SAND2015-6813PE

Exceptional service in the national interest

Dakota Software Training

Model Calibration

http://dakota.sandia.gov

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Module Learning Goals

In this module you will learn

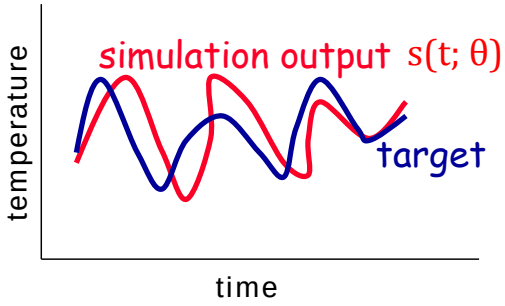
- Why you might want to tune models to match data via calibration (parameter estimation)
- How to formulate calibration problems and present them to Dakota
- What Dakota methods can help you achieve calibration goals

Exercise: create a Dakota calibration study and try to infer unknown parameters for a synthetic data set.

Calibration: Fitting Models to Data

- Use data to improve characterization of input parameter values, by maximizing agreement between simulation output and experiment target
  - Infer unknown conditions, source terms, or properties
  - Tune models to match specific scenarios
  - Make them more robust to predict a range of outcomes

adjust θ so simulation matches target…



- Also known as parameter estimation/identification, inverse modeling
- Can also calibrate one model to another (typically higher fidelity) model

➢ Calibration is not validation! Separate hold-out data must be used to assess whether a calibrated model is valid.

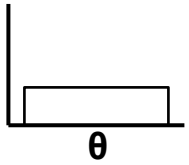
Classes of Model Calibration

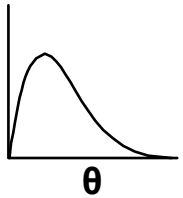
- Goal: maximize agreement between observations y_i and corresponding simulation output $s_i(\theta)$; typically a nonlinear, implicit function of θ (parameterized simulation)

- Deterministic calibration: seek one or more sets of parameter values that best match the data y, typically in the two-norm:

$$\min_{\theta} f(\theta) = SSE(\boldsymbol{\theta}) = \sum_{i=1}^{n} [(s_i(\boldsymbol{\theta}) - y_i]^2 = \sum_{i=1}^{n} [r_i(\boldsymbol{\theta})]^2$$

  - Least-squares: initial iterate θ_0, *nonlinear optimization*, updated values θ
- Statistical calibration: seek a statistical characterization of parameters most consistent with the data

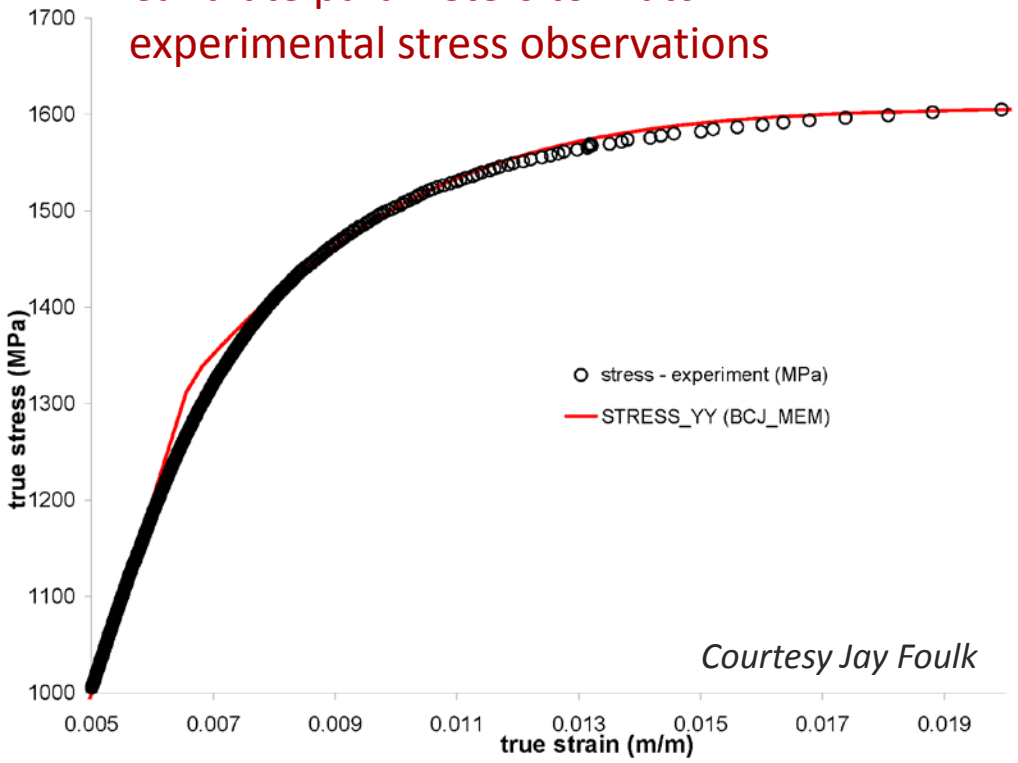




  - Bayesian: prior distribution, *statistical inference (MCMC)*, posterior distribution

