LA-UR-12-22020

Approved for public release; distribution is unlimited.

Title:	Computationally Efficient Use of Derivatives in Emulation of Complex Computational Models
Author(s):	Williams, Brian J. Marcy, Peter W.
Intended for:	IMS/ASA Spring Research Conference 2012, 2012-06-13/2012-06-15 (Cambridge, Massachusetts, United States)

Computationally Efficient Use of Derivatives in Emulation of Complex Computational Models

Brian Williams and Peter Marcy

LA-UR-12-

Statistical Sciences Group

Los Alamos National Laboratory

Abstract

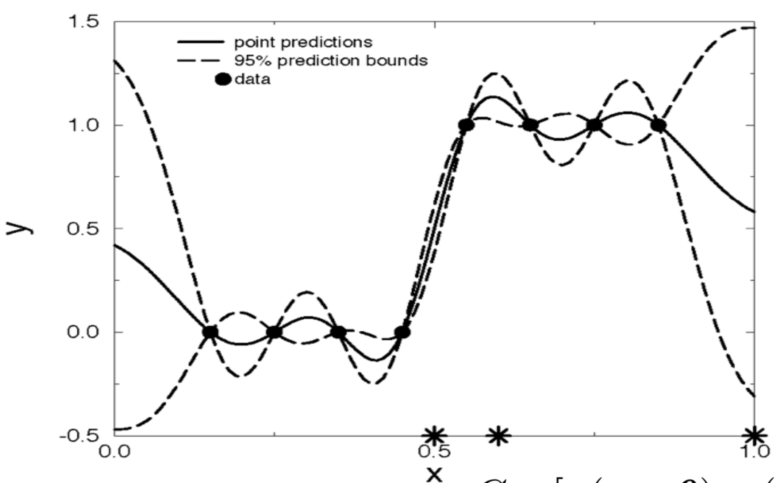
We will investigate the use of derivative information in complex computer model emulation when the correlation function is of the compactly supported Bohman class. To this end, a Gaussian process model similar to that used by Kaufman et al. (2011) is extended to a situation where first partial derivatives in each dimension are calculated at each input site (i.e. using gradients). A simulation study in the ten-dimensional case is conducted to assess the utility of the Bohman correlation function against strictly positive correlation functions when a high degree of sparsity is induced.

Outline

- Emulation of computational models
- Incorporation of derivative information
- Compactly supported covariance functions
- Simulation Study
- Results
- Conclusions

Emulator

- Use training runs to develop a statistical surrogate model for the complex code (i.e., the *emulator*)
  - Deterministic code is interpolated with zero uncertainty
- Kriging Predictor $\hat{\eta}(\boldsymbol{x};\boldsymbol{\theta}) = \boldsymbol{f}'(\boldsymbol{x})\hat{\boldsymbol{\beta}} + \boldsymbol{r}'(\boldsymbol{x};\boldsymbol{\theta})\boldsymbol{R}^{-1}(\boldsymbol{\theta})(\boldsymbol{\eta} - \boldsymbol{F}\hat{\boldsymbol{\beta}})$



correlations between prediction site **x** and training runs $\mathbf{x}_1,\ldots,\mathbf{x}_m$

outputs evaluated at training runs $\mathbf{x}_1,\ldots,\mathbf{x}_m$

Regression functions evaluated at prediction site **x** and parameters

pairwise correlations between training runs $\mathbf{x}_1,\ldots,\mathbf{x}_m$

- Kriging Variance

$$Cov[\eta(\boldsymbol{x}_1,\boldsymbol{\theta}),\eta(\boldsymbol{x}_2,\boldsymbol{\theta})|\boldsymbol{\eta}] = \sigma^2 \left(R(\boldsymbol{x}_1,\boldsymbol{x}_2;\boldsymbol{\theta}) - \boldsymbol{r}'(\boldsymbol{x}_1,\boldsymbol{\theta})\boldsymbol{R}^{-1}(\boldsymbol{\theta})\boldsymbol{r}(\boldsymbol{x}_2,\boldsymbol{\theta}) + \boldsymbol{h}'(\boldsymbol{x}_1,\boldsymbol{\theta})(\boldsymbol{F}'\boldsymbol{R}^{-1}(\boldsymbol{\theta})\boldsymbol{F})^{-1}\boldsymbol{h}(\boldsymbol{x}_2;\boldsymbol{\theta})\right)$$

$$\boldsymbol{h}(\boldsymbol{x};\boldsymbol{\theta}) = \boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{F}'\boldsymbol{R}^{-1}(\boldsymbol{\theta})\boldsymbol{r}(\boldsymbol{x};\boldsymbol{\theta})$$

  - Full Bayes inference for σ^2, θ

Los Alamos
NATIONAL LABORATORY
EST. 1943

Incorporating Derivatives

- Morris, Mitchell and Ylvisaker (*Technometrics*, 1993)

$$\begin{pmatrix} \eta(\boldsymbol{t}) \\ \frac{\partial \eta(\boldsymbol{t})}{\partial t_i} \end{pmatrix} \sim GP\,, \quad \text{mean} \begin{pmatrix} \mu(\boldsymbol{t}) \\ \frac{\partial \mu(\boldsymbol{t})}{\partial t_i} \end{pmatrix} ; \quad \text{covariance} \begin{pmatrix} C(\boldsymbol{t},\boldsymbol{s}) & \frac{\partial C(\boldsymbol{t},\boldsymbol{s})}{\partial s_i} \\ \frac{\partial C(\boldsymbol{t},\boldsymbol{s})}{\partial t_i} & \frac{\partial^2 C(\boldsymbol{t},\boldsymbol{s})}{\partial t_i \partial s_i} \end{pmatrix}$$

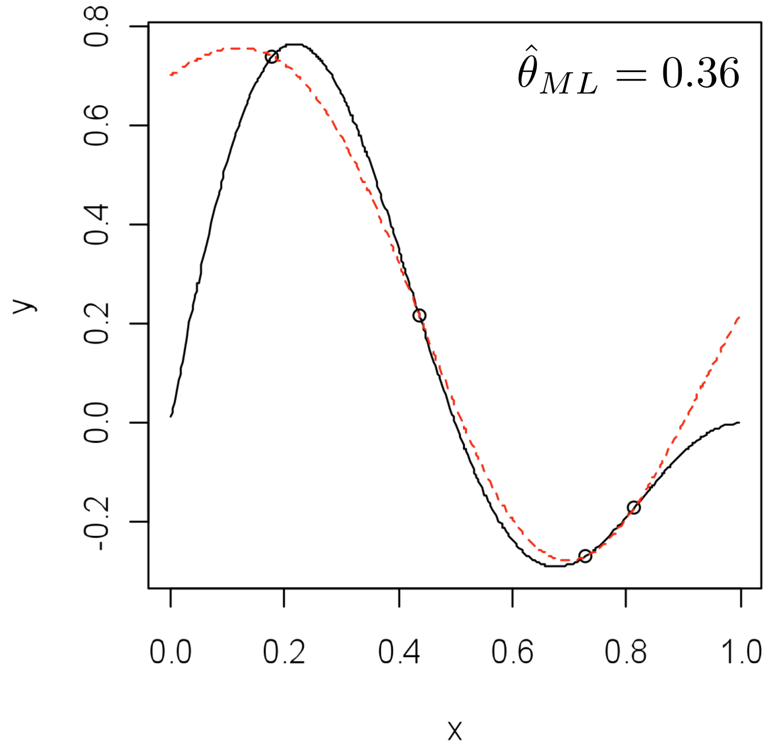
- Computational efficiency eventually required

