Gradient Enhanced Universal Kriging Model for Uncertainty Quantification in Reactor Safety Simulations.

Brian Lockwood, University of Wyoming

Mihai Anitescu, Argonne National Laboratory

VVUHPC Workshop, collocated with SC 2010,

November 2010

Uncertainty Quantification

- Uncertainty analysis of model predictions: given data about uncertainty parameters $u \in R^p$ and a code that creates output from it $y = f(u)$ characterize y.
- End-goal management under uncertainty: Choose the decision variables in a way that chooses the optimal outcome while accounting for the uncertainty.

- Challenge one: create model for $u \in R^p$ from data. It does not need to be probabilistic (see Helton and Oberkampf RESS special issue) but it tends to be. What is the statistical model*?
- Challenge two: uncertainty propagation. Since f is expensive to compute, we cannot expect to compute a statistic of y very accurately from direct simulations alone (and there is also curse of dimensionality; exponential growth of effort with dimension). How do I propagate the model if the code is very computationally intensive?
- Challenge three: design under uncertainty. How do I compute the best decision for systems with a large number of states AND a high-dimensional uncertainty space?

Uncertainty Propagation at Higher End of Computational Scale

- High performance computing techniques, and modern approaches to complex physical and engineering research lead to the use of complex simulation models. At higher resolution scale, they require very large computing times even on advanced architectures.

- In general, analysis of a model that barely fits the available hardware, may exceed the available computational power.
- We can rely only on techniques that admit a very small number of model runs.
- We must do better than brute-force sampling, since statistics of quantities of interest (such as quantiles) are unlikely to converge in small number of model runs.

- We study the ability of Gradient Enhanced Universal Kriging Model to provide accurate uncertainty quantification based out of very few samples.
- We rely on :

- Capability of the model to also output derivative information
- Expectation that the response is smooth on the scale of interest.
- Gaussian Processes to quantify propagation uncertainty.

Problem Statement

Setup

- Given a "big code" computational model with

$$F(T,R)=0 \qquad J=J(T)$$

$$R(T,\alpha)=R_0(T)\cdot(1+\Delta R(T,\alpha))$$

- Where:
  - State variables: $T=(T_1,T_2,...,T_n)$
  - Input Parameters $R=(R_1,R_2,...,R_N)$
  - Uncertainties $\Delta R=(\Delta R_1,\Delta R_2,...,\Delta R_N)$
  - Uncertainty Parameters: $\alpha=(\alpha_1,\alpha_2,...,\alpha_m)$
  - Output Functional of Interest: $\Im(\alpha):=J(T(\alpha))$
  - Solving Code $\alpha\rightarrow T(\alpha)\rightarrow J(T(\alpha)) \quad \text{where} \quad F(T(\alpha),R(T(\alpha)))=0$

- Problem: given a probabilistic distribution for α determine a statistic of interest for $J(T(\alpha))$

- Typical answer: brute-force sampling: produce, for a large N, $\{\alpha_i\}_{i=1}^N\sim\alpha\Rightarrow\{\Im(\alpha_i)\}_{i=1}^N\sim\Im(\alpha)$ and compute the intended statistic.
- It is folly here, for "big code": the time to compute $\Im(\alpha)$ is large.
- We must do better use of the information.

Modeling Uncertainty Propagation

- Idea: use the few samples to model the system response.
- Ansatz: system response is "smooth" so maybe I can get by with a limited number of samples.
- But since I do uncertainty quantification, I must model the error I am making.

- Possible approaches:
  - Use sensitivity approach $\Im(\alpha) \sim \Im(\alpha_0) + \nabla_\alpha \Im(\alpha_0)(\alpha - \alpha_0) + \ldots$
    With adjoints, O(1) "big code" runs, but large bias, and error hard to model.
  - Use L2 regression $\Im(\alpha) \sim \sum_{i=1,K} a_k \psi_k(\alpha)$. Robust and least subject to curse of dimensionality, but introduces bias whose error model is hard to gauge.
  - Use Kriging. $\Im(\alpha) \sim GP(0, k(\alpha, \alpha'; \theta))$; $\tilde{\Im}(\alpha) \sim GP(0, k(\alpha, \alpha'; \theta)) \big| \{\Im(\alpha_i)\}_{i=1}^{N}$. The bias goes to 0 under some conditions, but it is an interpolation, so subject to COD.
- But … what if we use all 3? Gradient-enhanced universal Kriging (GEUK). Hopefully, small exposure to COD (as regression), small/reducible bias as Kriging efficient use of information (as sensitivity).

