Surrogate and Reduced-Order Models

Problem: Difficult to obtain sufficient number of realizations of discretized PDE models for Bayesian model calibration, design and control.

Mass
$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0$$

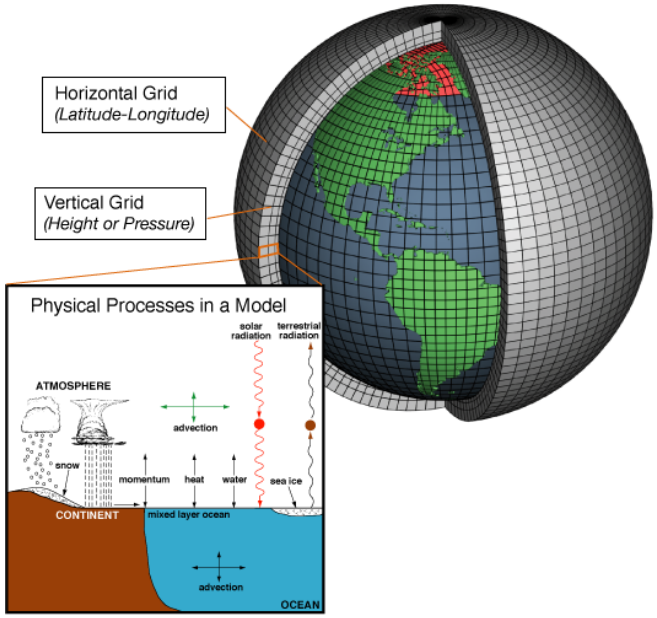
Momentum
$$\frac{\partial v}{\partial t} = -v \cdot \nabla v - \frac{1}{\rho}\nabla p - g\hat{k} - 2\Omega \times v$$

Energy
$$\rho c_V \frac{\partial T}{\partial t} + p\nabla \cdot v = -\nabla \cdot F + \nabla \cdot (k\nabla T) + \rho \dot{q}(T, p, \rho)$$

$$p = \rho RT$$

Water
$$\frac{\partial m_j}{\partial t} = -v \cdot \nabla m_j + S_{m_j}(T, m_j, \chi_j, \rho)\ ,\ j = 1, 2, 3,$$

Aerosol
$$\frac{\partial \chi_j}{\partial t} = -v \cdot \nabla \chi_j + S_{\chi_j}(T, \chi_j, \rho)\ ,\ j = 1, \cdots, J,$$



Constitutive Closure Relations: e.g.,

Solution: Construct surrogate models

- Also termed data-fit models, response surface models, emulators, meta-models
- Projection-based models often called reduced-order models

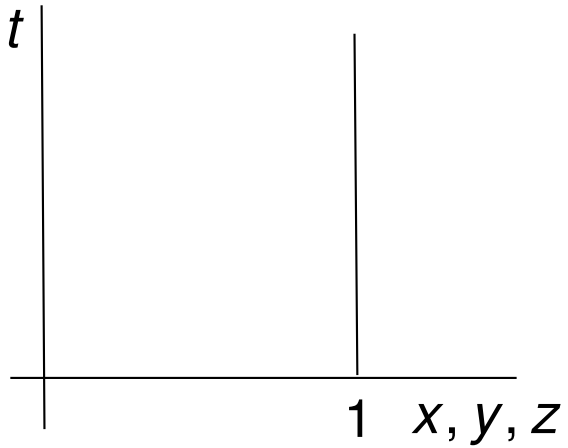
Surrogate Models: Motivation

Example: Consider the heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + f(q)$$

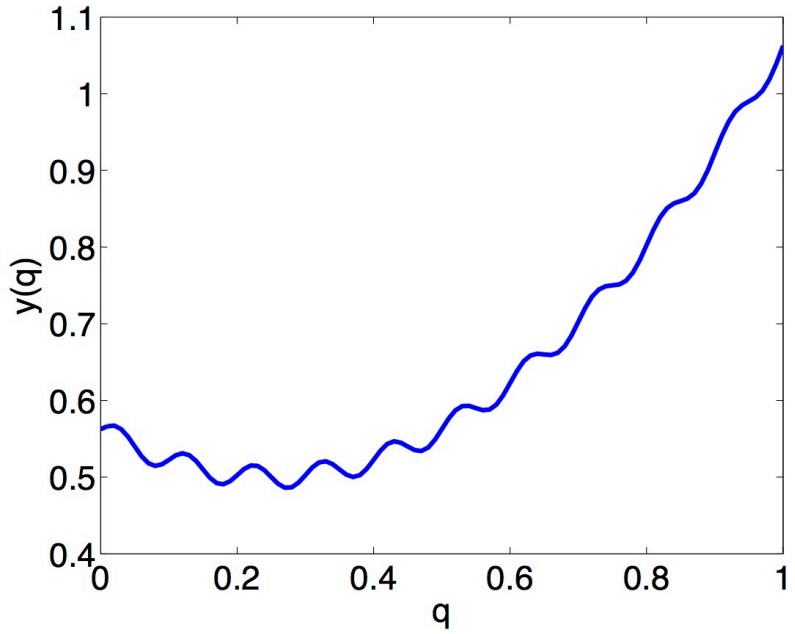
Boundary Conditions

Initial Conditions



with the response

$$y(q) = \int_0^1 \int_0^1 \int_0^1 \int_0^1 u(t, x, y, z)\, dx dy dz dt$$



Notes:

- **Requires approximation of PDE in 3-D**
- **What would be a simple surrogate?**

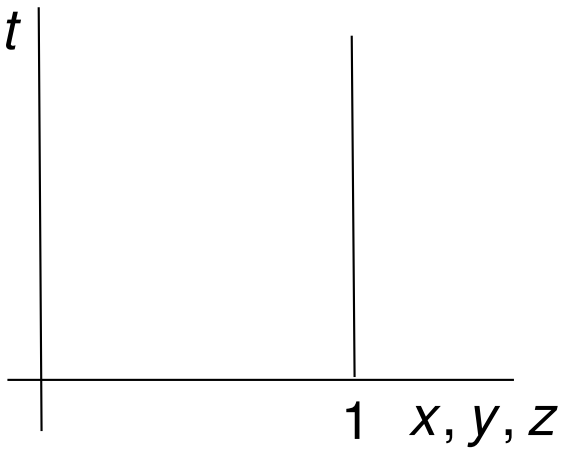
Surrogate Models: Motivation

Example: Consider the heat equation

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + f(q)$$

Boundary Conditions

Initial Conditions



with the response

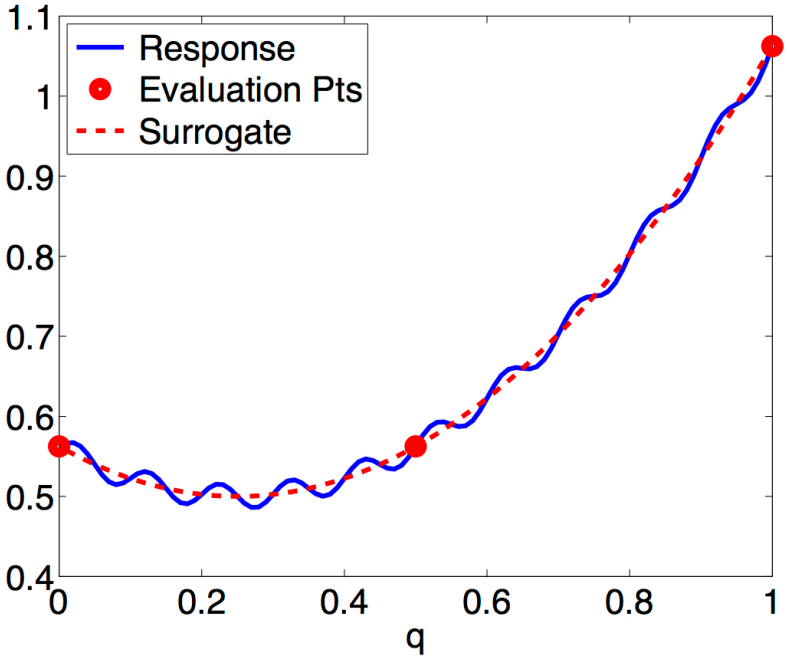
$$y(q) = \int_0^1 \int_0^1 \int_0^1 \int_0^1 u(t,x,y,z)\,dxdydzdt$$

Surrogate: Quadratic

$$y_s(q) = (q - 0.25)^2 + 0.5$$

Question: How do you construct a polynomial surrogate?

