsymbol{x}_i^{tr} = (\boldsymbol{x}_{c,i}^{tr}, \boldsymbol{x}_{e,i}^{tr})$ and $y(\boldsymbol{x}_{c,i}^{tr}, \boldsymbol{x}_{e,i}^{tr})$ for $1 \leq i \leq n$

• **Bad News** We don't know (and won't compute)

$$\mu_n^{\text{min}} \equiv \min\{\mu(\boldsymbol{x}_{c,1}^{tr}),\ \mu(\boldsymbol{x}_{c,n}^{tr}), \ldots, \mu(\boldsymbol{x}_{c,n}^{tr})\}$$

because we would need $y(\boldsymbol{x}_{c,1}^{tr}, \boldsymbol{x}_{e,j}^{sup})$ for $1 \leq j \leq n_e$, for example,

$$\mu(\boldsymbol{x}_{c,1}^{tr}) = \sum_{j=1}^{n_e} p_i\, y(\boldsymbol{x}_{c,1}^{tr}, \boldsymbol{x}_{\boldsymbol{e},j}^{sup})$$

• **Good News** Can predict μ_n^{min} because we can predict each component, $\mu(\boldsymbol{x}_{c,i}^{tr})$, by

$$\widehat{\mu}(\boldsymbol{x}_{c,i}^{tr}) = \sum_{j=1}^{n_e} p_j\, \widehat{y}(\boldsymbol{x}_{c,i}^{tr}, \boldsymbol{x}_{\boldsymbol{e},j}^{sup})$$

and, in addition $\mu(\boldsymbol{x}_{c,i}^{tr})$ has a prior distribution induced by the $Y(\boldsymbol{x}_{c,i}, \boldsymbol{x}_{\boldsymbol{e},j}^{sup})$

$$M(\boldsymbol{x}_{c,i}^{tr}) = \sum_{j=1}^{n_e} p_i\, Y(\boldsymbol{x}_{c,i}^{tr}\, , \boldsymbol{x}_{\boldsymbol{e},j}^{sup})$$

Idealized Improvment Function Define the improvement at (generic) control variable site $\boldsymbol{x_c}$

$$I(\boldsymbol{x_c}) = \begin{cases} 0, & \mu(\boldsymbol{x_c}) \geq \mu_n^{\min} \\ \mu_n^{\min} - \mu(\boldsymbol{x_c}), & \mu(\boldsymbol{x_c}) < \mu_n^{\min} \end{cases}$$

$$= \max\{0, \mu_n^{\min} - \mu(\boldsymbol{x_c})\}$$

is the improvement in using the mean response at $\boldsymbol{x_c}$.
[all terms are **unknown** :-(, BUT all terms **can be predicted** :-)]

p

Idea of the Algorithm

1. Obtain a starting design, ie., set of initial inputs $\boldsymbol{x}_i^{tr} = (\boldsymbol{x}_{c,i}^{tr}, \boldsymbol{x}_{e,i}^{tr})$ at which to calculate y(•) (e.g., Space-filling "Latin Hypercube Design")

2. Evaluate $y(\bullet, \bullet)$ at the $\left\{(\boldsymbol{x}_{c,i}^{tr}, \boldsymbol{x}_{e,i}^{tr})\right\}_{i=1}^{n}$ starting design and use the training data to estimate any unknown correlation parameters $\boldsymbol{\xi}$

3. Choose $\boldsymbol{x}_{c,n+1}^{tr}$ to maximize the posterior expected improvement, i.e.,

$$\boldsymbol{x}_{c,n+1}^{tr} \equiv \text{argmax}\, E\left\{I(\boldsymbol{x_c}) | Y^n \equiv \left\{y\left(\boldsymbol{x}_i^{tr},\, y(\boldsymbol{x}_i^{tr})\right)\right\}_{i=1}^{n}\right\} \tag{1}$$

(the expected value of $I(\boldsymbol{x})$ under the stochastic process model

4. Choose $\boldsymbol{x}_{e,n+1}^{tr}$ in the enviromental variable space to minimize Var($\hat{\mu}(\boldsymbol{x}_{c,n+1}^{tr})$) where $\hat{\mu}(\boldsymbol{x}_{c,n+1}^{tr})$ is the predicted mean (over the environmental variables) at the selected next control variable

5. **Stop** when the maximum expected improvement is "small," and set

$$\hat{\boldsymbol{x}}_{c,opt} = \text{argmin}\, \hat{\mu}(\boldsymbol{x}_c),$$

where $\hat{\mu}(\boldsymbol{x})$ is the EBLUP of $\mu(\boldsymbol{x})$. OW, augment the current design.