Size of Covariance Matrix

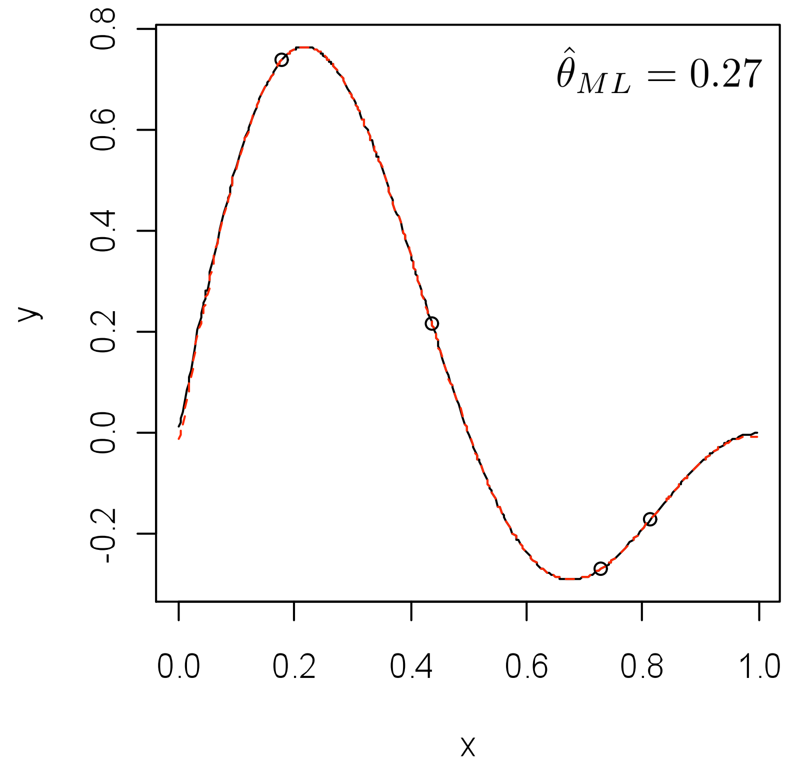
Sample Size	Input Dimension (D)				
	1	5	10	20	50
1.5D	3	45	165	630	3825
5D	10	150	550	2100	12750
10D	20	300	1100	4200	25500