- **Regression**
- **Interpolation**



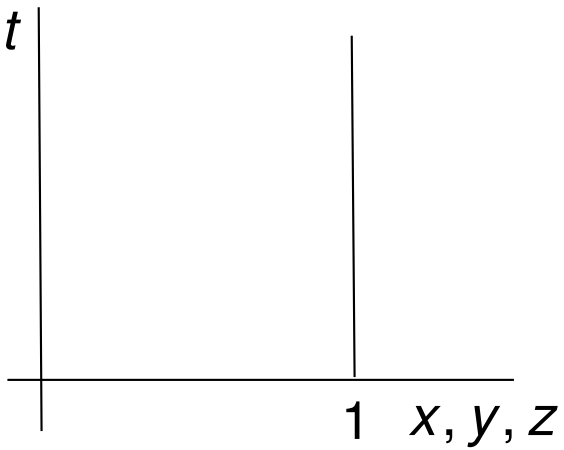
Surrogate Models

Recall: Consider the model

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} + f(q)$$

Boundary Conditions

Initial Conditions



with the response

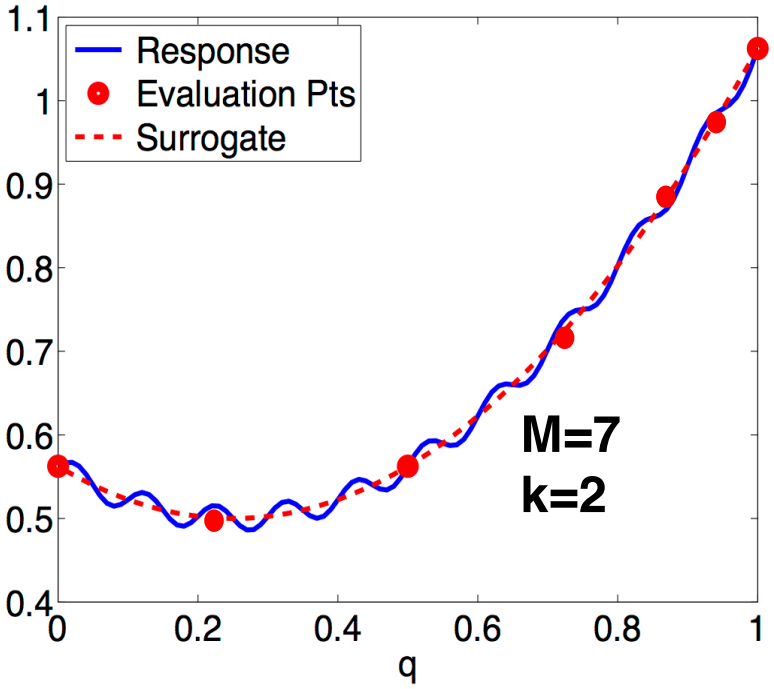
$$y(q) = \int_0^1 \int_0^1 \int_0^1 \int_0^1 u(t,x,y,z)\,dxdydzdt$$

Surrogate: Quadratic

$$y_s(q) = (q - 0.25)^2 + 0.5$$

Question: How do you construct a polynomial surrogate?

- **Interpolation**
- **Regression**



Surrogate and Reduced-Order Models

Issues:

- Techniques for regression versus interpolation
- Grid choices for interpolation
- Techniques for time-dependent problems

Data-Fit Models

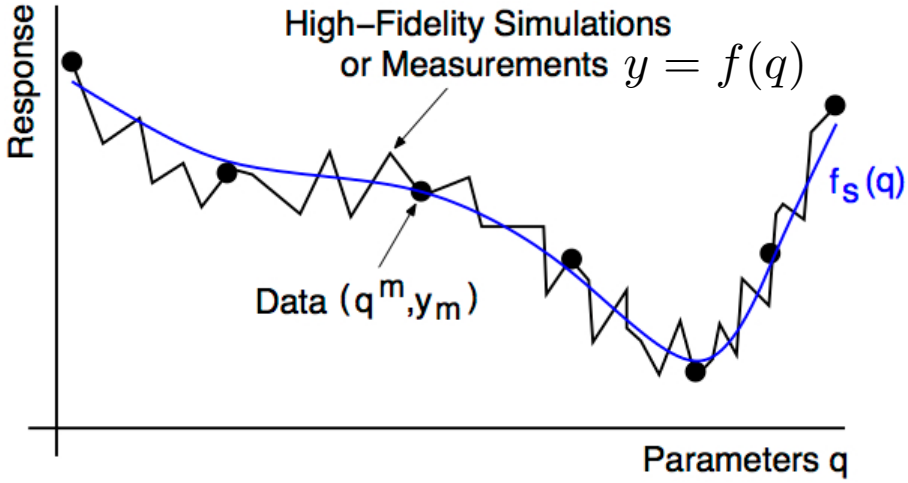
Notes:

• Often termed response surface models, surrogates, emulators, meta-models.

• Rely on interpolation or regression.

• Data can consist of high-fidelity simulations or experiments.

• Common techniques: polynomial models, kriging (Gaussian process regression), orthogonal polynomials.



Strategy: Consider high fidelity model

$$y = f(q)$$

with M model evaluations

$$y_m = f(q^m)\ ,\ \ m = 1, \ldots, M$$

Statistical Model: $f_s(q)$: Surrogate for $f(q)$

$$y_m = f_s(q^m) + \varepsilon_m\ ,\ \ m = 1, \ldots, M$$

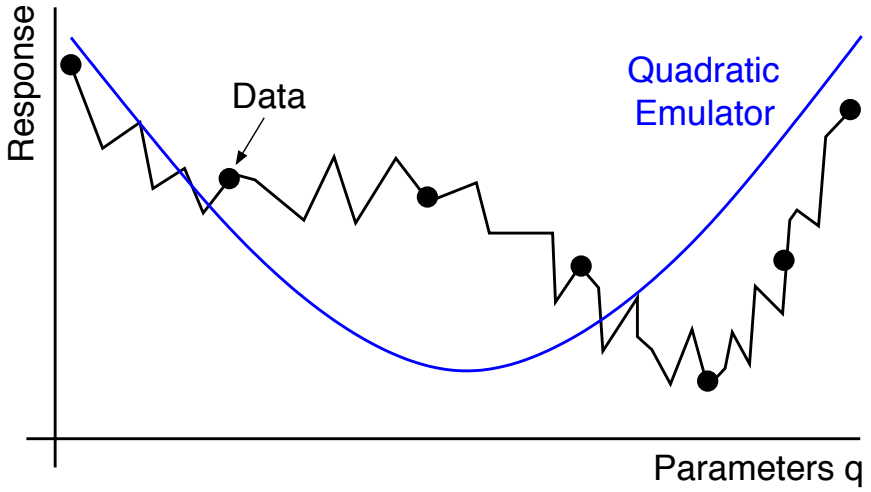
Data-Fit Models – Polynomial Emulator

Quadratic Emulator: Regression

$$f_s(q;\beta) = \beta_0 + \beta_1 q + \beta_2 q^2$$

Deterministic System: $y_{obs} = X\beta$

$$\begin{bmatrix} y_1 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} 1 & q^1 & (q^1)^2 \\ \vdots & & \vdots \\ 1 & q^M & (q^M)^2 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$



Least Squares Estimate:

$$\beta = [X^T X]^{-1} X^T y_{obs}$$

Notes:

- Good choice for optimization;
- Accurate approximation may require high-order polynomials;
- Does not provide uncertainty bounds for uncertainty quantification.

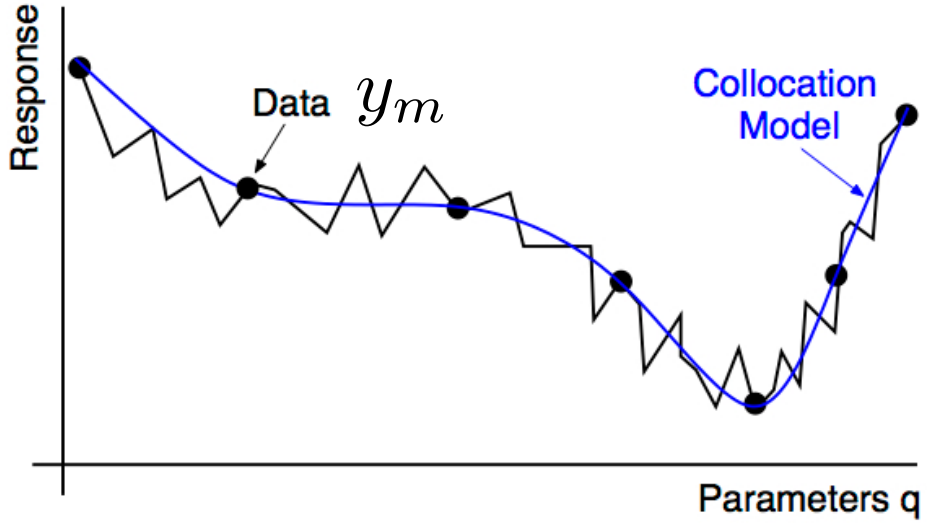
Data-Fit Models – Collocation

Strategy: Consider high fidelity model

$$y = f(q)$$

with M model evaluations

$$y_m = f(q^m)\ ,\ m = 1, \cdots, M$$



Collocation Surrogate:

$$y_s(q) = f_s(q) = \sum_{m=1}^{M} y_m L_m(q)$$

where $L_m(q)$ is a Lagrange polynomial, which in 1-D is represented by

$$L_m(q) = \prod_{\substack{j=0 \\ j \neq m}}^{M} \frac{q - q^j}{q^m - q^j} = \frac{(q - q^1) \cdots (q - q^{m-1})(q - q^{m+1}) \cdots (q - q^M)}{(q^m - q^1) \cdots (q^m - q^{m-1})(q^m - q^{m+1}) \cdots (q^m - q^M)}$$

Note:

$$L_m(q^j) = \delta_{jm} = \begin{cases} 0 & , \quad j \neq m \\ 1 & , \quad j = m \end{cases}$$