Preview of the method

(A) First-order gradient information on the model

PLUS

(B) Polynomial mean function

PLUS

(C) Uncertainty Propagation Model using Gaussian Processes, Kriging

=

Gradient-Enhanced Universal (non-constant mean function) Kriging, GEUK

Gaussian Processes approach, technical details:

- We assume that the response of the system can be represented as a Gaussian process with explicit mean function and specified covariance function governed by a set of parameters (*hyperparameters*):

$$\hat{\Im}(x) = N(R(x)a, K(x,x;\theta))$$

- Covariance matrix with derivative information is given by a block form:

$$K = \mathrm{cov}[Y,Y] = \begin{pmatrix} \mathrm{cov}[J,J] & \mathrm{cov}[J,\nabla J] \\ \mathrm{cov}[\nabla J,J] & \mathrm{cov}[\nabla J,\nabla J] \end{pmatrix}$$

- Regression parameters are computed as $a = (\Psi^T K^{-1} \Psi)\Psi^T K^{-1} Y$
or $a = \left(H^T K^{-1} H\right)^{-1} \cdot H^T K^{-1} \cdot Y$, with $H = \begin{pmatrix} \Psi \\ \nabla\Psi \end{pmatrix}$ $Y = \begin{pmatrix} J \\ \nabla J \end{pmatrix}$

- The mean and variance of the model are now predicted as

$$\mu[J] = \begin{pmatrix} \mathrm{cov}(Y_{i,:}, Y_{:,j}) & W \end{pmatrix} K^{-1} Y + R(x)a$$

$$\mathrm{var}[J] = \mathrm{cov}(S,S) - \begin{pmatrix} \mathrm{cov}(Y_{i,:}, Y_{:,j}) & W \end{pmatrix} K^{-1} \cdot \begin{pmatrix} \mathrm{cov}(Y_{i,:}, Y_{:,j}) \\ W \end{pmatrix} + R(x)\left(H^T K^{-1} H\right)^{-1} R(x)^T$$

- We now need to assume a functional form of the covariance function. Many options are available. According to Kriging approach, covariance is a function of distance between two points.
- For example: squared exponential form:

$$\mathrm{cov}(S_i, S_j; \theta) = \exp\left[-\left(\frac{S_i - S_j}{\theta_{ij}} \right)^2 \right]$$

How to compute the covariance of the derivative information

- First, the covariance function must support differentiable realizations. We will consider here only stationary covariance functions.

$$k(x,x') = k(x - x')$$

- The covariance function (of a stationary process) must be differentiable at 0 twice as many times as the realizations.
- E.g: The process is twice differentiable everywhere ➔ the covariance function must be four times differentiable at 0.
- For first-order derivative:

$$\operatorname{cov}\left(y, y'\right) = k(x,x') \quad \operatorname{cov}\left(\frac{\partial y}{\partial x_k}, y'\right) = \frac{\partial}{\partial x_k} k(x,x') \quad \operatorname{cov}\left(\frac{\partial y}{\partial x_k}, \frac{\partial y'}{\partial x_l'}\right) = \frac{\partial^2}{\partial x_k \partial x_l'} k(x,x').$$

Gaussian Processes approach, technical details:

- With the functional form of covariance specified, the hyperparameters θ are determined by maximizing the marginal likelihood function for the data. The logarithm of the likelihood is given by:

$$\log(p(J|S;\theta)) = -\frac{1}{2}Y^T K^{-1} Y + \frac{1}{2}Y^T K^{-1} H (H^T K^{-1} H)^{-1} H^T K Y - \frac{1}{2}\log|K| - \frac{m}{2}\log(2\pi)$$

- The optimization is carried out using standard tools (L-BFGS + active set algorithm).

- Computationally expensive parts of GP process: inverse of the covariance matrix, optimization problem, and very high resolution sampling. No part of the GP process scales at high resolution in current implementations, due to reliance on explicit, dense Cholesky.

- Both can be accelerated in future work:
  - Cholesky of covariance matrix, matrix- calculation of Q^0.5*N(0,I) (Chen, Anitescu, Saad, to appear in SISC) – not really attacked before.
  - Better optimization solver?

- But in current setup, expensive part is still sampling the code.

How do I choose the polynomials that represent the mean function?

- The basis spanning the mean function space has to have a reasonably small set of elements.
- Functional form: $R(x)a = \sum_{k=1,K} a_k \psi_k(x)$
- Here, $\{\psi_k(x)\}_{k=1}^{K}$ is a set of independent polynomials.
- How do I choose it? Oleg will tell you more about it.

How do I choose the covariance function?