Small Sample Sizes Enhanced With Derivatives

Gaussian Correlation
No Derivatives



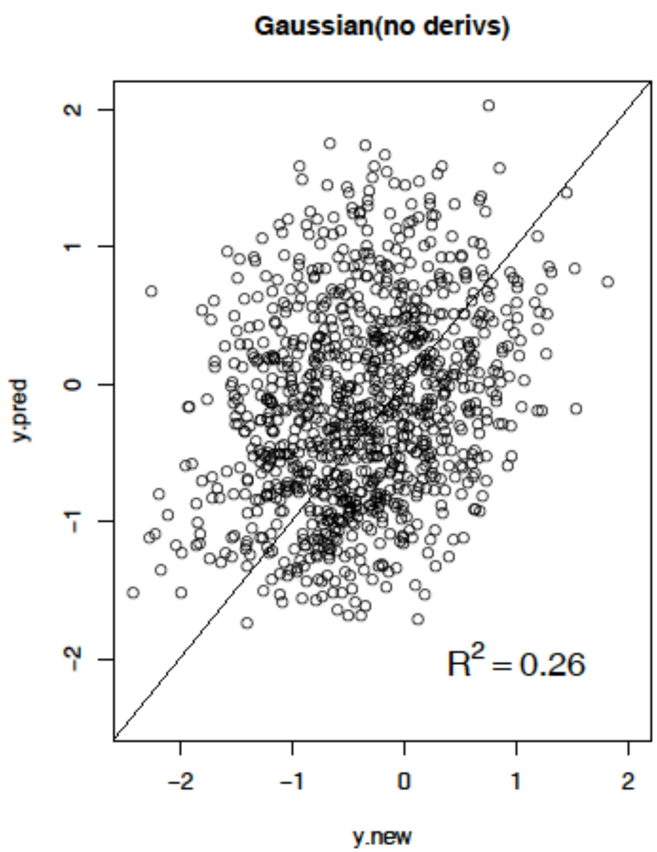
Gaussian Correlation
First Derivatives

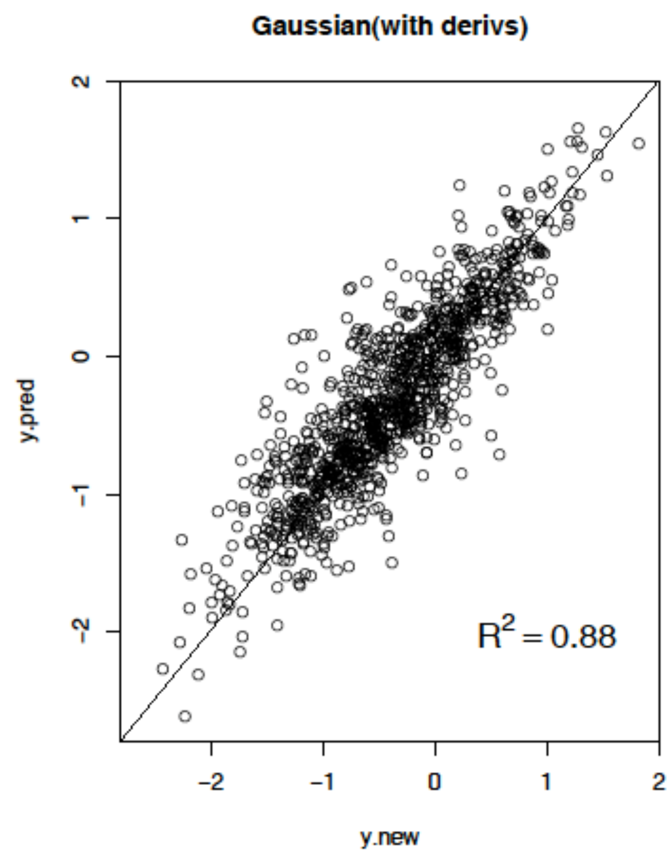


$$R(x_1, x_2) = \exp\left\{-[|x_1 - x_2|/\theta]^2\right\}$$

Benefits of Derivatives in Higher Dimensions

- D = 8, N = 2D = 16
- True function generated from GP with Gaussian correlation





Compactly Supported Covariance

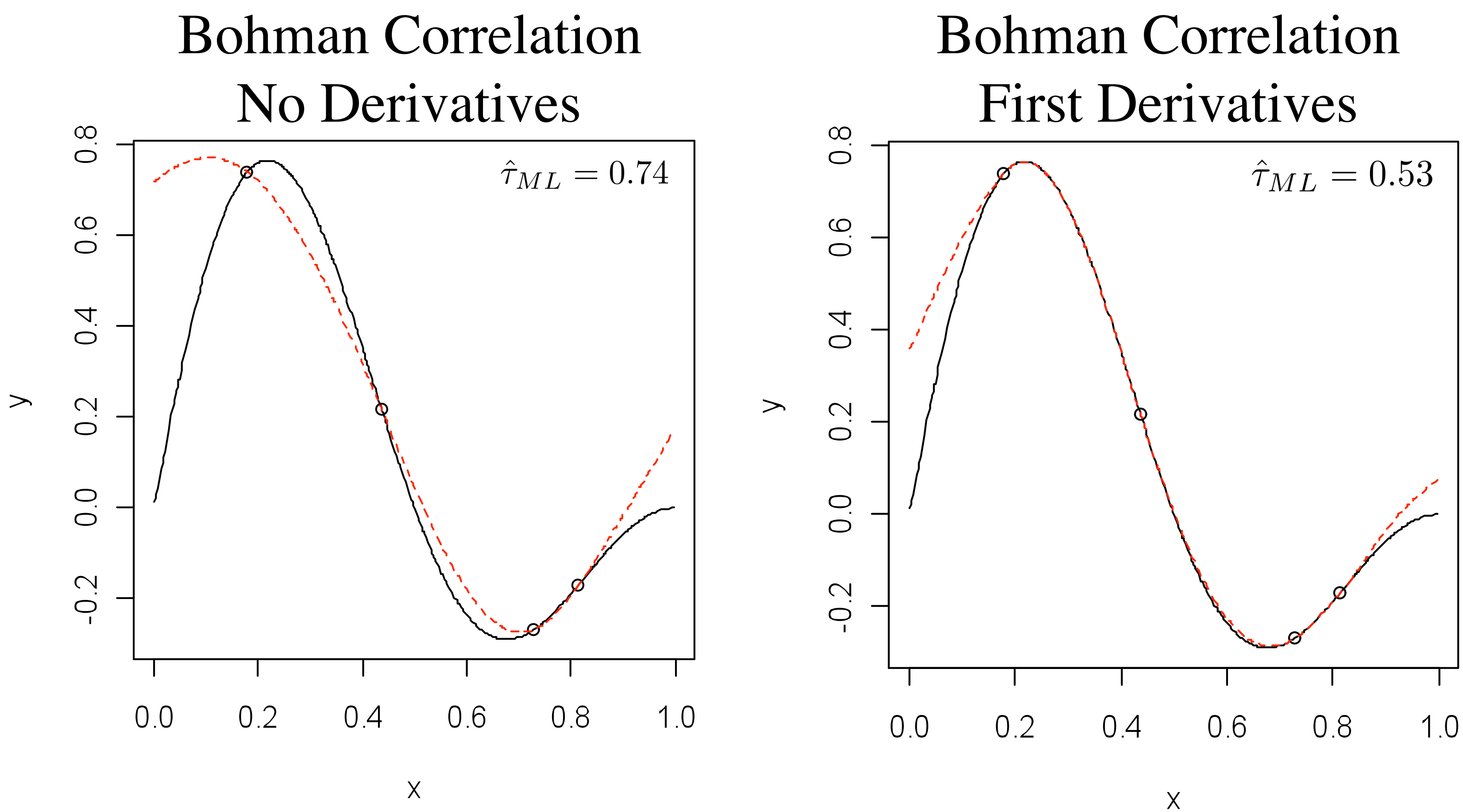
- Kaufman, *et al.* (2011)
  - 20,000 simulator evaluations
  - Use low order regression models and compactly supported covariance function to achieve a good fit with sparse covariance matrix
  - Increased sparsity resulted in less prediction efficiency, especially for more difficult functions, but improvement with sample size
  - Sparsity resulted in improved coverage properties but gains declined with sample size

- Bohman correlation function

$$R(t;\tau) = \begin{cases} (1 - t/\tau)\cos(\pi t/\tau) + \sin(\pi t/\tau)/\pi\,, & t < \tau \\ 0\,, & t \geq \tau \end{cases}$$

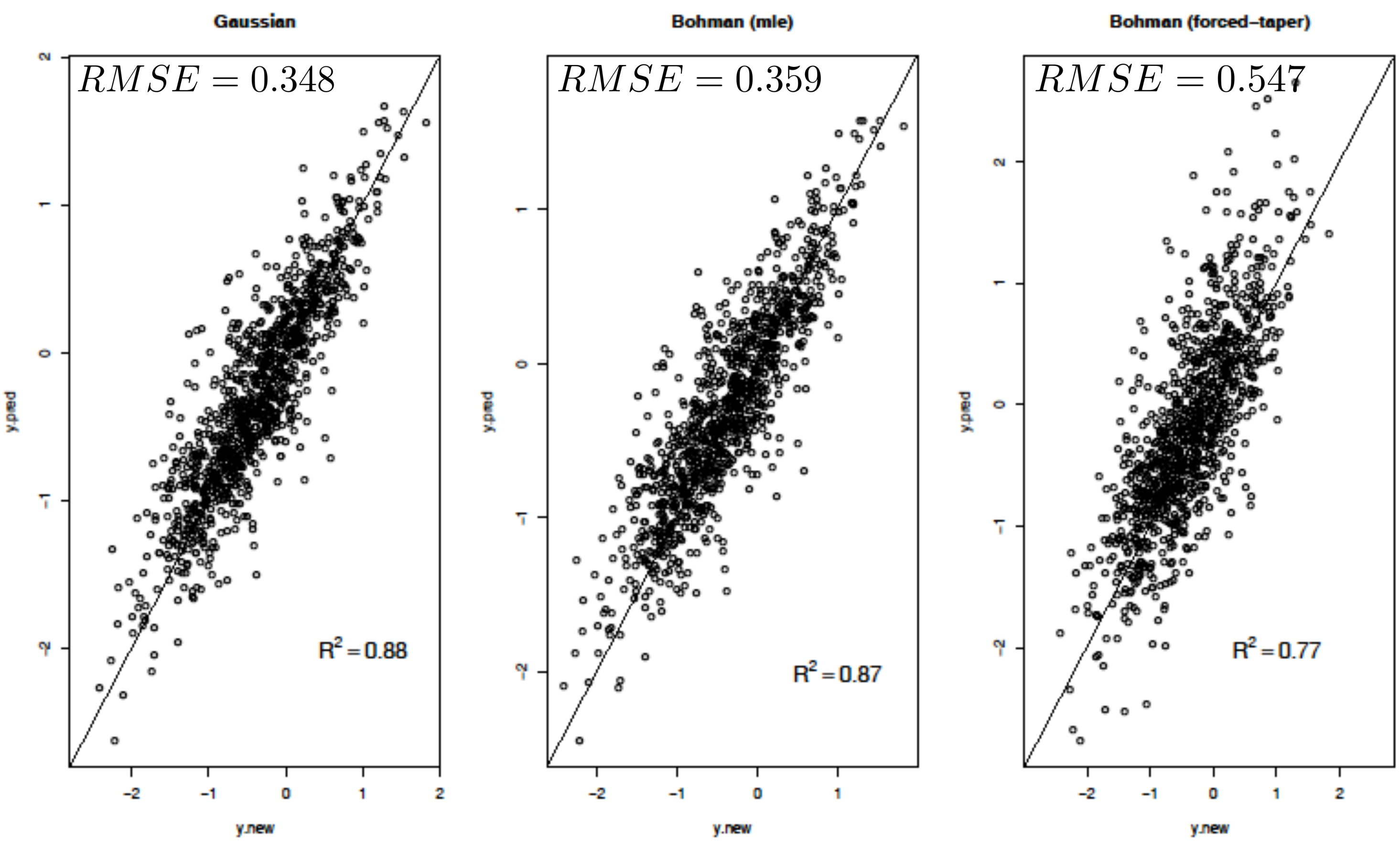
  - Prior restriction: $\boldsymbol{\tau} \in \mathcal{R}^D \;:\; \tau_j \geq 0 \text{ for all } D\,, \sum_{j=1}^{D} \tau_j \leq C\,,\, C > 0$