Example: Parameter Estimation for a Material Plasticity Model

Calibrate parameters to match experimental stress observations



f – yields rate dependence (fit)
Y – the yield stress (chosen)
n – exponent in flow rule (fit)
H – hardening in evolution of κ (fit)
R_d – recovery in evolution of κ (fit)

f	*4.52 x 10^4*
Y	*1325 MPa*
n	*0.386*
H	*1.10 x 10^5 MPa*
R_d	*389*

NOTE: Experimental data taken from a representative test, ph13-8-h950-test-3

Flow rule concentrating the effective stress

$$\dot{\epsilon_p} = f\{\sinh[\frac{\bar{\sigma}}{(1-\phi)(\kappa+Y)} - 1]\}^n$$

evolution of isotropic hardening

$$\dot{\kappa} = [H - R_d\kappa]\dot{\epsilon_p}$$

**Large values of f make the formulation rate independent. I did not need to fit f.*

Brief Group Discussion: Calibration Practice

- What types of parameter estimation or calibration questions do you ask in your engineering or science domain?
- What kind of data or simulation output are you trying to match?
- What metrics do you use to assess how well the simulation agrees?
- How do you answer your questions currently?
- What are the key challenges you face?

Specifying Calibration Parameters

- Deterministic calibration problems are presented to Dakota using design variables (same as optimization)
- Initial point starts the solve for local methods
- Bounds for the search are typical, but not required for all methods

- See advanced slides for Bayesian methods, which use uncertain variables instead of design

Cantilever calibration variable example

```
variables

 # calibration parameters
 continuous_design 3
  upper_bounds   3.1e7  10    10
  initial_point 2.9e7    4     4
  lower_bounds   2.7e7    1     1
  descriptors     'E'     'w'   't'

 # Fixed config parameters
 continuous_state 4
  initial_state 100 500 1000 500
  descriptors    'L' 'X'   'Y' 'p'
```

Defining Calibration Responses

$$\min_{\theta} f(\theta) = SSE(\boldsymbol{\theta}) = \sum_{i=1}^{n} [(s_i(\boldsymbol{\theta}) - y_i]^2 = \sum_{i=1}^{n} [r_i(\boldsymbol{\theta})]^2$$

Three main options:

1. Interface returns differences (residuals) $r_i(\theta) = s_i(\theta) - y_i$ to Dakota

```
responses
  calibration_terms = 2
  descriptors
    'stress_diff' 'displ_diff'
```

2. Interface returns simulation outputs $s_i(\theta)$ to Dakota; specify data file containing y_i values

```
responses
  calibration_terms = 2
  descriptors
    'sim_stress' 'sim_displ'
  calibration_data_file  'myobs.dat'
    num_experiments = 3
```

3. Interface returns composite objective $f(\theta)$; gives advanced users greater control

```
responses
  objective_functions = 1
  descriptors  'f_SSE'
```

Local nonlinear least squares, Bayesian methods require set of residuals (Option 1 or 2)