- Squared Exponential:

$$k_i(x_i - x_i') = e^{-\left(\frac{x_i - x_i'}{\theta_i}\right)^2}$$

- Matern Function $\nu = \frac{3}{2}$:

$$k_i(x_i - x_i') = \left(1 + \sqrt{3}\left|\frac{x_i - x_i'}{\theta_i}\right|\right) e^{-\sqrt{3}\left|\frac{x_i - x_i'}{\theta_i}\right|}$$

- Matern Function $\nu = \frac{5}{2}$:

$$k_i(x_i - x_i') = \left(1 + \sqrt{5}\left|\frac{x_i - x_i'}{\theta_i}\right| + \frac{5}{3}\left|\frac{x_i - x_i'}{\theta_i}\right|^2\right) e^{-\sqrt{5}\left|\frac{x_i - x_i'}{\theta_i}\right|}$$

- Covariance functions must be "positive definite".
- The square exponential is one of the most used in machine learning, but also assumes the underlying process is very smooth, which may make the error estimate completely unreliable.
- The Matern function is one of the most robust, for the derivative-free case and it has controllable smoothness.

How do I choose the covariance function II.

- Cubic Spline 1:

$$k_i(x_i - x_i') = \begin{cases} 1 - 15\left|\frac{x_i - x_i'}{\theta_i}\right|^2 + 30\left|\frac{x_i - x_i'}{\theta_i}\right|^3 & \text{for } 0 \leq \left|\frac{x_i - x_i'}{\theta_i}\right| \leq 0.2 \\ 1.25\left(1 - \left|\frac{x_i - x_i'}{\theta_i}\right|\right)^3 & \text{for } 0.2 \leq \left|\frac{x_i - x_i'}{\theta_i}\right| \leq 1 \\ 0 & \text{for } \left|\frac{x_i - x_i'}{\theta_i}\right| \geq 1 \end{cases}$$

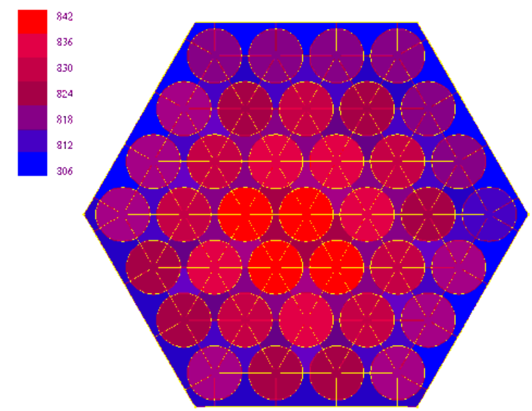
- Cubic Spline 2:

$$k_i(x_i - x_i') = \begin{cases} 1 - 6\left|\frac{x_i - x_i'}{\theta_i}\right|^2 + 6\left|\frac{x_i - x_i'}{\theta_i}\right|^3 & \text{for } 0 \leq \left|\frac{x_i - x_i'}{\theta_i}\right| \leq 0.5 \\ 2\left(1 - \left|\frac{x_i - x_i'}{\theta_i}\right|\right)^3 & \text{for } 0.5 \leq \left|\frac{x_i - x_i'}{\theta_i}\right| \leq 1 \\ 0 & \text{for } \left|\frac{x_i - x_i'}{\theta_i}\right| \geq 1 \end{cases}$$

- All the previous kernel functions result in DENSE matrices which may be a problem if I need to sample at many points.
- Cubic spline functions are examples of compact Kernels, with sparse covariance matrices that can be more easy to manipulate (e.g Cholesky, which is needed in max likelihood and sampling, may be doable)..

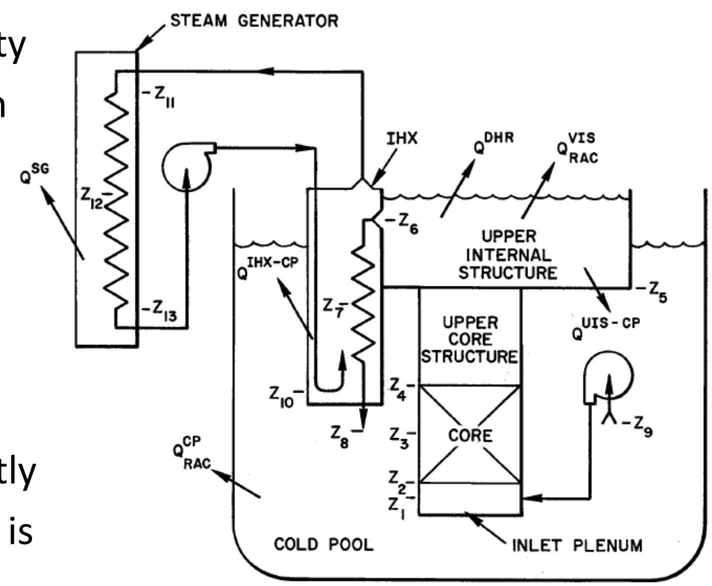
Uncertainty quantification, subject models