Advantage of Derivatives Persists with Sparsity



Benefits of Derivatives Even With High Sparsity

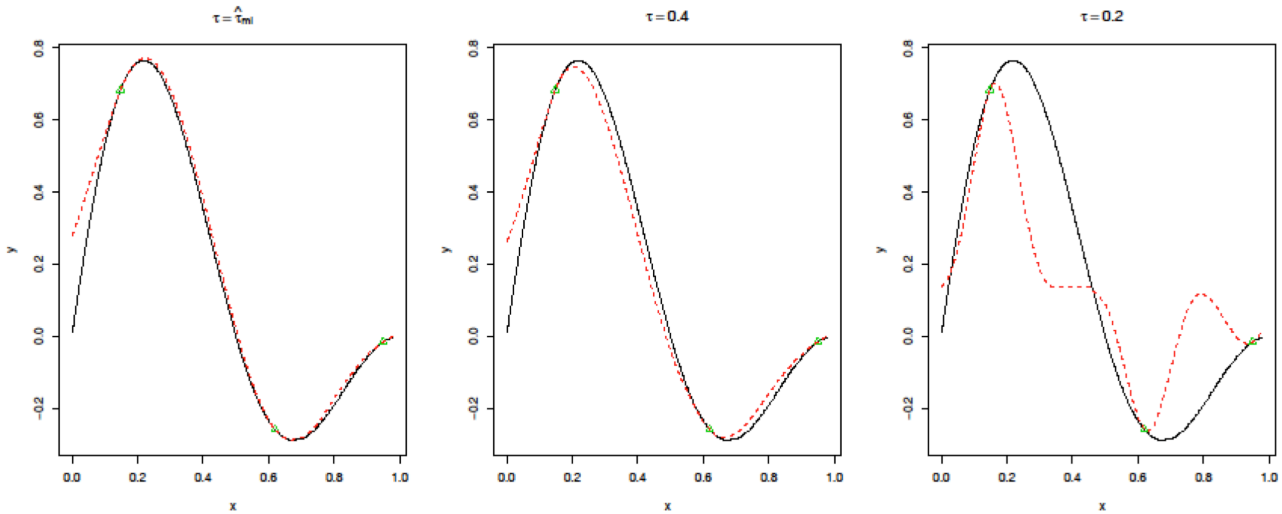
- D = 8, N = 2D = 16
- True function generated from GP with Gaussian correlation



90% sparsity

Inclusion of Regression Model

- Regression often necessary to allow for covariance sparsity



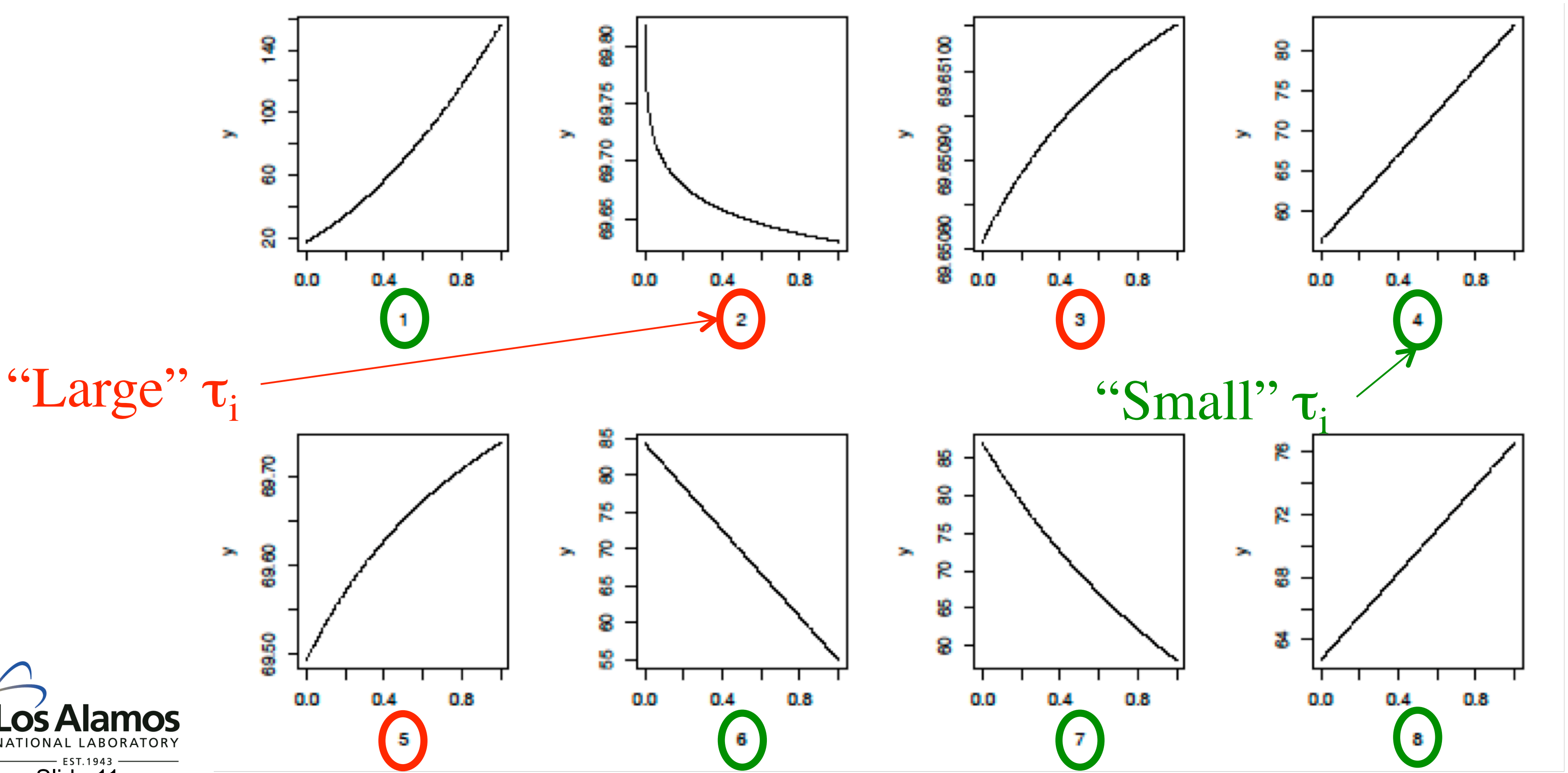
- Advantages are seen in higher dimensional input spaces