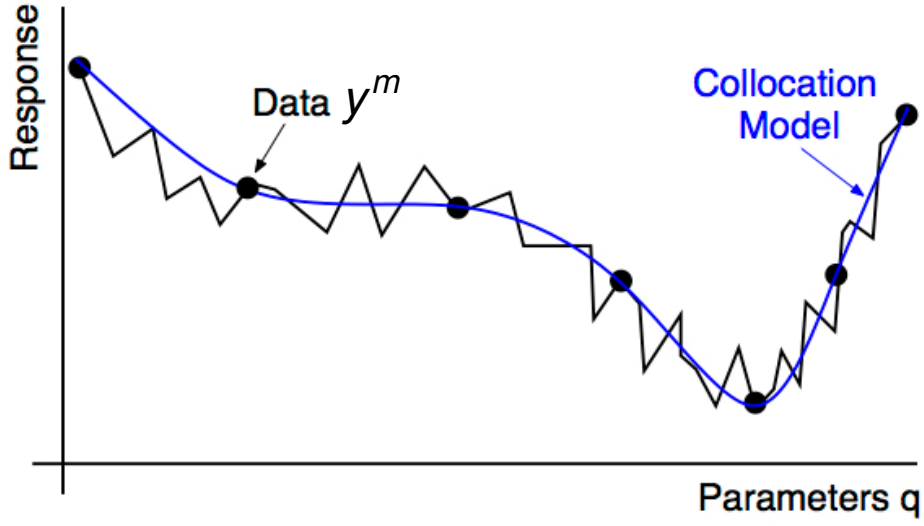
Result: $y_s(q^m) = f(q^m)$

Data-Fit Models – Collocation

Strategy: Consider high fidelity model

$$y = f(q)$$

with M samples y^m , $m = 1, \ldots, M$



Alternative: Collocation surrogate

$$y^M(q) = \sum_{m=1}^{M} y_m L_m(q)$$

where $L_m(q)$ is a Lagrange polynomial, which in 1-D, is represented by

$$L_m(q) = \prod_{\substack{j=0 \\ j \neq m}}^{M} \frac{q - q^j}{q^m - q^j} = \frac{(q - q^1)\cdots(q - q^{m-1})(q - q^{m+1})\cdots(q - q^M)}{(q^m - q^1)\cdots(q^m - q^{m-1})(q^m - q^{m+1})\cdots(q^m - q^M)}$$

Note:

$$L_m(q^j) = \delta_{jm} = \begin{cases} 0 & , \quad j \neq m \\ 1 & , \quad j = m \end{cases}$$

Result: $y^M(q^m) = y_m$

Properties:

- Easy to add points
- Method is nonintrusive and treats code as black box.

Data-Fit Models – Gaussian Process Emulator

High Fidelity Model: M evaluations

$$y_m = f(q^m)\ ,\ \ m = 1, \ldots, M$$

Statistical Model:

$$y_m = f_s(q^m, \beta) + \varepsilon_m\ ,\ \ \varepsilon_m \sim N(0, \sigma_0^2)$$

Gaussian Process: Kriging

$$f_s(q; \beta) = g^T(q)\beta + Z(q)$$

- $g^T(q)\beta$: Trend function

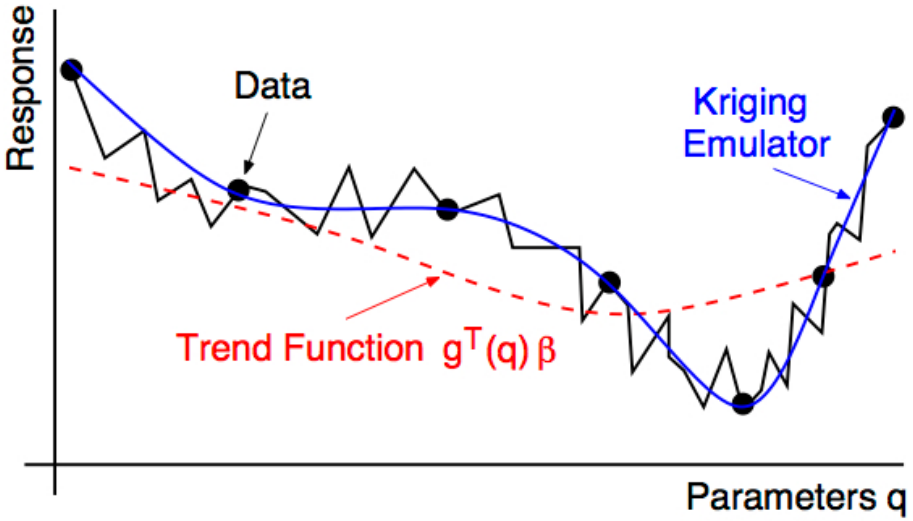
  Ordinary Kriging: $g^T(q)\beta = \beta_0$

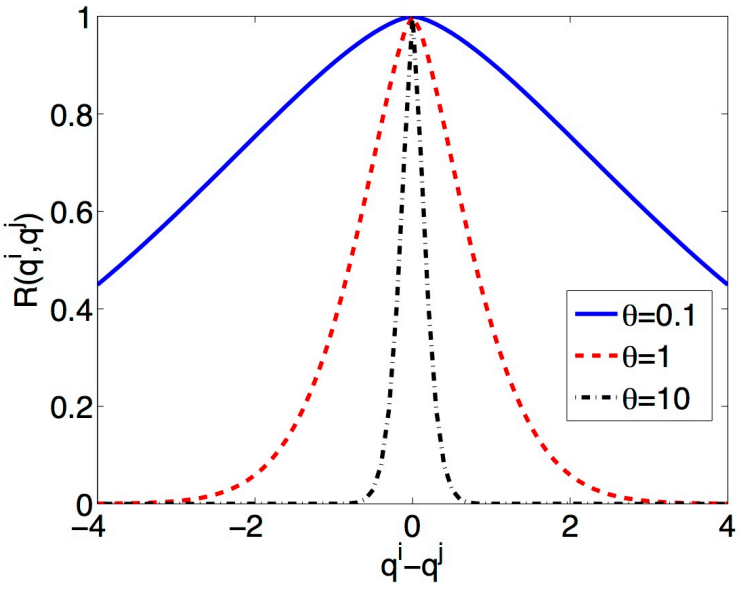
  Universal Kriging: $g^T(q)\beta = \sum_{k=0}^{K} \beta_k g_k(q)$

- $Z(q)$: Gaussian process with

$$\mathrm{cov}[Z(q^i), Z(q^j)] = \sigma^2 R(q^i, q^j) + \sigma_0^2 \delta(q^i, q^j)$$

$$R(q^i, q^j) = \exp\left(-\sum_{k=1}^{p} \left| \theta_k (q_k^i - q_k^j) \right|^{\gamma_k} \right)$$





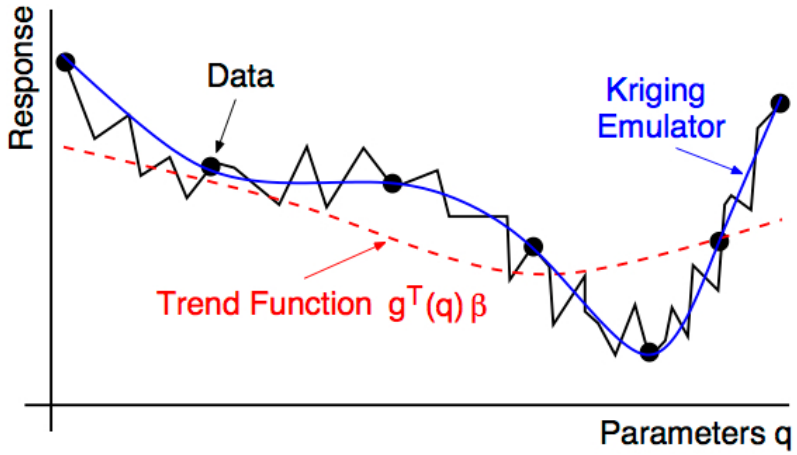
Note:

$$\delta(q^i, q^j) = \begin{cases} 1 & , \ \ \|q^j - q^i\| = 0 \\ 0 & , \ \ \text{else} \end{cases}$$

Gaussian Process Emulator

Statistical Model:

$$y_m = f_s(q^m, \beta) + \varepsilon_m\,,\ \ \varepsilon_m \sim N(0, \sigma_0^2)$$