- Model I. Matlab prototype code: a steady-state 3-dimensional finite-volume model of the reactor core, taking into account heat transport and neutronic diffusion. Parameters with uncertainty are the material properties: heat conductivity, specific coolant heat, heat transfer coefficient, and neutronic parameters: fission, scattering, and absorbtion-removal cross-sections. Chemical non-homogenuity between fuel pins can be taken into account. Available experimental data is parameterized by 12-66 quantifiers.



- Model II. MATWS, a functional subset of an industrial complexity code SAS4A/SASSYS-1: point kinetics module with a representation of heat removal system. >10,000 lines of Fortran 77, sparsely documented.

MATWS was used, in combination with a simulation tool Goldsim, to model nuclear reactor accident scenarios. The typical analysis task is to find out if the uncertainty resulting from the error in estimation of neutronic reactivity feedback coefficients is sufficiently small for confidence in safe reactor temperatures. The uncertainty is described by 4-10 parameters.



Our working hypotheses re GEUK

- H1: GEUK results in less error compared with L_2 regression (GP with iid noise).
- H2: GEUK results in less error compared with universal Kriging without derivative information. Idea: one gradient evaluation brings d/5 more information.
- H3: GEUK results in less error for the same number of sample values when compared with ordinary Kriging.
- H4. GEUK approximates well the statistics of output, and its predicted covariance is a good or conservative estimate of the error}.
- H5. Covariance matters. It will affect the predictions and usability of the model. Idea: it is best to assume as little differentiability as one can get by with, particularly in the dense limit of samples.

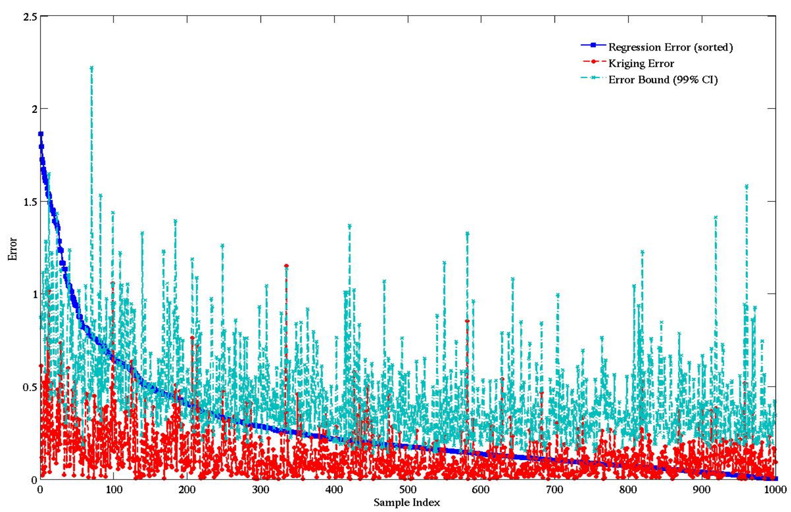
H1: Kriging versus Regression

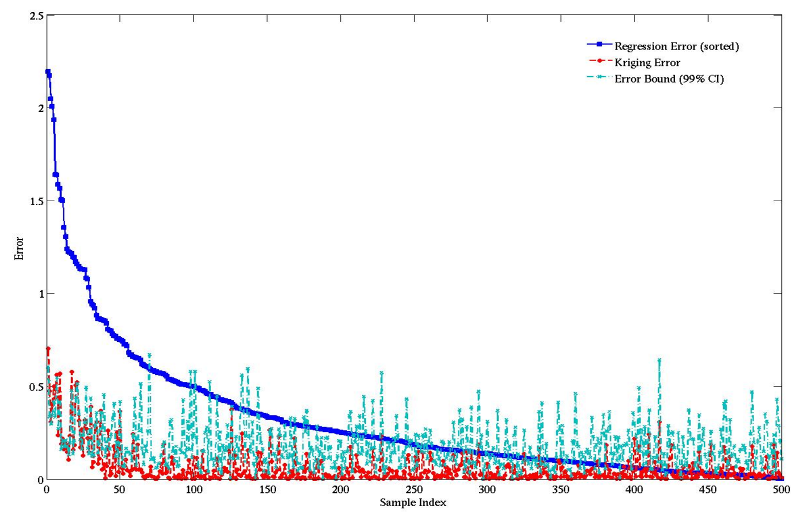
Table 3.9: MATWS – cubic spline 2 – comparison of error for Kriging and regression models