RMS Errors
D = 8

Degree of Regression Model	Unconstrained Bohman	Constrained Bohman (~90% sparsity)
0	0.3591	0.7464
1	0.3668	0.9553
2	0.4548	0.5474
3	0.4766	0.5611

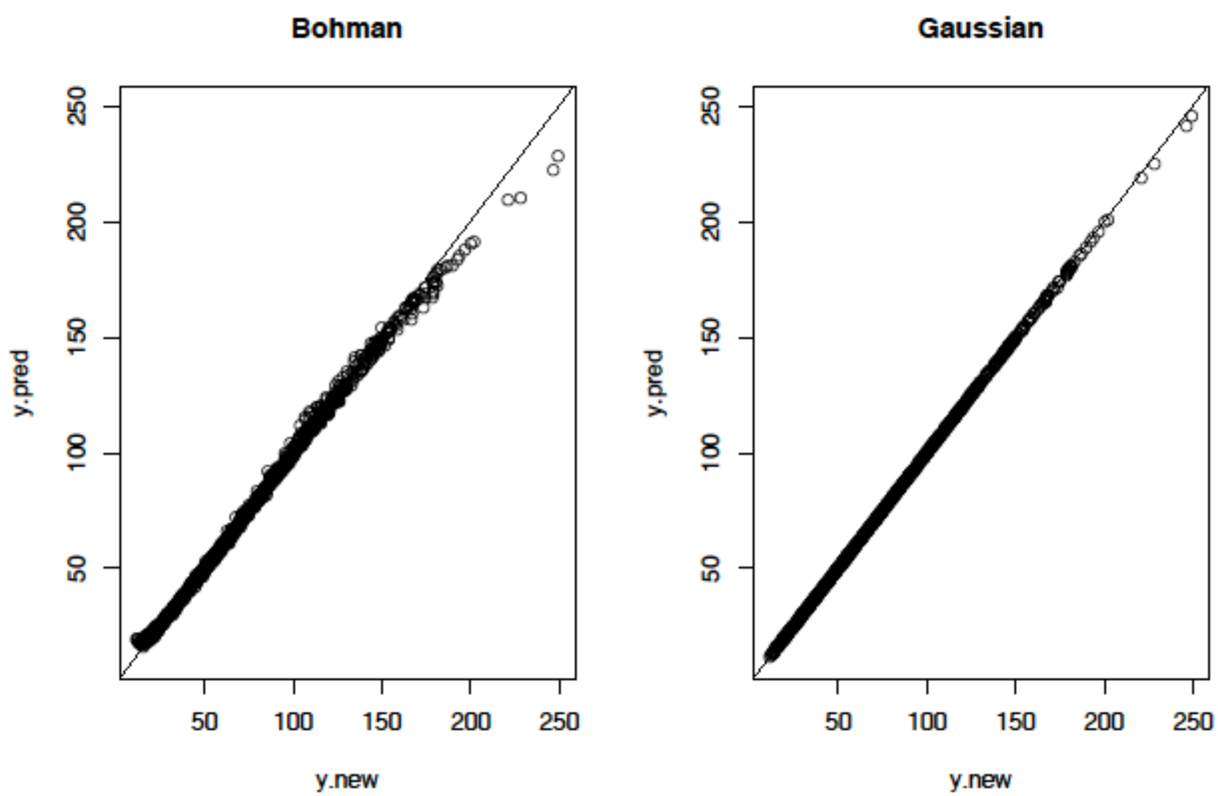
Regression Helps Induce Sparsity

- Borehole: Morris (1993) with inputs scaled to unit cube, $[0,1]^8$
- Sparsity constraint results in larger correlation length τ in dimensions not modeled well by regression



Sparsity and Predictive Capability

- Sparsity is pushed towards 99% but still produces an R^2 of 0.999
  - Dense correlation matrices (e.g. from Gaussian or Matern) may be ill-conditioned in this situation



Sparsity and Computing

- Sparsity depends on tapering parameters and design (which depends on dimension, sample size)
  - We used maximin Latin hypercube designs
- Computation time depends on sparse matrix algorithms used
  - In R 'spam' uses Yale sparse format
- Times using 'spam' in R are below
  - Predicting 1000 new values
  - Algorithms are even faster above ~95% sparsity

N = 2D	Sparse	Dense	N = 3.5D	Sparse	Dense
D = 10	0.05 / 1.34	0.06 / 2.17		0.06 / 1.52	0.11 / 2.24
D = 20	0.24 / 3.31	0.58 / 5.31		0.55 / 4.51	1.85 / 8.65
D = 30	1.69 / 9.09	4.45 / 15.4		6.17 / 18.5	16.1 / 36.2

Likelihood evaluation

Prediction

Simulation Study

- Generate a GP with specified covariance structure (Gaussian assumed here) and get response values and partial derivative information
- Two input dimensions (D = 10 and D = 20)
- Two complexity levels ("Full" and "KL group 3")
- Seven modeling choices:
  - True correlation model (Gaussian)
  - Bohman at two levels of sparsity (~90% and ~95%) and four levels of regression modeling (none, linear, quadratic, quadratic + some cubic terms)
- Five replications of each factor-level combination

KL Expansion

- Decomposition of process into eigenvalues and eigenfunctions

$$\eta(\boldsymbol{x},\omega) = \mu(\boldsymbol{x}) + \sum_{j=1}^{\infty} \sqrt{\lambda_j}\,\phi_j(\boldsymbol{x})\,Z_j(\omega)$$

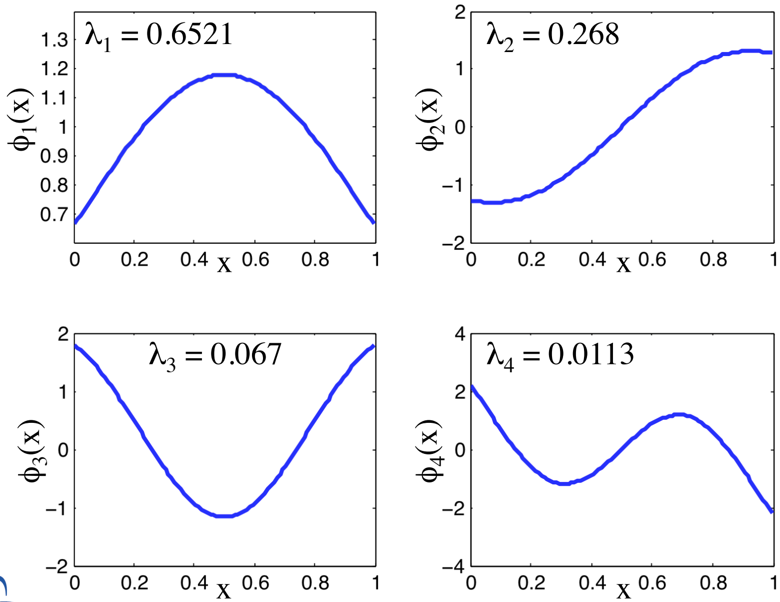
N(0,1)

$$\int_D C(\boldsymbol{u},\boldsymbol{v})\,\phi(\boldsymbol{v})\,d\boldsymbol{v} = \lambda\,\phi(\boldsymbol{u})$$