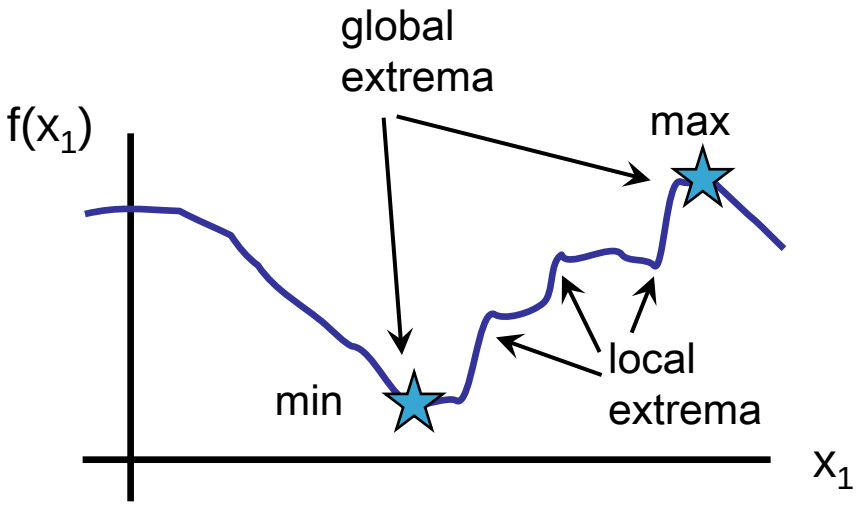
Dakota Calibration Methods

Deterministic

- For local parameter value improvement; reliable simulation derivatives: specialized local least-squares solvers
- Local search with unreliable derivatives: pattern search
- Global best parameter set: global optimizers such as DiRECT or genetic algorithms (can be costly)
- Other advanced optimization approaches

Statistical *(advanced topic)*

- Calibrate distribution parameters to match data: any of the above solvers with a nested model
- Bayesian inference: Markov Chain Monte Carlo (QUESO)



Classes of Methods

Gradient Descent

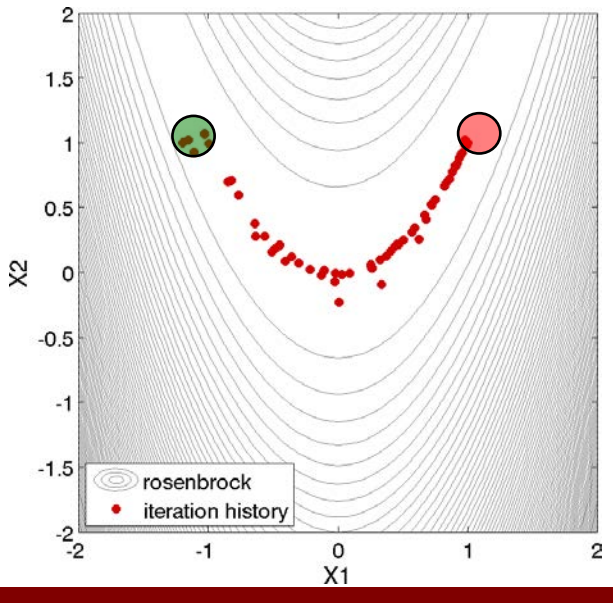
- Looks for improvement based on derivative
- Requires analytic or numerical derivatives
- Efficient/scalable for smooth problems
- Converges to local extreme

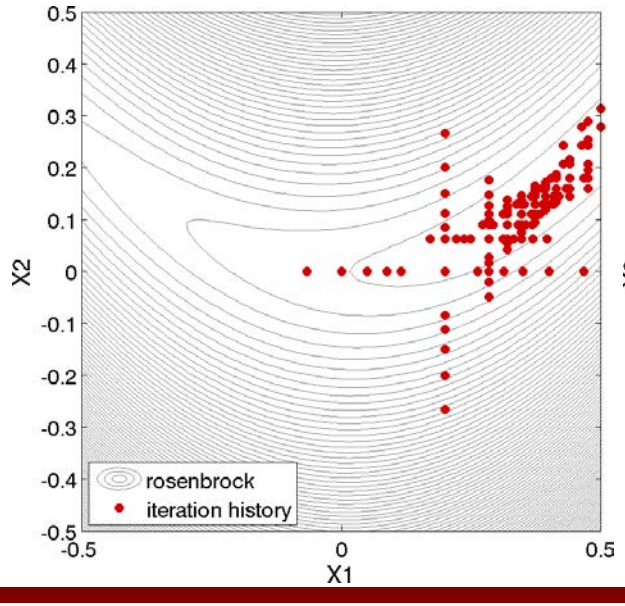
Derivative-Free Local

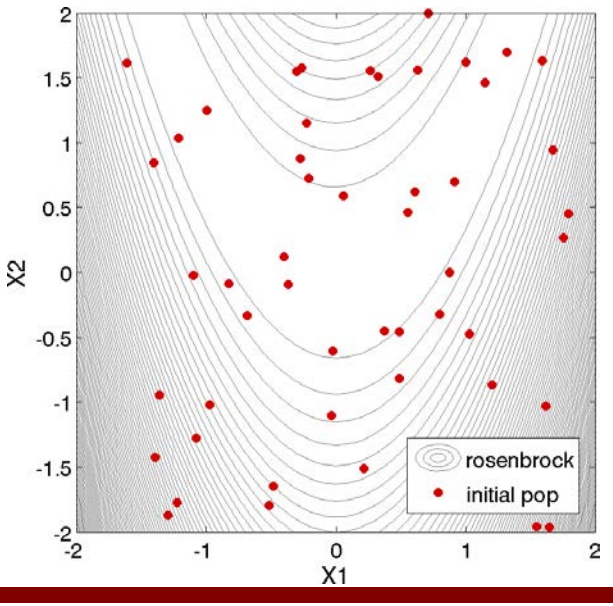
- Sampling with bias/rules toward improvement
- Requires only function values
- Good for noisy, unreliable or expensive derivatives
- Converges to local extreme