Sample Points	Kriging RMS	Regression RMS	Kriging Max	Regression Max
4 (p=2)	3.6433	15.2304	13.7491	56.6632
6 (p=2)	0.5260	3.2833	2.2040	14.0380
8 (p=3,trunc)	0.1841	0.5695	1.1980	3.1272
16	0.0766	0.427	0.747	2.404
24	0.0887	0.405	0.910	1.877
32	0.0995	0.309	1.118	1.959
40	0.0517	0.295	0.437	2.112
50	0.0508	0.251	0.386	1.476
100	0.0337	0.181	0.0998	1.068

Table 3.3: MATLAB – square exponential – Comparison of error for Kriging and regression models

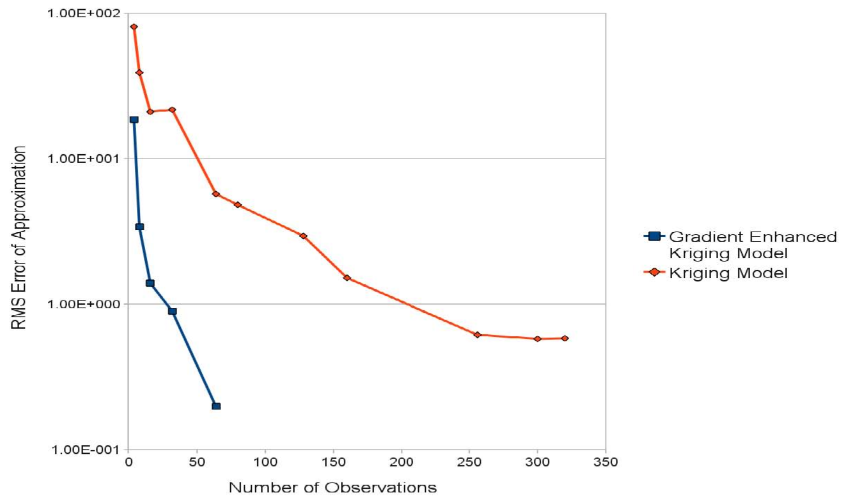
Data Set	Kriging RMS	Regression RMS	Kriging Max	Regression Max
1	0.11554	0.47118	0.70207	2.194
2	0.58351	0.76058	2.5731	3.2553
3	0.77163	1.1982	3.2202	4.8668
4	0.77163	1.289	3.2204	5.0067