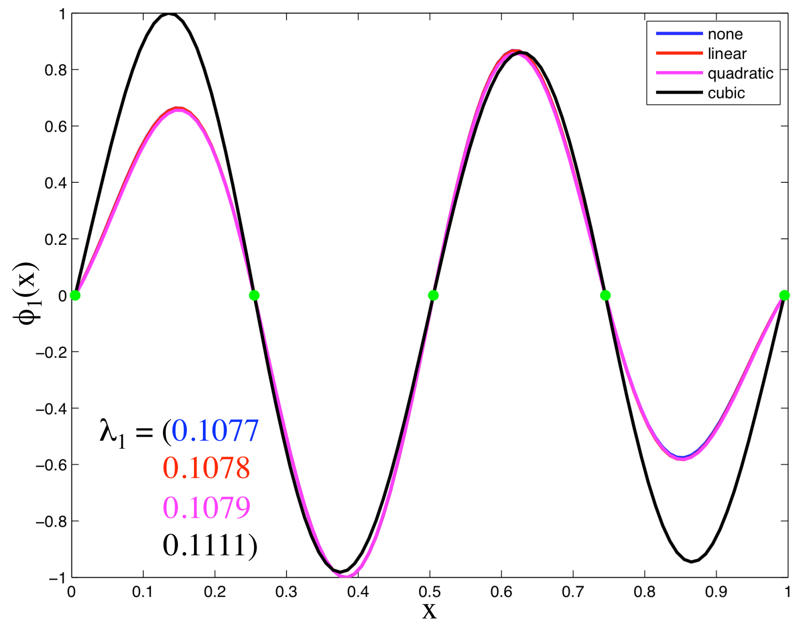
Fredholm integral equation of the second kind

- Solutions via Galerkin approximation (low-D)

Gaussian correlation



Bohman correlation



GP Simulation

- Simulate from zero-mean Gaussian process with product Gaussian correlation function
  - Complexity and sparsity specified (Loeppky, *et al.* (2010))
  - Output and derivatives desired
- Use 1-d KL expansion
  - Decomposition by dimension

eigenvalues	eigenfunctions
$\lambda_{1,1}, \ldots, \lambda_{1,m_1}$	$\phi_{1,1}(x_1), \ldots, \phi_{1,m_1}(x_1)$
$\vdots$	$\vdots$
$\lambda_{D,1}, \ldots, \lambda_{D,m_D}$	$\phi_{D,1}(x_D), \ldots, \phi_{D,m_D}(x_D)$

  - Eigenfunctions are tensor products of 1-d eigenfunctions

Group	Eigenfunction	Terms
first	$\prod_{j=1}^{D} \phi_{j,1}(x_j)$	1
second	replace $\phi_{i,1}(x_i)$ with $\phi_{i,2}(x_i)$	D
third	replace $\phi_{i_1,1}(x_{i_1})$ and $\phi_{i_2,1}(x_{i_2})$ with $\phi_{i_1,2}(x_{i_1})$ and $\phi_{i_2,2}(x_{i_2})$	$\binom{D}{2}$

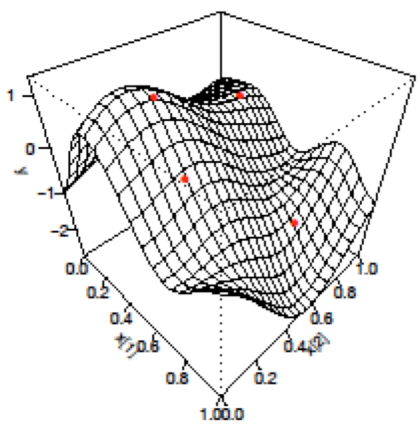
  - Partial derivatives are easily calculated

$$\text{replace } \phi_{i,j}(x_i) \text{ with } \frac{\partial \phi_{i,j}(x_i)}{\partial x_i}$$

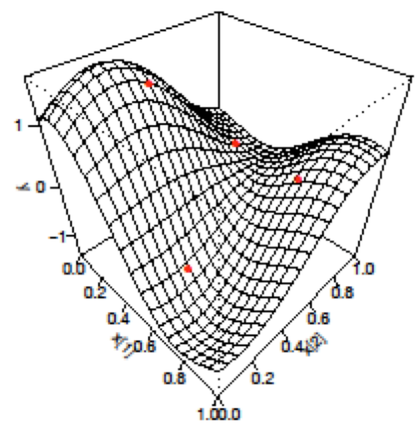
Response Surfaces for Simulation Study

- D = 10 dimensional input space
- More complex functions obtained with use of more second order eigenfunctions

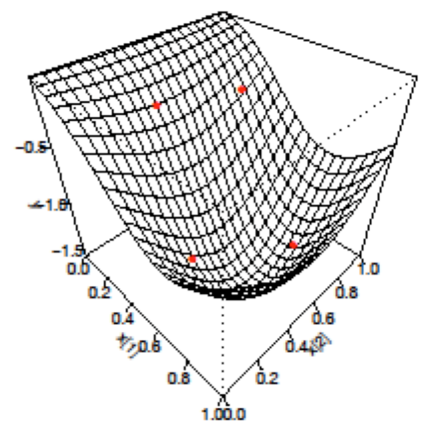
"Full"
Replace 3 $\phi_{.,1}$ with $\phi_{.,2}$



"Group 3"
Replace 2 $\phi_{.,1}$ with $\phi_{.,2}$



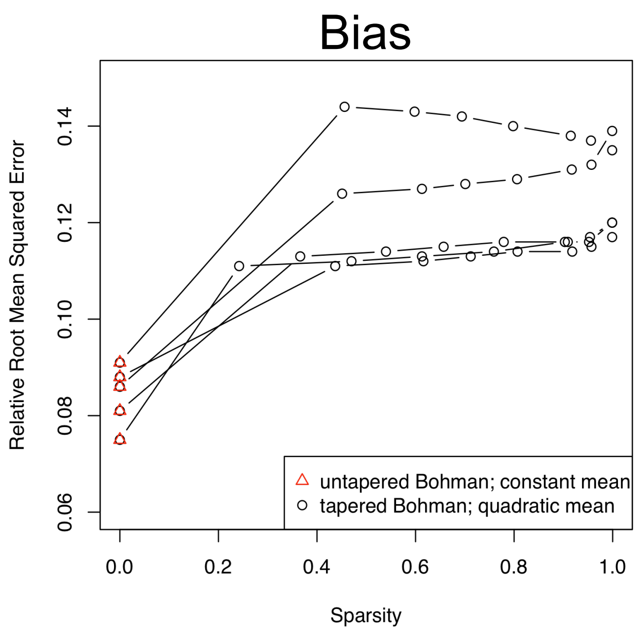
Replace 1 $\phi_{.,1}$ with $\phi_{.,2}$