Derivative-Free Global

- Broad exploration with selective exploitation
- Requires only function values
- Typically computationally intensive
- Converges to global extreme







More About Local Calibration

- Local, derivative-based least squares solvers are similar to Newton methods for general nonlinear programming
- They can take advantage of the squared residual formulation

$$\frac{SSE}{2} = f(\theta) = \frac{1}{2} r(\theta)^T r(\theta) = \frac{1}{2}\left[s(\theta) - y\right]^T \left[s(\theta) - y\right]$$

$$\nabla f(\theta) = J(\theta)^T r(\theta); \quad J_{ij} = \frac{\partial r_i}{\partial \theta_j} \quad \nabla^2 f(\theta) = J^T J + \sum_{i=1}^{n} r_i(\theta) \nabla^2 r_i(\theta)$$

  and either ignore the circled Hessian term (as residuals should be small as the algorithm converges), or successively approximate it during optimization
- Dakota's NL2SOL local calibration algorithm uses a quasi-Newton update scheme to approximate the Hessian, and is often more robust than other solvers when the residuals are not small.
- These methods can be very efficient, converging in a few function evaluations

Scenario: Calibrating Cantilever Beam

Scenario: Recall that you are using Sandia's Cantilever Physics code to simulate coat hooks as cantilever beams. Before your summer intern left, she also worked with some experimentalists to gather some data to characterize a beam. You have data files with observations of mass, stress, and displacement.

Unfortunately, the experimentalist was unable to fully characterize the beam and experimental conditions. He measured beam width w of 2.5 in., thickness t of 3.0 in., and knows that the vertical load Y was 500.0 lb.

Your task: Use the provided data files and cantilever beam simulator, together with Dakota, to **determine the length L, density ρ, and horizontal load X, of the beam in the experiments**. *Assume a Young's modulus E = 2.9e7.*

Exercise: Find Beam Properties

- The directory ~/exercises/calibration/cantilever contains data files with observations of mass, displacement, and stress
- As experiments were conducted, the observation error was gradually reduced from 10%, though 5%, 2%, 1%, 0.5%, and finally 0.1% by improving the measurement equipment. The files are labeled 10perc through 0.1perc.
- For each level of error, several runs of identical experiments were performed. The data files also are marked according to the number of replicates they contain.
- Complete the Dakota input file dakota_calibration_sketch.in to use NL2SOL to determine the length L, density ρ, and horizontal load X for the beam used in the experiment. Hold the other parameters fixed.

Hints:

- Previous example input files can help with the variables blocks
- See the reference manual sections on:
  - Variables: continuous design, continuous state
  - Responses: calibration_terms (the simulator returns the predicted QOIs), calibration data file and its format, gradient types
  - Scaling (method, variables, responses)

Exercise: Find Beam Properties

- Initially use the data file with 2% observation error and 5 replicates. How do your estimated parameter values compare to your neighbors?
- Is it sensitive to the initial point?
- How do your parameter estimates change as the noise level in the data is reduced from 10% to 0.1%?
- What happens if you use fewer data points? More?
- What do you observe in the final residuals and SSE?
- What happens if you use a pattern search or DiRECT method?

Parameter Identifiability

- Looking at the cantilever beam equations, which parameters would you expect to be able to estimate given data on which responses?
- How would you determine this for an implicit function (black-box simulation)?

$$M = \rho * wt * \frac{L}{12^3}$$

$$S = \frac{600}{wt^2} Y + \frac{600}{w^2 t} X$$

$$D = \frac{4L^3}{Ewt} \sqrt{\left(\frac{Y}{t^2}\right)^2 + \left(\frac{X}{w^2}\right)^2}$$