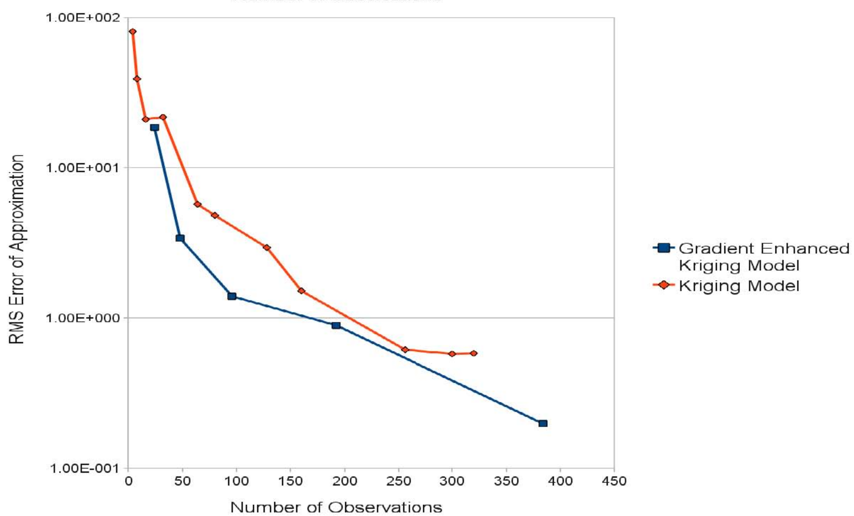




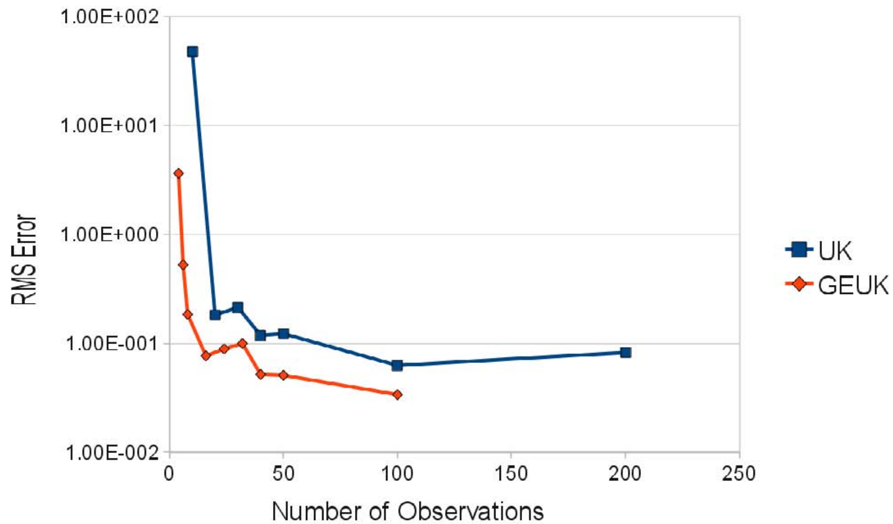
H2: Effect of gradient information

MATLAB

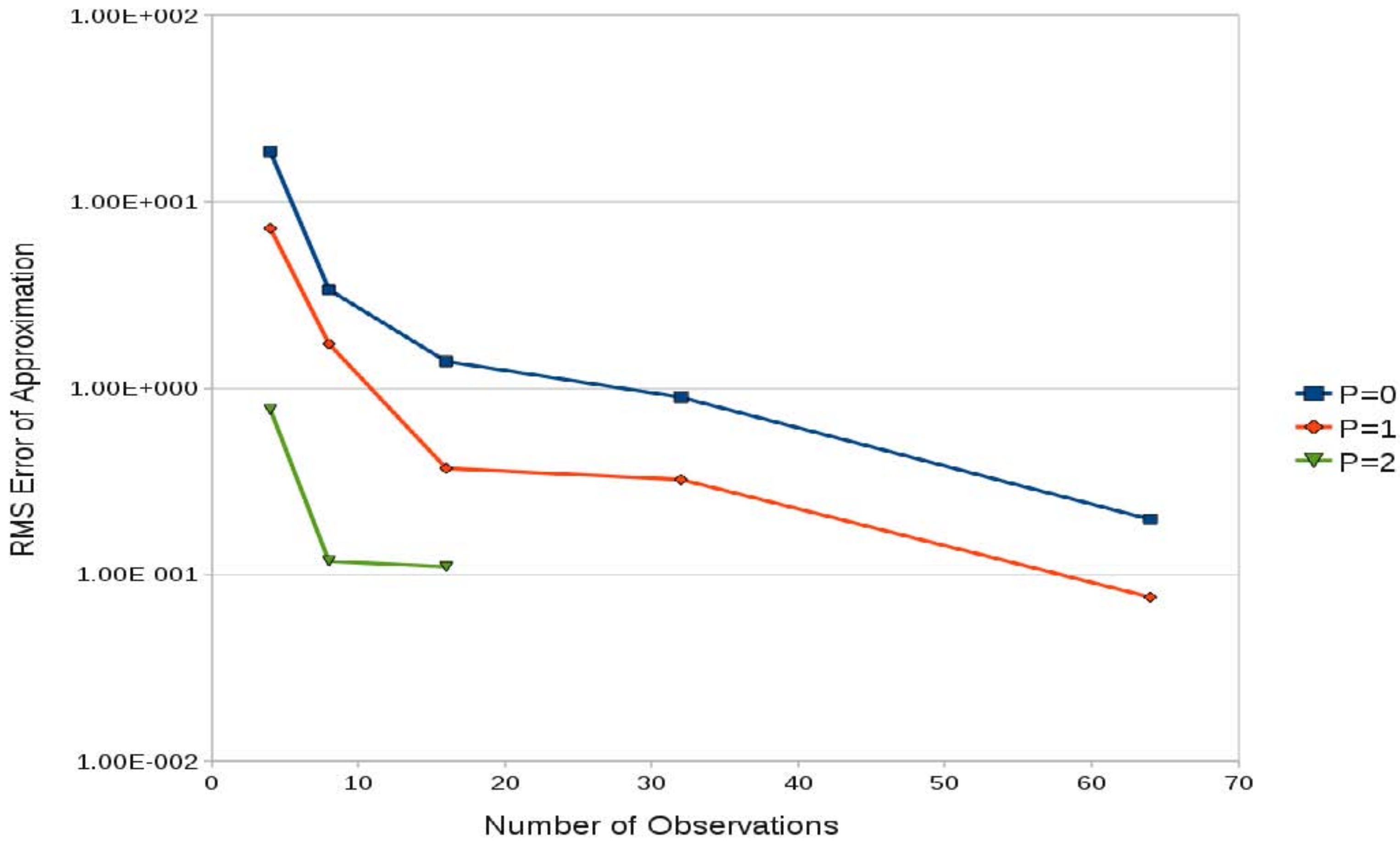




MATWS



H3: Using a mean function versus ordinary kriging



H4: Kriging gives me the a good approximation of the error

Table 3.5: MATLAB – square exponential – statistics for Kriging prediction

Data Set	$\pm 1\sigma$	$\pm 2\sigma$	$\pm 3\sigma$
1	0.690	0.882	0.950
2	0.378	0.568	0.668
3	0.362	0.626	0.720
4	0.362	0.626	0.720