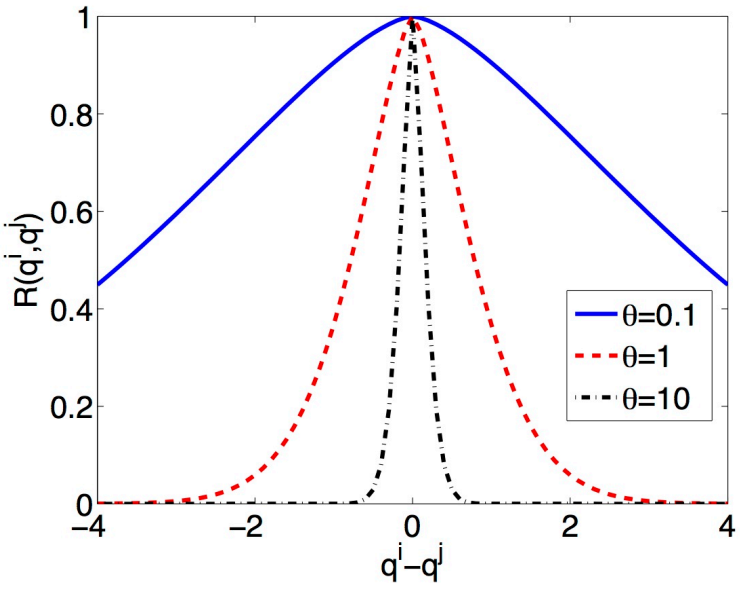
Gaussian Process: Kriging

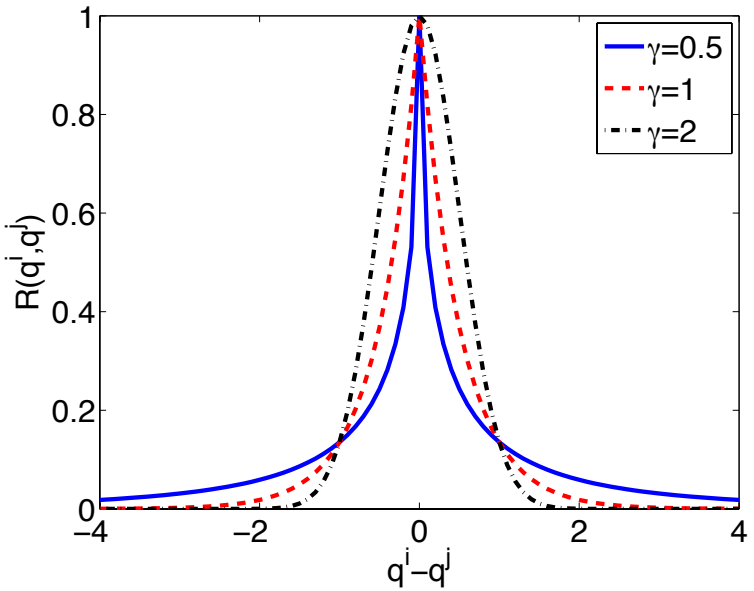
$$f_s(q;\beta) = g^T(q)\beta + Z(q)$$

$$\mathrm{cov}[Z(q^i), Z(q^j)] = \sigma^2 R(q^i, q^j) + \sigma_0^2 \delta(q^i, q^j)$$

$$R(q^i, q^j) = \exp\left(-\sum_{k=1}^{p} \left|\theta_k (q_k^i - q_k^j)\right|^{\gamma_k}\right)$$

Note: Hyper-parameters θ_k, γ_k tuned to achieve varying degrees of correlation





Gaussian Process Emulator

Statistical Model:

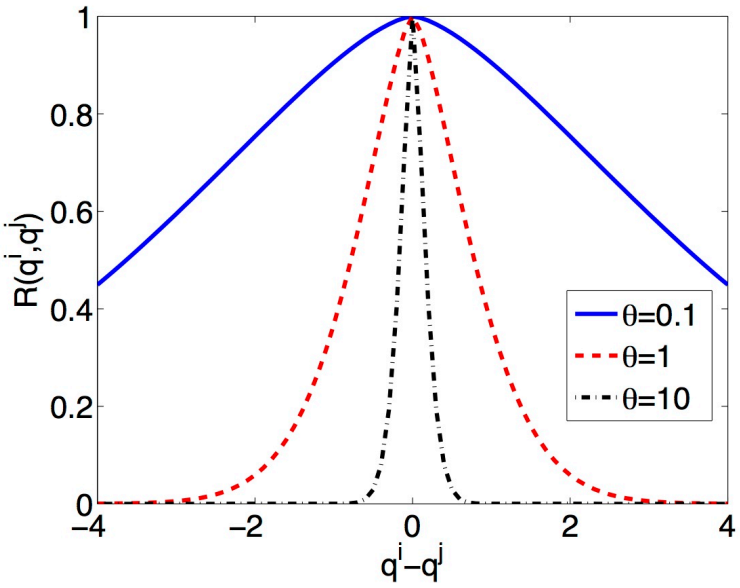
$$y_m = f_s(q^m, \beta) + \varepsilon_m\,,\ \ \varepsilon_m \sim N(0, \sigma_0^2)$$

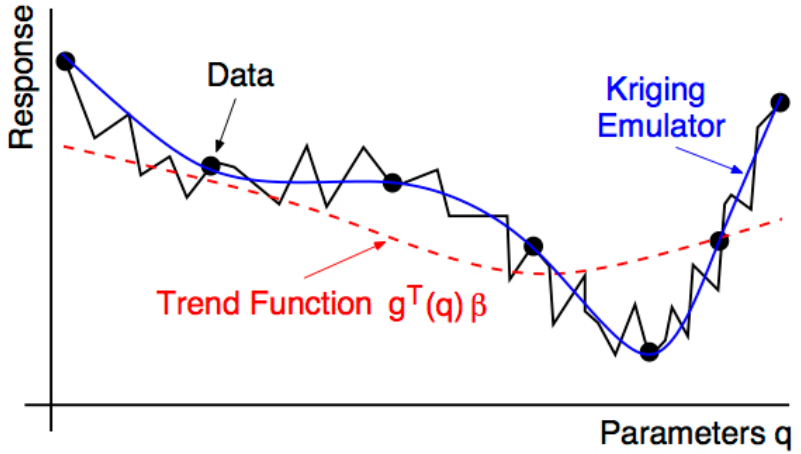
Gaussian Process: Kriging

$$f_s(q; \beta) = g^T(q)\beta + Z(q)$$

$$\mathrm{cov}[Z(q^i), Z(q^j)] = \sigma^2 R(q^i, q^j) + \sigma_0^2 \delta(q^i, q^j)$$

$$R(q^i, q^j) = \exp\left(-\sum_{k=1}^{p} \left|\theta_k (q_k^i - q_k^j)\right|^{\gamma_k}\right)$$

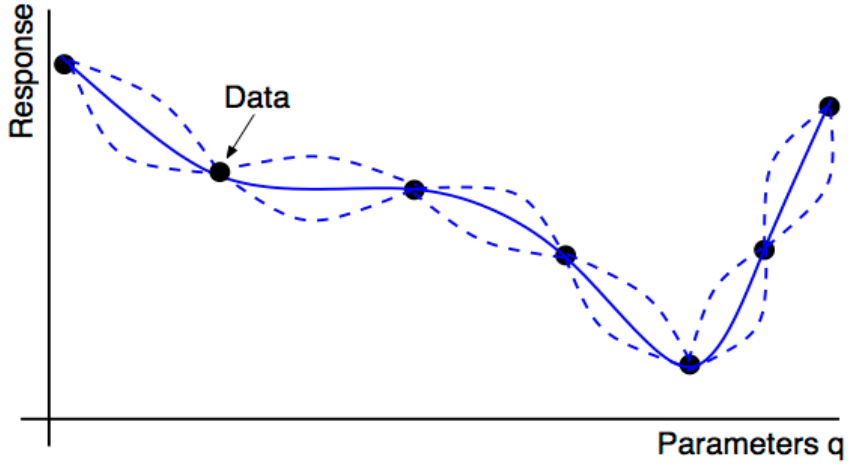




Note: In absence of observation noise, GP surrogate interpolates; i.e.,

$$y_m = f_s(q^m, \beta)\,,\ \ m = 1, \ldots, M$$

Uncertainty Bounds:



Gaussian Process Emulator

Observations: Take $y_s = [y_1, ..., y_M]^T$, $\mathbf{1} = [1, \cdots, 1]$ and $\mathcal{R}_{ij} = R(q^i, q^j)$

Likelihood:

$$L(\beta_0, \sigma^2; q) = \frac{1}{(2\pi\sigma^2)^{M/2}|\mathcal{R}|^{1/2}} \exp\left[-\frac{1}{2\sigma^2}(y_s - \beta_0\mathbf{1})^T\mathcal{R}^{-1}(y_s - \beta_0\mathbf{1})\right]$$

Log-likelihood:

$$\ell = -\frac{M}{2}\ln(2\pi) - \frac{M}{2}\ln(\sigma^2) - \frac{1}{2}\ln(\mathcal{R}) - \frac{1}{2\sigma^2}(y_s - \beta_0\mathbf{1})^T\mathcal{R}^{-1}(y_s - \beta_0\mathbf{1})$$

$$\Rightarrow \frac{\partial\ell}{\partial\beta_0} = \frac{1}{2\sigma^2}\mathbf{1}^T\mathcal{R}^{-1}(y_s - \beta_0\mathbf{1}) \cdot 2 = 0$$

$$\Rightarrow \beta_0(\theta, \gamma) = \left[\mathbf{1}^T\mathcal{R}^{-1}\mathbf{1}\right]^{-1}\left[\mathbf{1}^T\mathcal{R}^{-1}y_s\right]$$

and

$$\frac{\partial\ell}{\partial\sigma^2} = -\frac{M}{2}\frac{1}{\sigma^2} + \frac{1}{2(\sigma^2)^2}(y_s - \beta_0\mathbf{1})^T\mathcal{R}^{-1}(y_s - \beta_0\mathbf{1}) = 0$$

$$\Rightarrow \sigma^2(\theta, \gamma) = \frac{1}{M}\left[y_s - \beta_0\mathbf{1}\right]^T\mathcal{R}^{-1}\left[y_s - \beta_0\mathbf{1}\right]$$

Note: Optimize to infer θ, γ

Gaussian Process Prediction

Strategy: Assume we have M points $y_s = [f(q^1), \ldots, f(q^M)]^T$ and we want to make prediction y_p that maximizes likelihood of observed data y_s and prediction y_p. Form augmented vector

$$\widetilde{y} = \begin{bmatrix} y_s \\ y_p \end{bmatrix} = \begin{bmatrix} f(q^1) \\ \vdots \\ f(q^M) \\ y_p \end{bmatrix}$$