Guide to Calibration Methods

See Usage Guidelines in User's Manual

Category	Specialized Calibration Methods	General Optimization Methods	Continuous Variables	Categorical/ Discrete Variables	Bound Constraints	General Constraints
Gradient-Based Local (smooth)		optpp_cg	x			
	nl2sol	dot_bfgs, dot_frcg, conmin_frcg	x		x	
	nlssol_sqp, optpp_g_newton	npsol_sqp, nlpql_sqp, dot_mmfd, dot_slp, dot_sqp, conmin_mfd, optpp_newton, optpp_q_newton, optpp_fd_newton	x		x	x
Gradient-Based Global (smooth)	hybrid*, multi_start*	hybrid, multi_start	x		x	x
Derivative-Free Local (nonsmooth)		optpp_pds	x		x	
	surrogate_based_local*	coliny_cobyla, coliny_pattern_search, coliny_solis_wets	x		x	x
		asynch_pattern_search, mesh_adaptive_search	x	x	x	x
Derivative-Free Global (nonsmooth)		ncsu_direct, genie_direct, genie_opt_darts	x		x	
		coliny_direct, efficient_global, surrogate_based_global	x		x	x
		coliny_ea, soga	x	x	x	x

**: in conjunction with a specialized gradient-based method (nl2sol, nlssol, optpp_g_newton)*

Calibration References

- G. A. F. Seber and C. J. Wilde, “Nonlinear Regression”, John Wiley and Sons, Inc., Hoboken, New Jersey, 2003.

- M. C. Hill and C. R. Tiedeman, “Effective Groundwater Model Calibration: With Analysis of Data, Sensitivities, Predictions, and Uncertainty”, John Wiley and Sons, Inc., Hoboken, New Jersey, 2007.

- R. C. Aster, B. Borchers, and C. H. Thurber, “Parameter Estimation and Inverse Problems”, Elsevier, Inc., Oxford, UK, 2005.

- Dakota User’s Manual
  - Nonlinear Least Squares Capabilities
  - Surrogate-Based Minimization
- Dakota Reference Manual

BACKUP/ADVANCED TOPIC SLIDES

Local Least-squares Methods
Take Advantage of Special Structure

Alternate objective formulation: $f(x) = \frac{1}{2} r(x)^T r(x) = \frac{1}{2}\left[s(x) - d\right]^T \left[s(x) - d\right]$

Derivative formulations:

$$\nabla f(x) = J(x)^T r(x); \quad J_{ij} = \frac{\partial r_i}{\partial x_j} \qquad \nabla^2 f(x) = J^T J + \sum_{i=1}^{n} r_i(x) \nabla^2 r_i(x)$$

Methods vary in second derivative approximation:

Gauss-Newton: $J(x)^T J(x)$

Levenberg-Marquardt: $J(x)^T J(x) + \mu I$, with $\mu \geq 0$

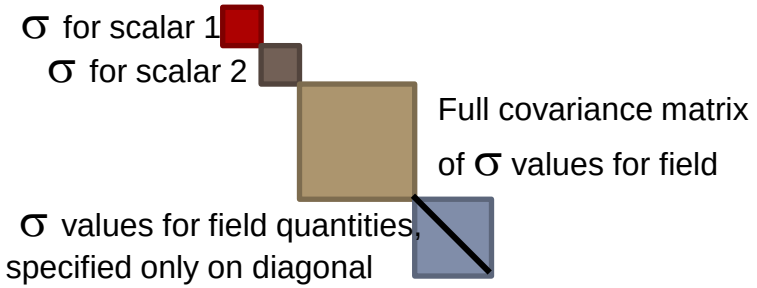
$NL2SOL: J(x)^T J(x) + S,$

with $S = 0$ or $S =$ Quasi-Newton approximation to $\sum_{i=1}^{n} f_i(x) \nabla^2 f_i(x)$

Field Response Capability

Simulation and Experiment

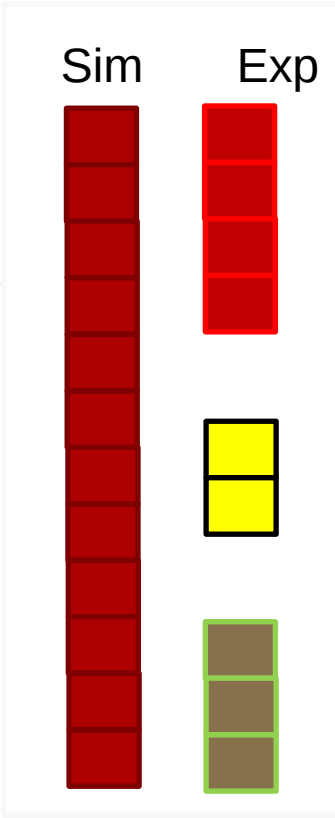
- Dakota response comprised of scalars and/or fields
- Field responses are a relatively new feature, meant to streamline multiple responses (e.g. instead of the user pulling 50 temperature values from a time-temperature curve for 50 separate scalar responses, there is one field response of length 50. In this way, it is easier to have more complete fields: the field may have 1000 values).
- Field has zero or more independent coordinates (abscissas), e.g., f(t,x,y)
- Number of simulation and experiment observations may differ: Dakota provides interpolation to calculate the sum-of-squares error.
- New specification for "measurement" uncertainty:
  (block) scalar, diagonal, full