Table 3.7: MATLAB – comparison of data distribution between covariance functions

Covariance Function	$\pm 1\sigma$	$\pm 2\sigma$	$\pm 3\sigma$
Cubic Spline 1	0.290	0.592	0.758
Cubic Spline 2	0.776	0.878	0.930
Squared Exponential	0.690	0.882	0.95
Matern-3/2	0.676	0.874	0.932
Matern-5/2	0.704	0.884	0.928

Table 3.16: MATWS – Matern-3/2 – Z scores for Kriging Prediction

Data Set	KS metric	$\pm 1\sigma^*$	$\pm 2\sigma^*$	$\pm 3\sigma^*$
4 (p=2)	0.5580	0.0030	0.0090	0.0150
6 (p=2)	0.2782	0.2510	0.4780	0.6030
8	0.1557	0.4640	0.7070	0.8130
16	0.0645	0.6150	0.8810	0.9510
24	0.0297	0.6890	0.9450	0.9840
32	0.0770	0.8200	0.9690	0.9840
40	0.0269	0.7150	0.9590	0.9900
50	0.0601	0.7890	0.9870	1.0000

Table 3.12: MATWS – cubic spline 2 – statistics for kriging prediction

Data Set	$\pm 1\sigma$	$\pm 2\sigma$	$\pm 3\sigma$
4 (p=2)	0.847	0.977	0.999
6 (p=2)	0.513	0.964	1.000
8 (p=3,trunc)	0.599	0.968	1.000
16	0.697	0.942	1.000
24	0.533	0.857	0.971
32	0.525	0.817	0.942
40	0.475	0.800	0.937
50	0.366	0.685	0.870
100	0.221	0.424	0.600

H5: The choice of covariance function matters

Table 3.10: MATWS – comparison of covariance functions for 8 pt GEUK model

Covariance Function	RMS Error	Max Error
Cubic Spline 1	0.2083	1.3567
Cubic Spline 2	0.1841	1.1980
Squared Exponential	0.1245	1.0125
Matern-3/2	0.1704	1.0645
Matern-5/2	0.1530	1.0566

Table 3.8: MATLAB – square exponential – statistics for Kriging prediction using decc

Covariance Function	KS Metric	$\pm 1\sigma^*$	$\pm 2\sigma^*$	$\pm 3\sigma^*$
Cubic Spline 1	0.2939	0.3080	0.4520	0.5960
Cubic Spline 2	0.2766	0.2820	0.3800	0.6540
Squared Exponential	0.1412	0.4880	0.7220	0.8380
Matern-3/2	0.1974	0.4200	0.6820	0.8320
Matern-5/2	0.2383	0.3300	0.5700	0.7160

Table 3.11: MATWS – comparison of covariance functions for 50 pt GEUK model

Covariance Function	RMS Error	Max Error
Cubic Spline 1	0.0567	0.3963
Cubic Spline 2	0.0528	0.4551
Squared Exponential	0.1487	2.0268
Matern-3/2	0.0398	0.2552
Matern-5/2	0.0749	0.7991

As in the derivative-free case, Matern 3/2 seems a good choice

Quantile Estimation

- This is a critical statistic in nuclear engineering.
- Particularly, the 95% statistics with 95 % confidence.
- "Conservative Estimate" using the Uniform distribution and properties of order statistics and uniform distributions quantiles.