Example:

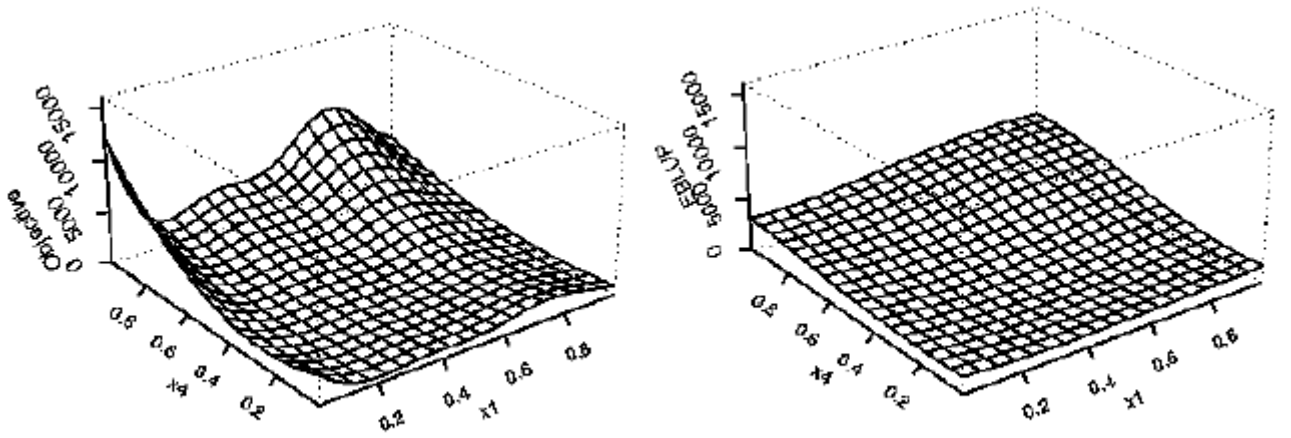
1. $y(x_1, x_2, x_3, x_4)$ has $\boldsymbol{x}_c = (x_1, x_2)$ and $\boldsymbol{x}_e = (x_3, x_4)$ and is based on the "Branin function" (Dixon and Szego, 1978)

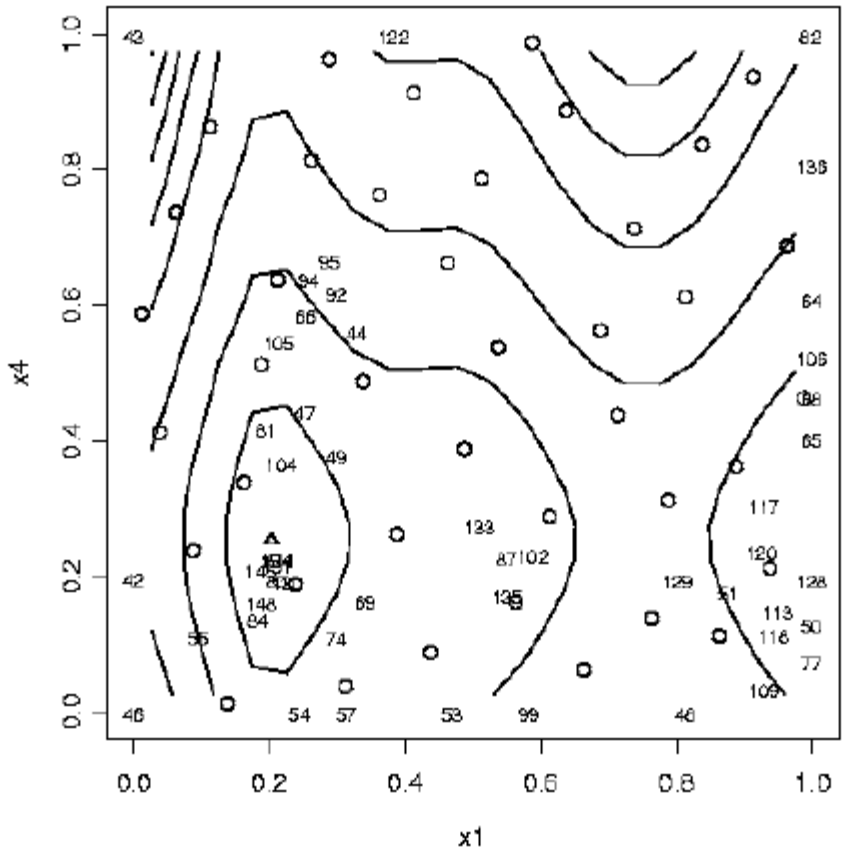
2. $0 \leq x_1, x_2, x_3, x_4 \leq 1$

3. $\boldsymbol{X}_e$:

		x_3			
		0.2	0.4	0.6	0.8
x_4	0.25	0.0375	0.0875	0.0875	0.0375
	0.50	0.0750	0.1750	0.1750	0.0750
	0.75	0.0375	0.0875	0.0875	0.0375

The **true** mean surface $\mu(x_1, x_2)$ and the estimated $\mu(x_1, x_2)$ based on a maximin LHD of size 40. The **global** minimizer $\boldsymbol{x}_{c,opt}$ is (0.2036, 0.2545) with $\mu(\boldsymbol{x}_{c,opt}) = 323.01174$ ($\mu(x_1, x_2)$ has **local** minimum at (1,0.25445) and (0.46287,0.25445))

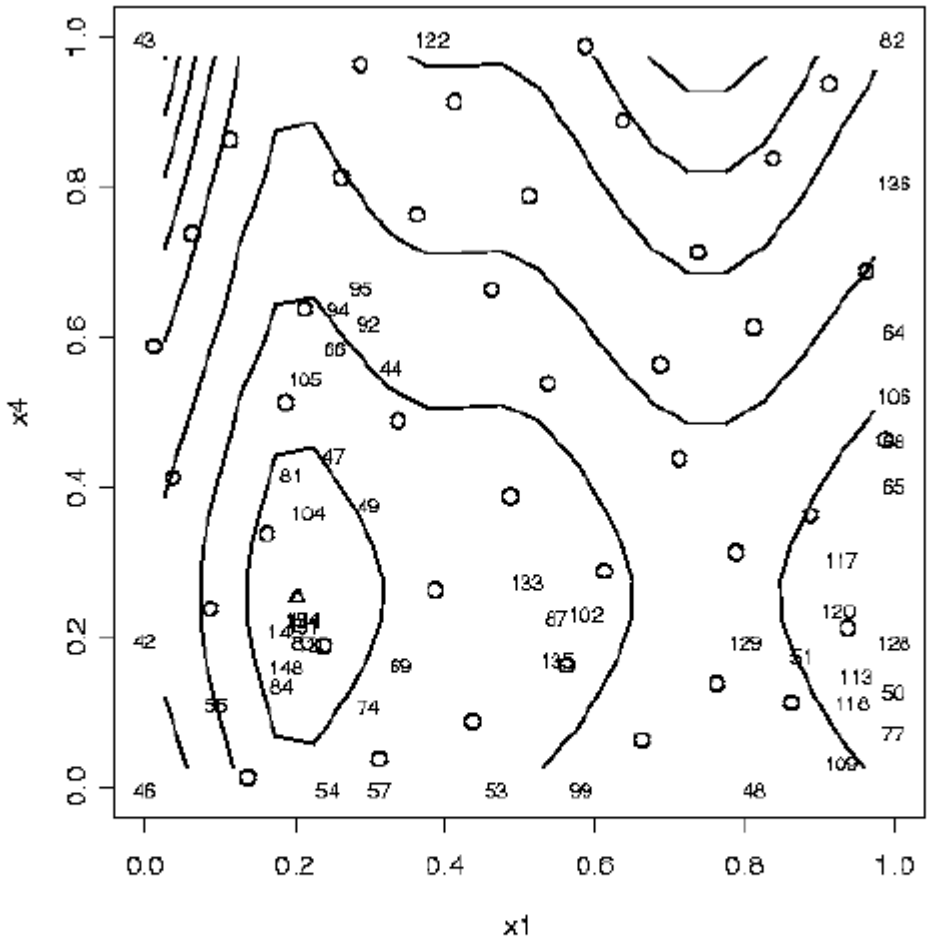




Projections of initial 40 points onto (x_1, x_2) plane

1. The initial fit is poor
2. Running the algorithm with the Matern correlation function, stopping occurs after 156 total $y(\bullet)$ function evaluations with $\hat{\boldsymbol{x}}_{c,opt} = (0.21096, 0.23324)$ (and $\mu(\hat{\boldsymbol{x}}_{c,opt}) = 326.67005$)
3. The global minimum of $\mu(\bullet)$ is within 1.15% of the true global minimum