Let

$$r(q) = \begin{bmatrix} \text{cov}[Z(q^1), Z(q)] \\ \vdots \\ \text{cov}[Z(q^M), Z(q)] \end{bmatrix} = \begin{bmatrix} R(q^1, q) \\ \vdots \\ R(q^M, q) \end{bmatrix}$$

and

$$\widetilde{\mathcal{R}} = \begin{bmatrix} \mathcal{R} & r \\ r^T & 1 \end{bmatrix} = \begin{bmatrix} M \times M & M \times 1 \\ 1 \times M & 1 \times 1 \end{bmatrix}$$

Gaussian Process Prediction

Log-Likelihood:

$$\ell = -\frac{M}{2}\ln(2\pi) - \frac{M}{2}\ln\sigma^2 - \frac{1}{2}\ln|\widetilde{\mathcal{R}}| - \frac{1}{2\sigma^2}\left[\widetilde{y} - \mathbf{1}\beta_0\right]^T \widetilde{\mathcal{R}}^{-1}\left[\widetilde{y} - \mathbf{1}\beta_0\right]$$

Last term:

$$\frac{-1}{2\sigma^2}\begin{bmatrix} y_s - \mathbf{1}\beta_0 \\ y_p - \beta_0 \end{bmatrix}^T \begin{bmatrix} \mathcal{R} & r \\ r^T & 1 \end{bmatrix}^{-1} \begin{bmatrix} y_s - \mathbf{1}\beta_0 \\ y_p - \beta_0 \end{bmatrix}$$

Note: Form partitioned matrix

$$\widetilde{\mathcal{R}}^{-1} = \begin{bmatrix} \mathcal{R}^{-1} + \mathcal{R}^{-1} r \left(1 - r^T\mathcal{R}^{-1} r\right)^{-1} r^T\mathcal{R}^{-1} & -\mathcal{R}^{-1} r \left(1 - r^T\mathcal{R}^{-1} r\right)^{-1} \\ -\left(1 - r^T\mathcal{R}^{-1} r\right)^{-1} r^T\mathcal{R}^{-1} & \left(1 - r^T\mathcal{R}^{-1} r\right)^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} \widetilde{\mathcal{R}}_{11}^{-1} & \widetilde{\mathcal{R}}_{12}^{-1} \\ \widetilde{\mathcal{R}}_{21}^{-1} & \widetilde{\mathcal{R}}_{22}^{-1} \end{bmatrix}$$

Note:

$$\begin{bmatrix} y_s - \beta_0\mathbf{1} \\ y_p - \beta_0 \end{bmatrix}^T = \left[f(q^1) - \beta_0, \ldots, f(q^M) - \beta_0 \,|\, y_p - \beta_0\right]$$

Gaussian Process Prediction

Then

$$\begin{bmatrix} y_s - \mathbf{1}\beta_0 \\ y_p - \beta_0 \end{bmatrix}^T \begin{bmatrix} \mathcal{R} & r \\ r^T & 1 \end{bmatrix}^{-1} \begin{bmatrix} y_s - \mathbf{1}\beta_0 \\ y_p - \beta_0 \end{bmatrix}$$

$$= \begin{bmatrix} (y_s - \beta_0\mathbf{1})^T\widetilde{\mathcal{R}}_{11}^{-1} + (y_p - \beta_0)\widetilde{\mathcal{R}}_{21}^{-1} & (y_s - \beta_0\mathbf{1})^T\widetilde{\mathcal{R}}_{12}^{-1} + (y_p - \beta_0)\widetilde{\mathcal{R}}_{22}^{-1} \end{bmatrix} \begin{bmatrix} y_s - \beta_0\mathbf{1} \\ y_p - \beta_0 \end{bmatrix}$$

$$= F(y_s) + (y_p - \beta_0)\widetilde{\mathcal{R}}_{21}^{-1}(y_s - \beta_0\mathbf{1}) + (y_s - \beta_0\mathbf{1})^T\widetilde{\mathcal{R}}_{12}^{-1}(y_p - \beta_0) + (y_p - \beta_0)^2\widetilde{\mathcal{R}}_{22}^{-1}$$

$$= F(y_s) + \left[\frac{-2r^T\mathcal{R}^{-1}(y_s - \beta_0\mathbf{1})}{1 - r^T\mathcal{R}^{-1}r}(y_p - \beta_0) + \frac{1}{1 - r^T\mathcal{R}^{-1}r}(y_p - \beta_0)\right]$$

Thus

$$\frac{\partial \ell}{\partial y_p} = 0$$

$$\Rightarrow \frac{-1}{1 - r^T\mathcal{R}^{-1}r}(y_p - \beta_0) + \frac{r^T\mathcal{R}^{-1}(y_s - \beta_0\mathbf{1})}{1 - r^T\mathcal{R}^{-1}r} = 0$$

Best Unbiased Estimate for Mean:

$$y_p(q) = \mathbb{E}[f_s(q, \beta_0)] = \beta_0 + r^T(q)\mathcal{R}^{-1}(y_s - \beta_0\mathbf{1})$$

Gaussian Process Prediction

Best Unbiased Estimate for Mean:

$$y_p(q) = \mathbb{E}[f_s(q, \beta_0)] = \beta_0 + r^T(q)\mathcal{R}^{-1}(y_s - \beta_0 \mathbf{1})$$

Note:

$$r(q) = \begin{bmatrix} R(q^1, q) \\ \vdots \\ R(q^M, q) \end{bmatrix} \Rightarrow r(q^j) = \begin{bmatrix} R(q^1, q^j) \\ \vdots \\ R(q^M, q^j) \end{bmatrix}$$

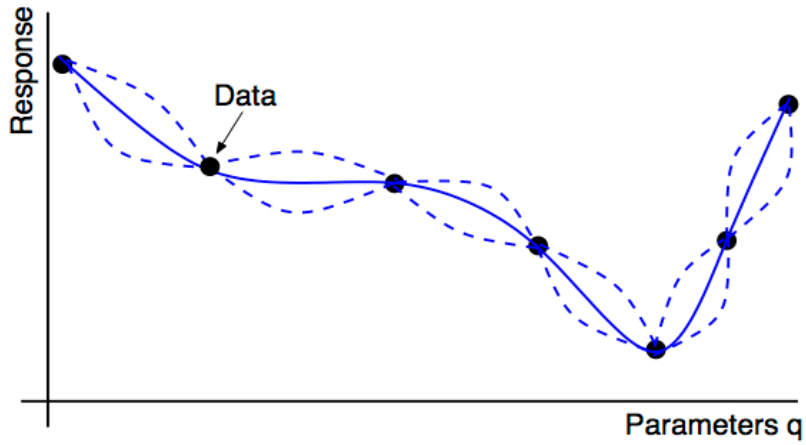
Then

$$r^T\mathcal{R}^{-1}[y_s - \beta_0 \mathbf{1}] = \left[R(q^1, q^j), \ldots, R(q^M, q^j)\right] \begin{bmatrix} \\ \mathcal{R}^{-1} \\ \\ \end{bmatrix} \begin{bmatrix} y_1 - \beta_0 \\ \vdots \\ y_M - \beta_0 \end{bmatrix}$$

$$= y_j - \beta_0$$

so

$$y_p(q^j) = y_j$$



Gaussian Process Prediction

Variance:

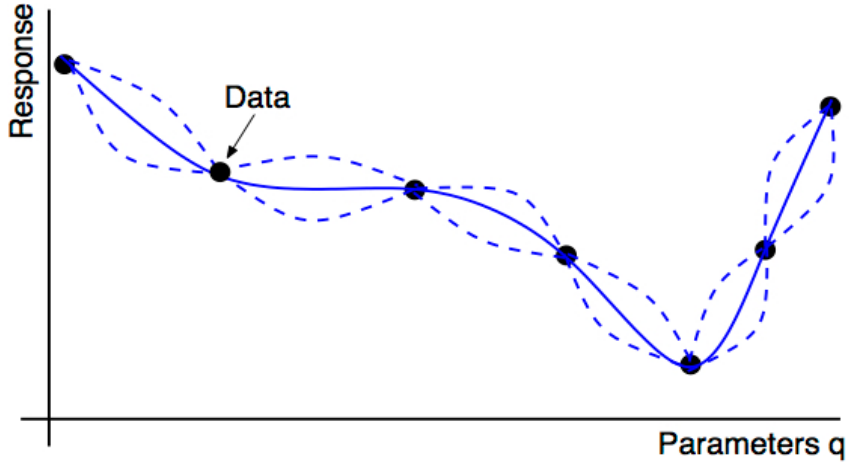
$$\mathrm{var}[f_s(q, \beta_0)] = \sigma^2 \left[1 - r^T \mathcal{R}^{-1} r\right]$$

where

$$\sigma^2 = \frac{1}{M} \left[y_s - \beta_0(\theta, \gamma)\mathbf{1}\right]^T \mathcal{R}^{-1} \left[y_s - \beta_0(\theta, \gamma)\mathbf{1}\right]$$

results from the condition

$$\frac{\partial \ell}{\partial \sigma^2} = 0$$



References:

- C. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006
- A.I.J. Forrester, A.Sóbester and A.J. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*, Progress in Astronautics and Aeronautics Volume 26, John Wiley and Sons, Chickester, UK, 2007

Example: Scaled Branin Function

Function:

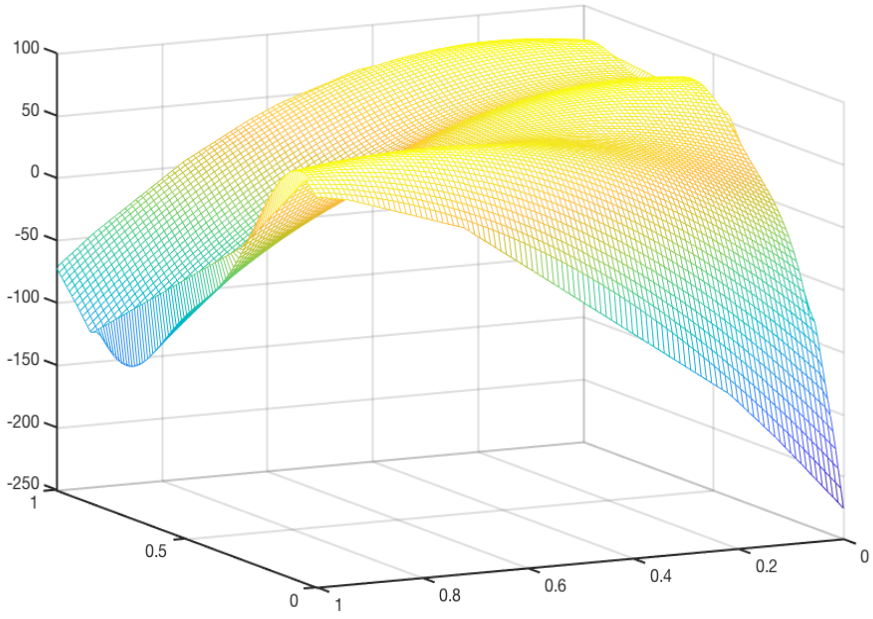
$$f(q_1, q_2) = a(x_2 - bx_1^2 + cx_1 - r)^2 + s(1-t)cos(x_1) + s$$

where $a = 1,\ b = \frac{5}{4\pi^2},\ c = \frac{5}{\pi},$

$$r = 6,\ s = 10,\ t = \frac{1}{8\pi}$$

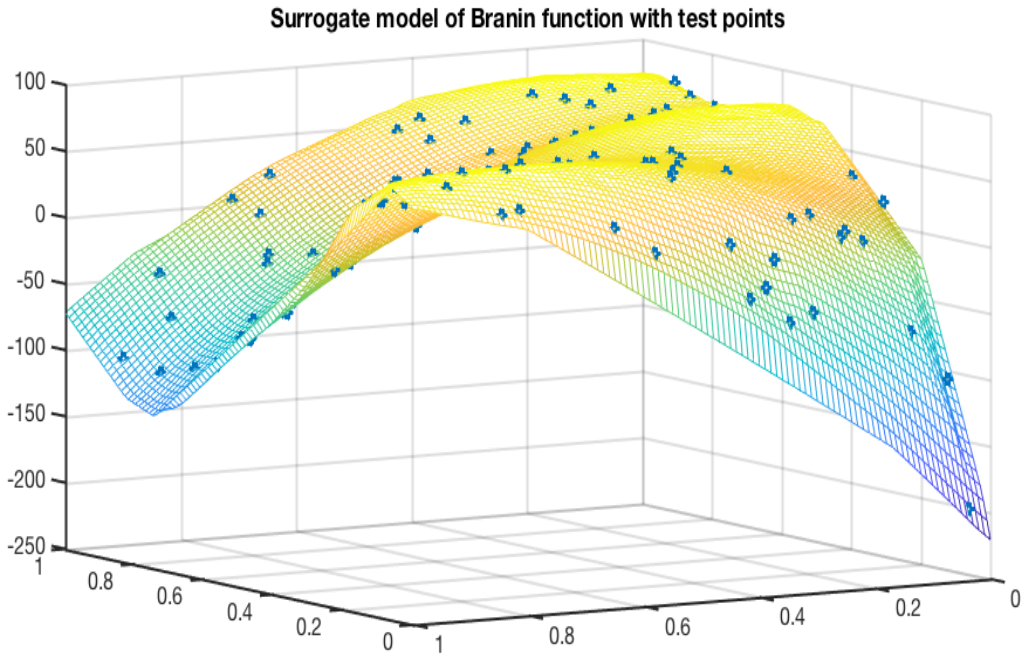
and $x_1 = 15q_1 - 5,\ x_2 = 15q_2$

- Function moved up 80 and flipped
- Used primarily to test optimization routines
  - 3 unique maxima



Example: Scaled Branin Function

- N=1000 randomly selected points $(q_1, q_2) \in [0, 1]$ used to construct surrogate model
- m=100 randomly selected test points used to assess accuracy
- MATLAB fitrgp and predict functions
  - Maximum likelihood estimate of hyperparameters
- RMS error over m predictions = 0.0460



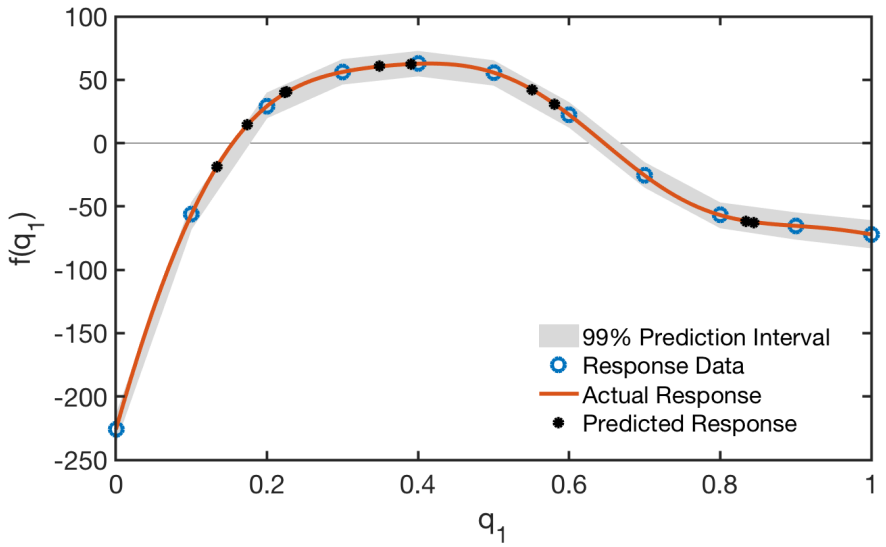
Example: Scaled Branin Function

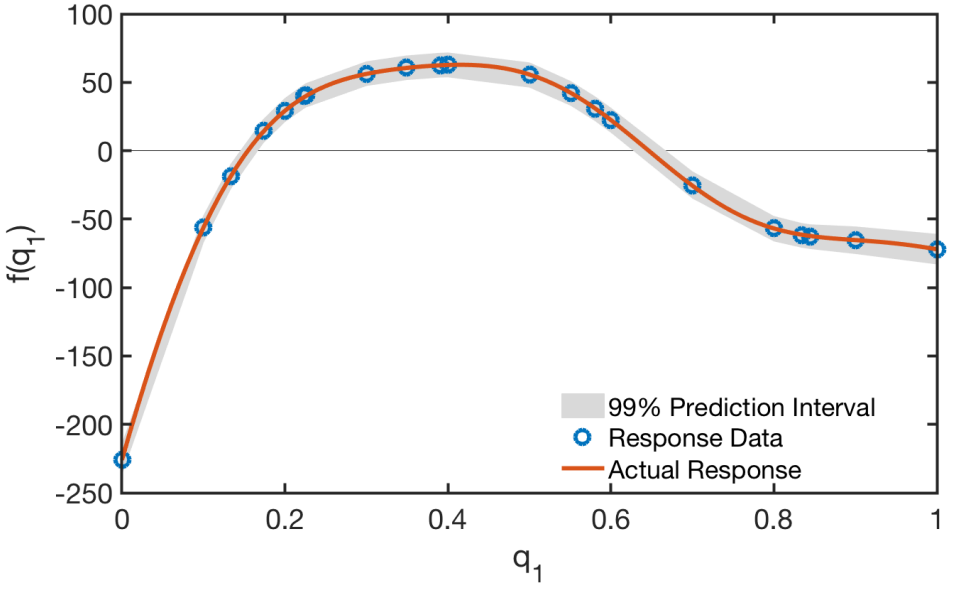
1-D Function:

$$f(q_1) = a((1+c)x_1 - bx_1^2 - r)^2 + s(1-t)cos(x_1) + s$$

- Same constants used as before
- Unique maximum
- MATLAB provides 99% prediction intervals

- Adding new data tightens prediction intervals
- Regression, so not exact at data points