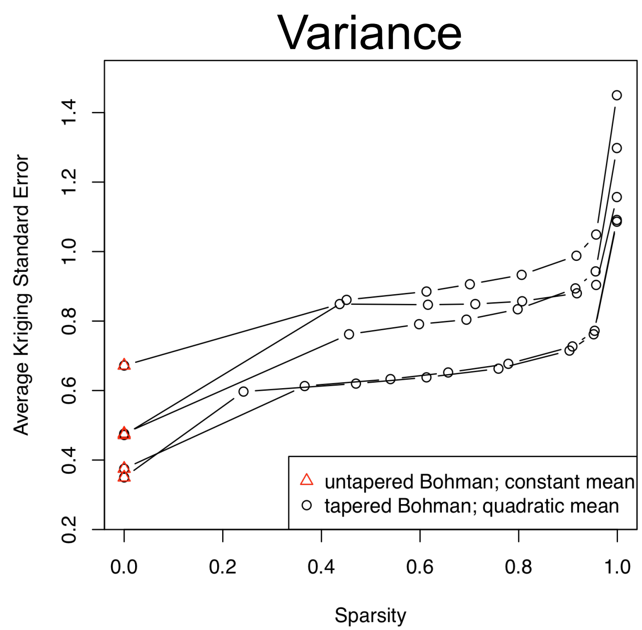


Los Alamos
NATIONAL LABORATORY
EST. 1943

Results: Effects of Sparsity

Surfaces drawn from GP with Gaussian correlation function
(D = 10; N = 20)

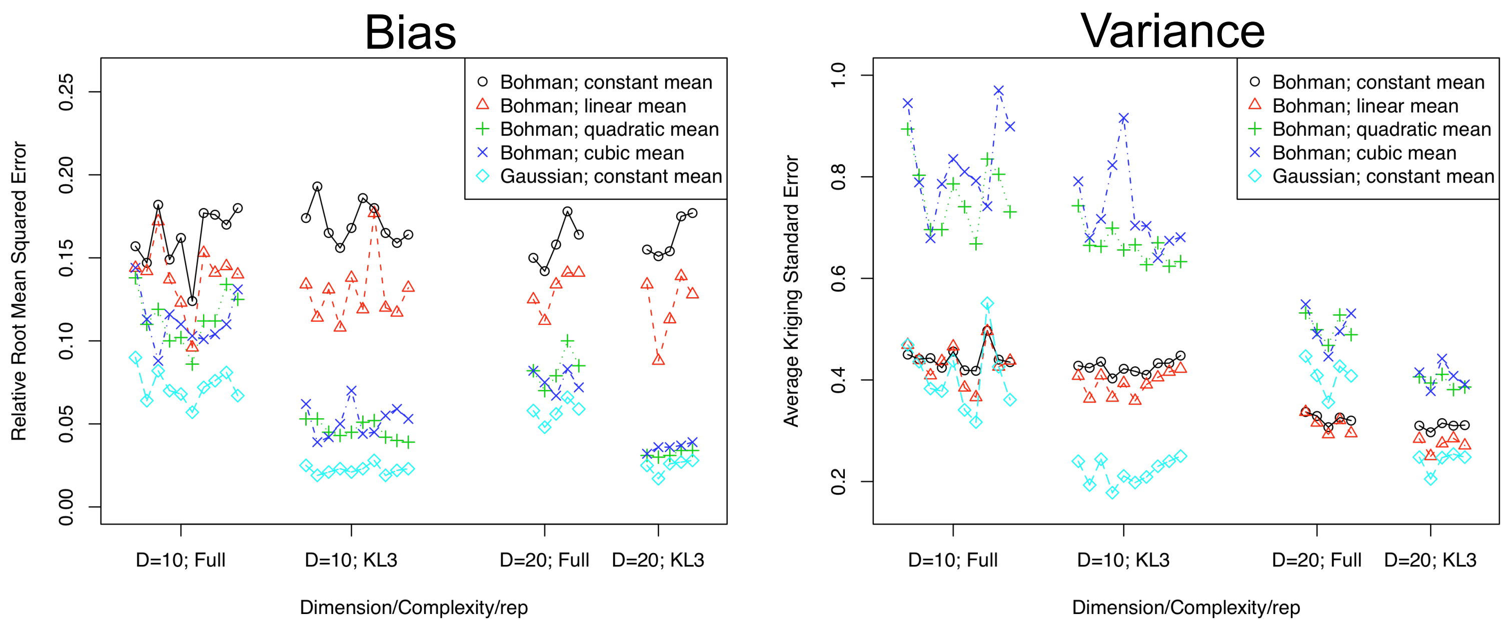




- Results using just model outputs (ignoring first partials)
  - ✧ untapered Bohman, constant mean
  - ✧ Relative RMSE: 0.165 (> RMSE with derivatives at all sparsity levels)
  - ✧ ARMPSE: 0.75 (comparable to avg. ARMPSE with derivatives at ~95% sparsity)

Results: Accuracy and Precision of Emulators

Surfaces drawn from GP with Gaussian correlation function
Full and KL3 complexity
(D = 10; N = 20 and D=20; N=40)
Bohman correlation tapered to 90+% sparsity

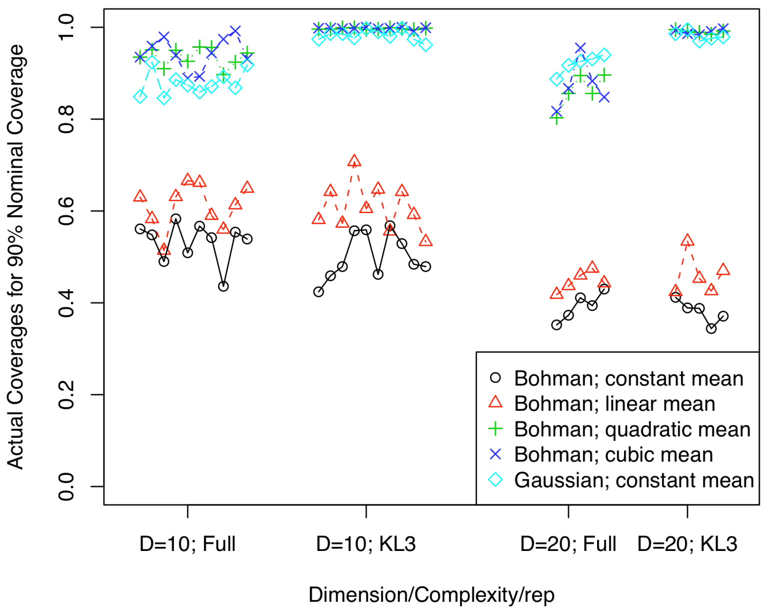


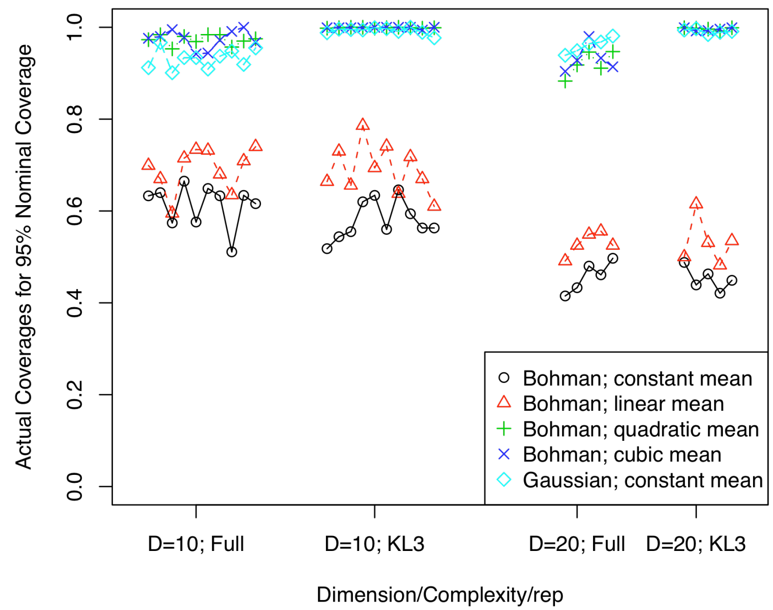
- RMSE values lower for higher order regression models with tapering
- ARMPSE values higher for higher order regression models with tapering