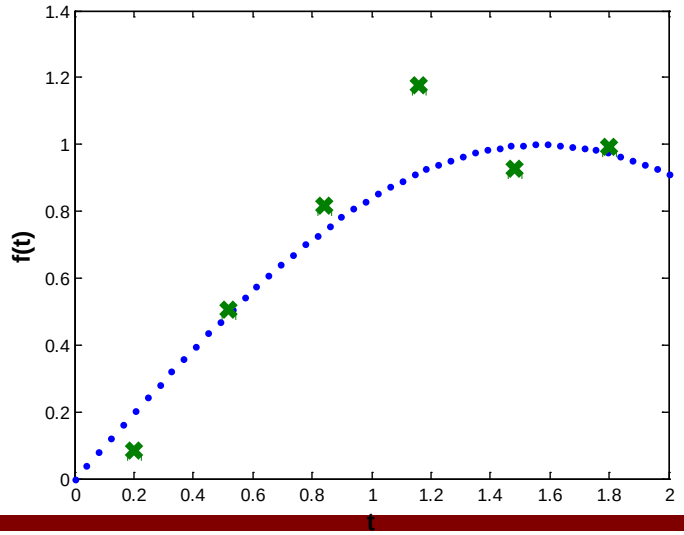


- Can be used in traditional deterministic or Bayesian calibration

Basic Interpolation Capability

- For each field, allow (1-D) interpolation of simulation sim(t_i) onto observed experiment abscissas exp(t_j)
- Each experiment can have a different number of observations, e.g., 4, 2, 3 time points
- Results in 1 calibration residual per experiment datum
- Does not supplant discretization-aware interpolation or user-provided simulation to experiment metrics





Potential Solution: Cantilever Least-Squares

```
# Calibrate to area, stress, and displacement data generated with
# E = 2.85e7, w = 2.5, t = 3.0

method
  nl2sol
    convergence_tolerance = 1.0e-6

variables
  continuous_design = 3
    upper_bounds  3.1e7 10.0 10.0
    initial_point 2.9e7 4.0  4.0
    lower_bounds  2.7e7 1.0  1.0
    descriptors   'E' 'w' 't'
  # Fix at nominal
  continuous_state = 3
    initial_state 40000 500 1000
    descriptors 'R' 'X' 'Y'

interface
  direct
    analysis_driver = 'mod_cantilever'

responses
  calibration_terms = 3
#    calibration_data_file = 'dakota_cantilever_clean.dat'
    calibration_data_file = 'dakota_cantilever_witherror.dat'
    descriptors = 'area' 'stress' 'displacement'
  analytic_gradients
  no_hessians
```

Confidence Intervals approximated by calculating the variance of the parameter vector as diagonal elements of:

$$\hat{\sigma}^2 (J^T J)^{-1}$$

```
CIs without error:
E: [ 2.850e+07, 2.850e+07 ]
w: [ 2.500e+00, 2.500e+00 ]
t: [ 3.000e+00, 3.000e+00 ]
```

```
CIs with error:
E: [ 1.992e+07, 4.190e+07 ]
w: [ 1.962e+00, 3.918e+00 ]
t: [ 1.954e+00, 3.309e+00 ]
```

Bayesian Calibration of Computer Models

- Combine prior parameter distributions with data to update the statistical characterization of the parameters
  - Prior distribution and statistical inference (MCMC) → posterior distribution
- Generate posterior distributions on model parameters, given
  - Experimental data
  - A prior distribution on model parameters
  - A presumed probabilistic relationship between experimental data and model output that can be defined by a likelihood function

$$\pi(\theta \mid d) \propto \pi(\theta) L(d \mid \theta)$$

Model parameters

Observed Data

Prior parameter distribution

Likelihood function which Incorporates the model

Bayesian Calibration of Computer Models

- Experimental data = Model output + error

$$d_i(\boldsymbol{x}_i) = M(\boldsymbol{\theta}, \boldsymbol{x}_i) + \varepsilon_i$$