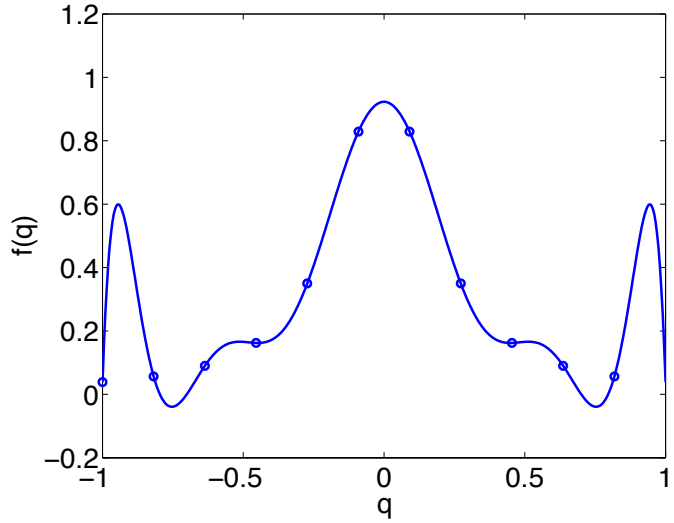


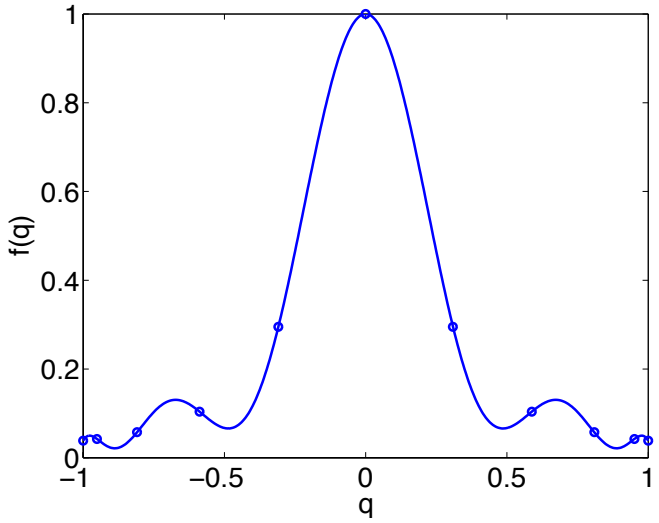


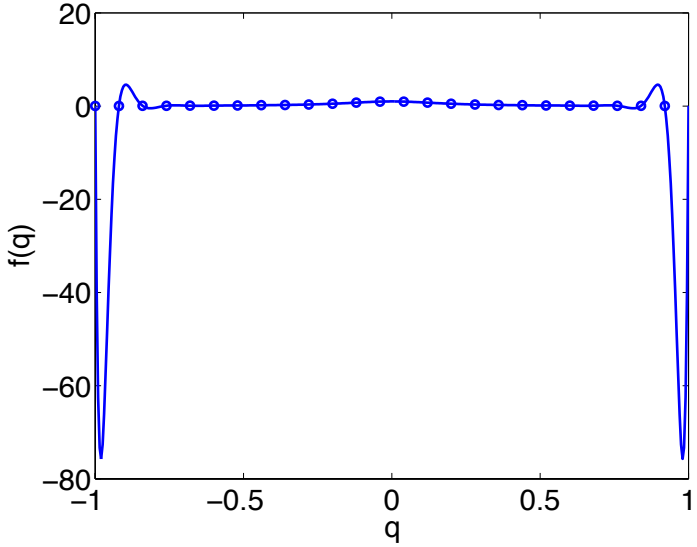
Surrogate Models – Grid Choice

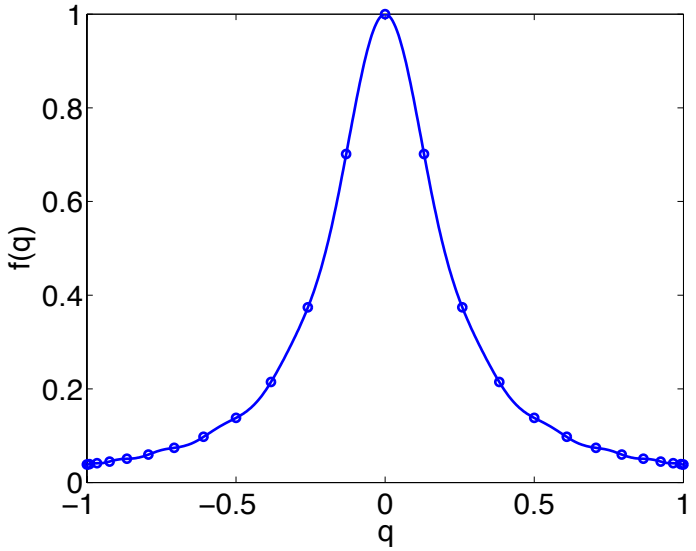
Example: Consider the Runge function $f(q) = \frac{1}{1+25q^2}$ with points

$$q^j = -1 + (j-1)\frac{2}{M}\ ,\ j = 1, \ldots, M \qquad q^j = -\cos\frac{\pi(j-1)}{M-1}\ ,\ j = 1, \ldots, M$$



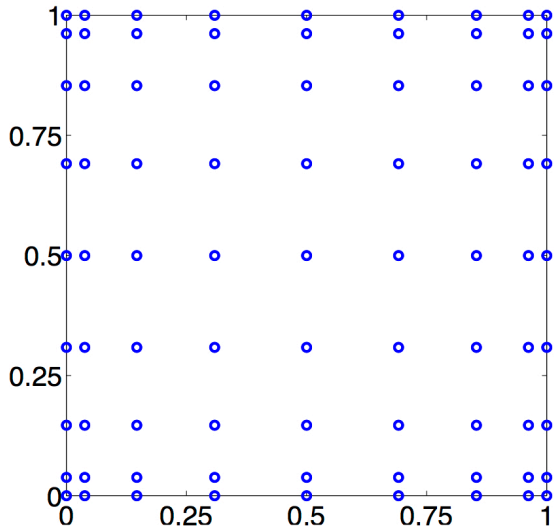




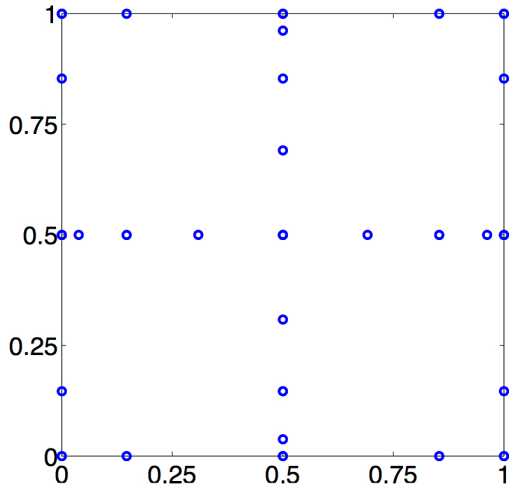


Sparse Grid Techniques

Tensored Grids: Exponential growth as a function of dimension



Sparse Grids: Same accuracy with significantly reduce number of points

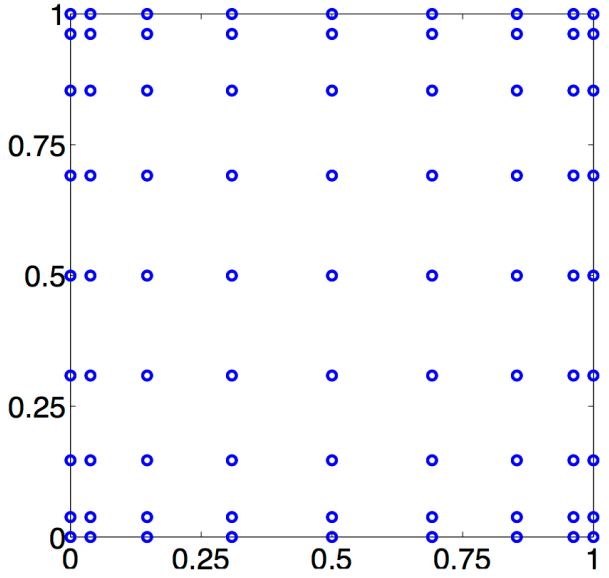


Motivation: Do not need full set of points to achieve same degree of accuracy

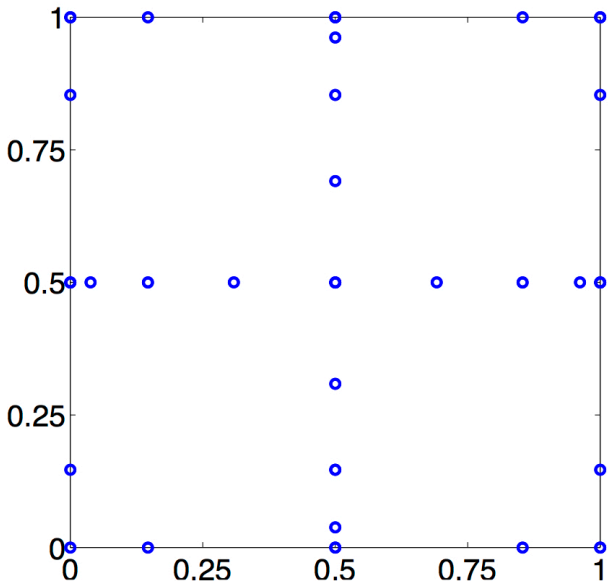
R									
0					1				
1				x		y			
2			x^2		xy		y^2		
3		x^3		x^2y		xy^2		y^3	
4	x^4		x^3y		x^2y^2		xy^3		y^4

Sparse Grid Techniques

Tensored Grids: Exponential growth



Sparse Grids: Same accuracy



p	R_ℓ	Sparse Grid $\mathcal{R}$	Tensored Grid $R = (R_\ell)^p$
2	9	29	81
5	9	241	59,049
10	9	1581	$> 3 \times 10^9$
50	9	171,901	$> 5 \times 10^{47}$
100	9	1,353,801	$> 2 \times 10^{95}$

Reduced-order Models

Based on Projections: e.g., Evolution model: for all $v \in V$

$$\int_{\mathcal{D}} \frac{\partial u}{\partial t} v dx + \int_{\mathcal{D}} N(u,q) S(v) dx = \int_{\mathcal{D}} F(q) v dx$$