Table 4.1: MATLAB – cubic spline 2 – quantile calculation for MATLAB data

Sample Points	Regression Order	Kriging Estimate	Regression Estimate	Training Estimate
4	2	2446.6	2447.2	2323.8
6	2	2448.2	2447.5	2335.2
8	3	2449.1	2448.4	2360.4

Actual Value = 2456.0

Table 4.2: MATWS – cubic spline 2 – quantile calculation for MATWS data

Sample Points	Kriging Estimate	Regression Estimate	Training Estimate
4 (p=2)	865.73	864.78	863.55
6 (p=2)	865.86	871.15	863.55
8	866.08	866.60	863.46
16	865.89	866.51	865.45
24	865.83	866.49	865.56
32	865.87	866.32	865.76
40	865.82	866.37	865.86
50	865.83	866.42	865.86

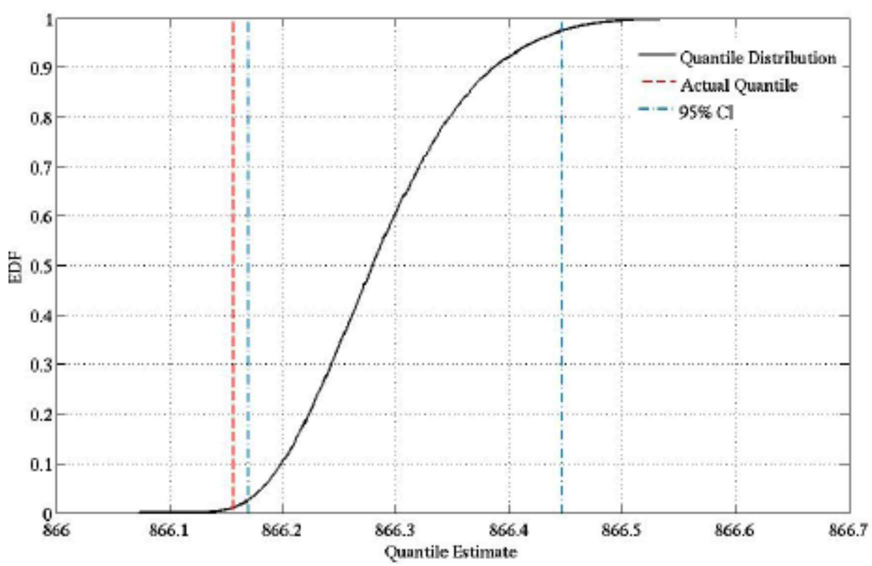
Actual Value = 866.16

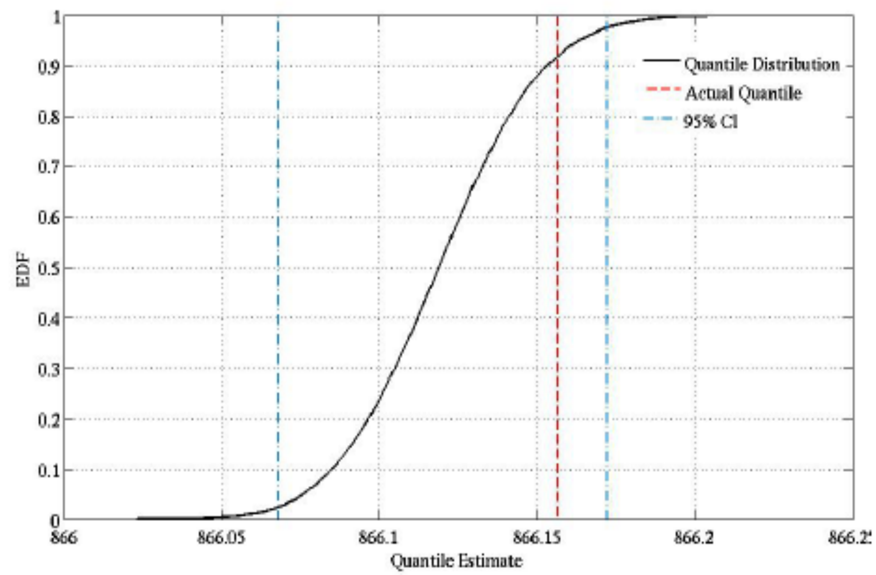
Estimating Quantiles Using Asymptotic tests

Table 4.5: Quantile – Matern-3/2 – Example results for Kriging quantile estimate with confidence interval

DATA	Training Points	Confidence(1-α)	Lower Bound	Upper Bound	Median
MATWS	8	0.95	866.17	866.45	866.28
MATWS	8	0.99	866.17	866.45	866.28
MATWS	50	0.95	866.07	866.17	866.12
MATLAB	8	0.95	2454.3	2455.6	2445.0
MATLAB	8	0.99	2454.2	2.455.8	2455.0

MATWS: 8 samples and 50 samples





Conclusions

- Gradient-enhanced universal kriging brings combines the best advantages in sensitivity, regression, and Gaussian processes.
- It can provide good statistics for nuclear engineering codes with 6-8 samples for the limited examples we tried.
- More accurate that regression, more efficient than kriging.

- Future:
  - Larger number of parameters
  - How to create the basis function for regression
  - Approximated the gradients of very large scale codes.]