- If we assume error terms are independent, zero mean Gaussian random variables with variance σ^2, the likelihood is:

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(d_i(\boldsymbol{x}_i) - M(\boldsymbol{\theta}, \boldsymbol{x}_i))^2}{2\sigma^2} \right]$$

- How do we obtain the posterior?
  - It is usually too difficult to calculate analytically
  - We use a technique called Markov Chain Monte Carlo (MCMC)
  - In MCMC, the idea is to *generate a sampling density that is approximately equal to the posterior.* We want the sampling density to be the stationary distribution of a Markov chain.

Markov Chain Monte Carlo

Posterior $\propto$ Likelihood x Prior

- Metropolis-Hastings is a commonly used algorithm
- It has the idea of a "proposal density" which is used for generating θ_{i+1} in the sequence, conditional on θ_i.
  - Generate new θ' from current θ by drawing sample from multivariate proposal dist. centered at θ
  - Evaluate ratio $\alpha = p(\theta' | \mathbf{d}) / p(\theta | \mathbf{d})$, from likelihood and prior
  - Accept sample for $\alpha \geq 1$; conditionally accept $\alpha < 1$ with probability α
- MCMC Method Issues:
  - Typically requires $O(10^4) – O(10^5)$ samples → prohibitive for direct use with high fidelity models
  - Convergence rate can degrade with dimension due to increased mixing time, producing higher variance in posterior estimates
  - Need to tune the proposal density to get an "optimal" acceptance rate, 45-50% for 1-D problems, 20-30% for high dimensional problems
- COMPUTATIONALLY VERY EXPENSIVE

Surrogate Models

- Since MCMC requires tens of thousands of function evaluations, it is necessary to have a fast-running surrogate model of the simulation
- Dakota has the capability for using the following surrogates in the Bayesian calibration:
  - Gaussian Processes
  - Polynomial Chaos Expansions
  - Stochastic Collocation
- Steps for a Bayesian analysis:
  - Take initial set of samples from simulation
    - Use LHS or Sparse Grid
  - Develop surrogate approximation of the simulation
  - Define priors on the input parameters
  - Perform Bayesian analysis using MCMC
  - Generate and analyze posterior distributions

Bayesian Calibration Methods in Dakota

- **QUESO** is a library of UQ methods developed at the UT PECOS center.
- Allows for four variations of the DRAM MCMC algorithm:
  - `metropolis_hastings` (no DR, no AM)
  - `delayed_rejection` (DR only)
  - `adaptive_metropolis` (AM only)
  - `dram` (both DR and AM)
  - Also have the `multilevel` option, an algorithm which uses an homotopy-like approach to move the prior incrementally to the posterior.
- We currently can perform Bayesian calibration with a simulation directly (no emulator), with a Gaussian process emulator, or with a polynomial chaos or stochastic collocation emulator.
- The user can specify a variety of options for the `proposal_covariance`, including
  - `derivatives` (precondition the proposal covariance with the inverse of the Hessian of the misfit)
  - `prior` (use the variance of the prior input distributions as diagonal elements of the covariance matrix)
  - `values` (specify what values you want, as diagonal elements or a full covariance matrix)

Bayesian Calibration Methods in DAKOTA

- **DREAM.**
  - DiffeRential Evolution Adaptive Metropolis
  - Algorithm implementation developed by John Burkardt.
  - The DREAM approach runs multiple different chains simultaneously for global exploration, and automatically tunes the proposal covariance during the process by a self-adaptive randomized subspace sampling
- **GPMSA.** GPMSA (Gaussian Process Models for Simulation Analysis) is a code developed by Brian Williams, Jim Gattiker, Dave Higdon, et al. at LANL.
  - Original LANL code is in Matlab.
  - GPMSA was re-implemented in the QUESO framework. We have an initial wrapper to it in Dakota, but much of it is hardcoded, not ready for general applications yet.
  - Need to handle “functional” data.