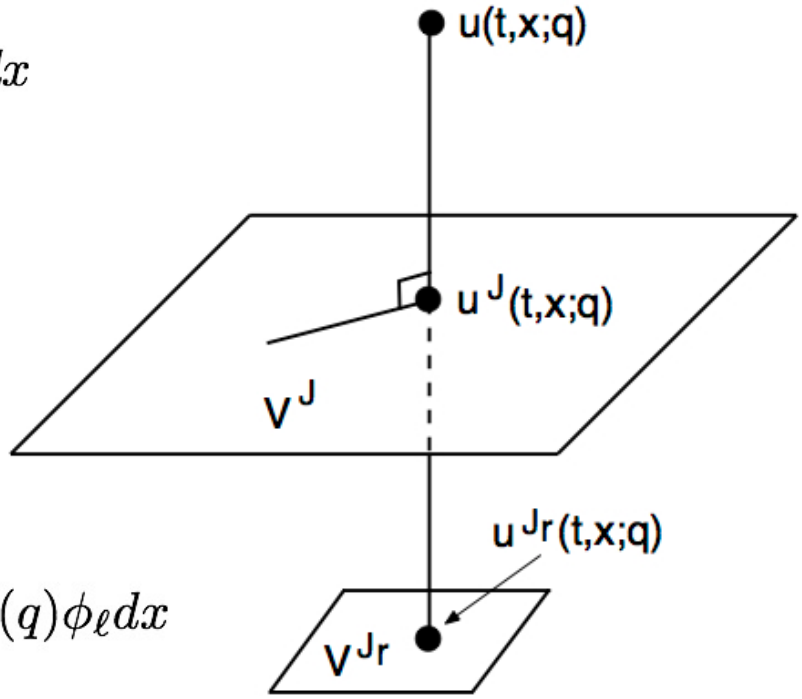
Full-Order Approximate Solution

$$u^J(t,x;q) = \sum_{j=1}^{J} u_j(t)\phi_j(x)$$

satisfies

$$\int_{\mathcal{D}} \frac{\partial u^J}{\partial t} \phi_\ell dx + \int_{\mathcal{D}} N(u^J,q) S(\phi_\ell) dx = \int_{\mathcal{D}} F(q) \phi_\ell dx$$

on high-dimensional space V^J = span(ϕ_j).



Goal: Construct reduced-order solution

$$u^{J_r}(t,x;q) = \sum_{j=1}^{J_r} u_j(t)\phi_j(x)$$

on low-dimensional space V^{J_r} where $J_r << J$

Note:

- Basis construction crux of method.
- Eigenfunctions
- Proper Orthogonal Decomposition (POD)

Reduced-order Models

Example: Eigenfunctions for heat equation

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \quad , \quad 0 < x < L \, , t > 0$$
$$T(t,0) = T(t,L) = 0 \quad , \quad t > 0$$
$$T(0,x) = T_0(x) \quad , \quad 0 < x < L$$

Weak Formulation: For all $\phi \in H_0^1(0,L)$

$$\int_0^L \frac{\partial T}{\partial t}\phi dx + \alpha \int_0^L \frac{\partial T}{\partial x}\frac{d\phi}{dx} dx = 0$$

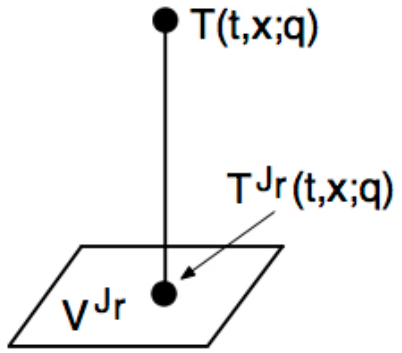
Reduced-Order Basis: $\phi_j^r(x) = \sin\left(\frac{j\pi x}{L}\right)$

Reduced-Order Model:

$$T^{J_r}(t,x) = \sum_{j=1}^{J_r} T_j(t) \sin\left(\frac{j\pi x}{L}\right)$$

satisfying

$$\int_0^L \frac{\partial T^{J_r}}{\partial t}\phi_i^r dx + \alpha \int_0^L \frac{\partial T^{J_r}}{\partial x}\frac{d\phi_i^r}{dx} dx = 0$$



Reduced-Order System:

$$\frac{dT_j}{dt} = -\left(j\pi/L\right)^2 T_j$$
$$T_j(0) = \gamma_j$$

Reduced-order Models

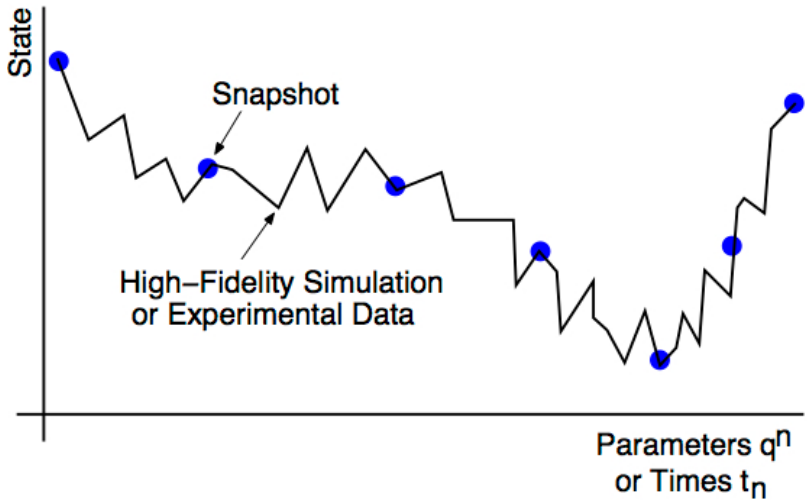
Proper Orthogonal Decomposition (POD):

•Based on snapshots $v_n = u^J(t_n, x; q)$ or $u^J(t, x; q^n)$

•Set typically highly redundant;

•POD extracts coherent structure having largest mean square projection onto set of observations.

•Related to Karhunen-Loeve expansion and singular value decomposition.



Strategy: Seek basis function of the form

$$\phi(x) = \sum_{i=1}^{N} a_i v_i(x)$$

where the coefficients a_i are chosen to maximize

$$\frac{1}{N}\sum_{i=1}^{N} |\langle v_i, \phi\rangle|^2 \quad \text{subject to } \langle \phi, \phi\rangle = \|\phi\|^2 = 1.$$

Reduced-Order Basis:

$$\phi_j^r(x) = \sum_{i=1}^{N} \frac{1}{\sqrt{N\lambda_j}} a_i^j v_i(x)$$

where (λ_j, a_i^j) are eigenpair

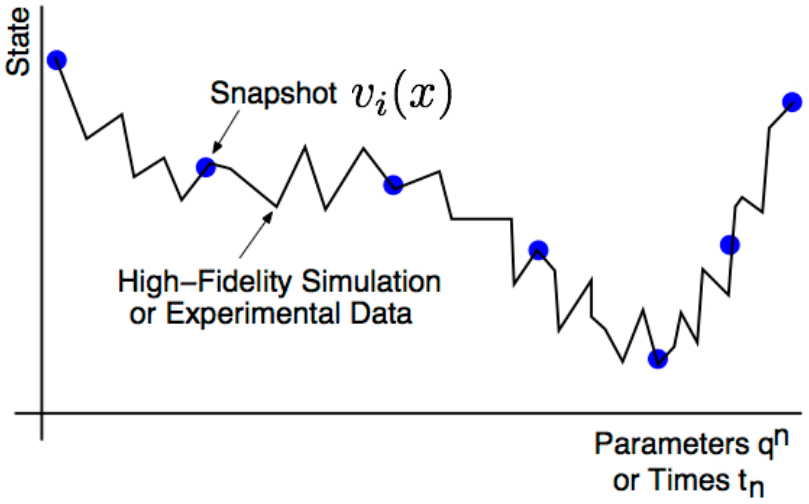
Reduced-order Models

Strategy: Seek basis function of the form

$$\phi(x) = \sum_{i=1}^{N} a_i v_i(x)$$

where the coefficients a_i are chosen to maximize

$$\frac{1}{N}\sum_{i=1}^{N} |\langle v_i, \phi\rangle|^2 \quad \text{subject to } \langle \phi, \phi\rangle = \|\phi\|^2 = 1.$$



Note: $C(x,y) = \frac{1}{N}\sum_{i=1}^{N} v_i(x)v_i(y) \Rightarrow \langle R\phi, \phi\rangle = \frac{1}{N}\sum_{i=1}^{N} |\langle v_i, \phi\rangle|^2 \Rightarrow$ R is positive, self-adjoint operator

$$R\phi = \int_{\mathcal{D}} C(x,y)\phi(y)dy \qquad \langle R\phi, \psi\rangle = \langle \phi, R\psi\rangle$$

Equivalent Problem: Find largest eigenvalue of

$$R\phi = \lambda\phi \quad \text{subject to } \|\phi\| = 1$$

or

$$\int_{\mathcal{D}} C(x,y)\phi(y)dy = \lambda\phi \quad \text{with } \|\phi\| = 1$$

Reduced-Order Basis:

$$\phi_j^r(x) = \sum_{i=1}^{N} \frac{1}{\sqrt{N\lambda_j}} a_i^j v_i(x)$$

where (λ_j, a_i^j) are eigenpair