Los Alamos
NATIONAL LABORATORY
EST. 1943

Results: Frequentist Coverage

Surfaces drawn from GP with Gaussian correlation function
Full and KL3 complexity
(D = 10; N = 20 and D=20; N=40)
Bohman correlation tapered to 90+% sparsity





- Actual coverages low for inadequate regression models with tapering
- Higher kriging SE required with tapering versus true correlation model to achieve nominal coverage

Compromise

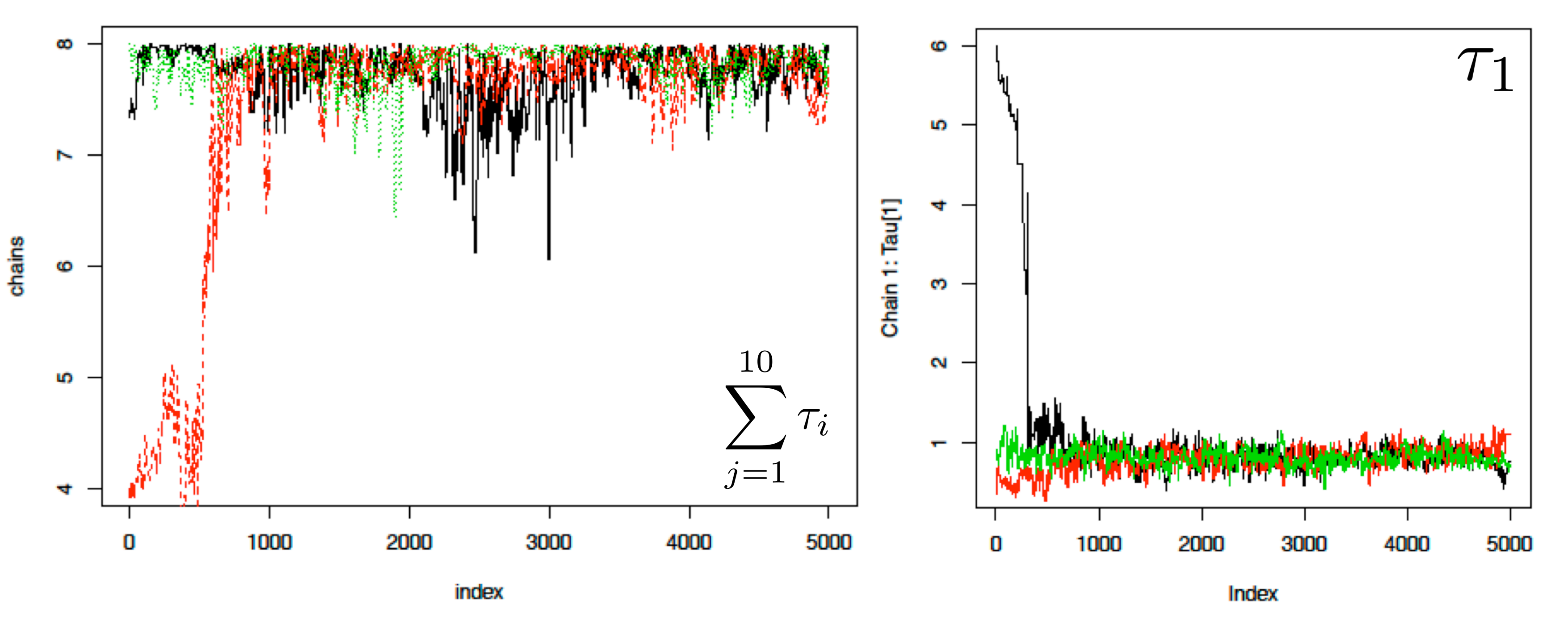
- Trade-off between computation time and predictive capability
  - Sparsity restriction complicates the ability to model deviations from smooth global trend and can lead to increased prediction variances and, to a lesser extent (depending on regression model), increased prediction biases relative to "ideal" emulator
- In the extreme case of ~100% sparsity, the surrogate is essentially regression with derivatives, but the prediction intervals become useless
- In the case of <100% sparsity, the full Bayesian approach (with informative prior) can be used to get improved prediction intervals relative to use of plug-in parameter estimates

MCMC: Metropolis Within Gibbs

- Gibbs updates of regression parameters
- Metropolis sampling of τ vector
  - Optimal τ vector may lie on a boundary of the simplex defined by the sparsity constraint
- Start at the center of the simplex: $C/2 = \sum_{j=1}^{D} \tau_j/2$
  - Multivariate normal proposal distribution
  - weak proposal covariance matrix
- Every 100 iterations tune covariance to the sample covariance of the parameters
  - Do this 50-100 times
  - Multiply covariance matrix obtained at end of burn-in by $(2.38^2)/D$ (this is for optimal acceptance when target is multivariate normal)
  - Proceed with Metropolis updates of entire τ vector
- Generate predictions using a thinned sample of the parameters

MCMC Worst Case Scenario

- Adaptive MCMC performs well across a variety of scenarios, including when starting values are poor (in the wrong corner of the prior simplex, as below).



Conclusions

- Using compactly-supported covariance can speed up computation time
  - Allows use of derivative information in situations when computations would be prohibitively expensive with dense covariance matrices
- Using compactly-supported covariance may compromise predictive capability relative to “ideal” covariance
  - Higher prediction variances required to achieve nominal coverage when >90% sparsity is required for computational efficiency
  - Depending on complexity of underlying response, larger than nominal prediction bias may be introduced with forced sparsity
- Using only outputs requires 3-4 times more runs to achieve similar prediction quality with regression and 90% sparsity
  - 5-8 times more runs with dense correlation function

Future work

- Test the method against surfaces generated by real simulator codes that produce first partial derivatives, especially as the input dimension grows
- Better understand the equivalent sample size of using gradients vs. not using gradients

References

- Kaufman, C.G.; Bingham, D; Habib, S; Heitmann, K; Frieman, J. (2011), "Efficient emulators of computer experiments using compactly supported correlation functions," *Annals of Applied Statistics*, 5, 2470-2492
- Loeppky, J; Sacks, J; Welch, W (2009), "Choosing the sample size of a computer experiment: a practical guide," *Technometrics*, 51, 366-376
- Morris, M; Mitchell, T; Ylvisaker, D (1993), "Bayesian design and analysis of computer experiments: use of derivatives in surface prediction," *Technometrics*, 35, 243-255
- Santner, T; Williams, B; Notz, W (2003), *The Design and Analysis of Computer Experiments*; Springer, New York.

Thanks!