Point	Max Expected Improvement
153	0.21567
154	3.61129×10^{-2}
155	5.91403×10^{-2}
156	2.48648×10^{-2}



Projections of additional design points onto (x_1, x_2) plane

6. Take Home Points

1. An increasing number of phenomenon that could be previously be studied only by physical experiments, can now be investigated using "computer experiments"

2. Modeling responses from computer experiment must account for the (highly) correlated nature of the output $y(\boldsymbol{x})$ over the input space. Predictive models are used to interpolate the computer response at untried locations

3. The design of most computer experiments is naturally sequential; we evaluate $y(\bullet)$ at one set of inputs $\boldsymbol{x}$, learn more about $y(\bullet)$, and select new inputs to achieve some objective

4. EGO algorithms balance sampling an output (surface) where the predictor indicates the minimum is located, with improving our knowledge of the surface at points having large prediction error.

5. The EGO algorithms can be modified to accomodate optimization with noisy output (simulation error/numerical or modeling bias) or from calibrated computer and physical experiments

6. Pattern search and other algorithms have been used for the same purpose (Booker, Dennis, Torozon, Trosset). Their usefulness depends on the cost of $y(\bullet)$ evaluations.

Some References

Booker, A. J., A. R. Conn, J. E. Dennis, P. D. Frank, M. Trosset, and V. Torczon (1995). Global modeling for optimization. Technical Report ISSTECH-95-032. Boeing Information & Support Services, Seattle, WA. Boeing/IBM/Rice Collaborative Project 1995 Final Report.

Booker, A. J., J. E. Dennis, P. D. Frank, D. B. Serafini, V. Torczon, and M. W. Trosset (1999). A rigorous framework for optimization of expensive functions by surrogates. *Structural Optimization* 17, 1-13.

Dennis, J. E. and V. Torczon (1997). Managing approximation models in optimization. In: Alexandrov, N. and M. Y. Hussaini, Eds., *Multidisciplinary Design Optimization: State of the Art*, SIAM, Philadelphia, 330-347.

Glad, T. and A. Goldstein (1977). Optimization of functions whose values are subject to small errors. *BIT* 17, 160-169.

Lehman, J., T. J. Santner, and W. I. Notz (2004) Design of Computer Experiments to Determine Robust Control Variables, Statistica Sinica, **14**, 571-580.

Lewis, R. M. and V. Torczon (1998a). A globally convergent augmented Lagrangian pattern search method for optimization with general constraints and simple bounds. Technical Report 98-31. Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, VA 23681-0001.

Sacks, J. W. J. Welch, T. J. Mitchell, and H. P. Wynn (1989). Design and analysis of computer experiments, *Statistical Science* 4, 409-435. Includes discussion.

Sasena, M. J., P. Papalambros, and P. Goovaerts (2002). Exploration of metamodeling sampling criteria for constrained global optimization. In: *Engineering Optimization*, pages 263-278.

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

Schonlau, M. (1997). Computer Experiments and Global Optimization. PhD thesis, University of Waterloo, Waterloo, Ontario.

Schonlau, M., and W. J. Welch (1996). Global optimization with nonparametric function fitting. In *Proceedings of the Section on Physical and Engineering Sciences*, American Statistical Association, 183-186.

Schonlau, M., W. J. Welch, and D. R. Jones (1997). A data-analytic approach to Bayesian global optimization. In: Proceedings of the Section on Physical and Engineering Sciences, American Statistical Association, 186-191.

Schonlau, M., W. J. Welch, and D. R. Jones (1998). Global versus local search in constrained optimization of computer models. In: Flournoy, N., W. F. Rosenberger, and W. K. Wong, Eds., *New Developments and Applications in Experimental Design*. **34**, 11-28, Institute of Mathematical Statistics, Hayward, CA.

Trosset, M. W. and V. Torczon (1997). Numerical optimization using computer experiments. Technical Report 97-38. Institute for Computer Applications in Sciences & Engineering, NASA Langley Research Center, Hampton, VA 23681-0001.

Williams, B. J., T. J. Santner, and W. I. Notz (2000) Sequential Design of Computer Experiments to Minimize Integrated Response Functions, *Statistica Sinica*, **10**, 1133-1152.

Williams, B. J., T. J. Santner, and W. I. Notz Sequential Design of Computer Experiments to Minimize Integrated Response Functions with Constraints, in preparation.