DAKOTA Bayesian Example

```
method,
   bayes_calibration queso
   samples = 1000 seed = 348
   dram
   # | delayed_rejection | adaptive_metropolis
   # | metropolis_hastings
   proposal_covariance
      diagonal values 1.0e6 1.0e-1

variables,
   uniform_uncertain 2
     upper_bounds  1.e8  10.0
     lower_bounds 1.e6  0.1
     initial_point  2.85e7  2.5
   descriptors 'E' 'w'
      continuous_state 4
      initial_state 3   40000   500   1000
      descriptors 't' 'R' 'X' 'Y'

interface,
  system
  analysis_driver = 'cantilever3'
```

```
responses,
   calibration_terms = 2
   calibration_data_file =
'dakota_cantilever_queso.withsigma.dat'
   freeform
   num_experiments = 10
   variance_type = 'scalar'
   descriptors = 'stress' 'displacement'
   no_gradients
   no_hessians
```

Bayesian Calibration for Heat Transfer Closure Law

- Use data to refine predictions with COBRA-TF
- **Nusselt number (*Nu*):** ratio of convective to conductive heat transfer normal to the boundary (wall heat transfer)
- **Dittus-Boelter**, originally calibrated to 13 data sets:

$$Nu = 0.023\ \mathrm{Re}^{0.8}\ \mathrm{Pr}^{0.4} = \theta_1 \mathrm{Re}^{\theta_2}\ \mathrm{Pr}^{\theta_3}$$

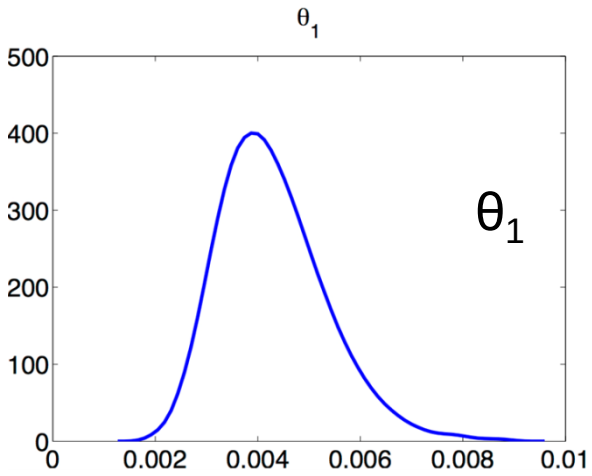
- Make the Dittus-Boelter coefficients into parameters θ. In a typical sensitivity or uncertainty study, might use expert opinion to say

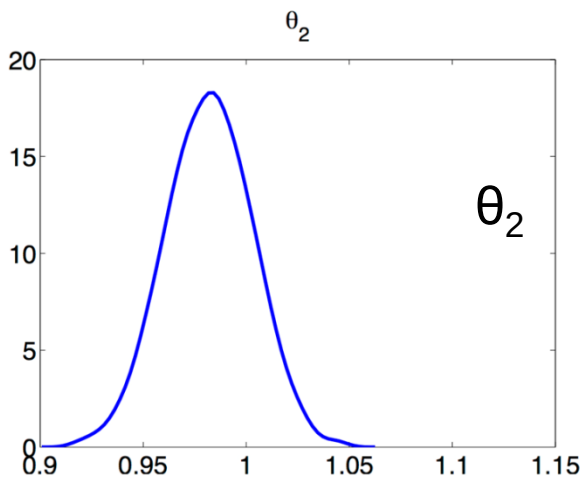
$$\theta_1 \in [0.0, 0.046] \qquad \theta_2 \in [0.0, 1.6] \qquad \theta_3 \in [0.0, 0.8]$$

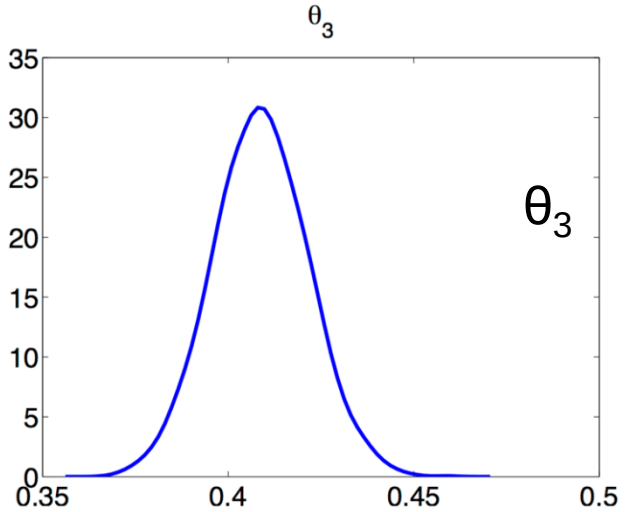
- Instead use DRAM for Bayesian calibration to find possible θ consistent with data
- This analysis based on one data set:
  Morris and Whitman, "Heat Transfer for Oils and Water in Pipes," *Industrial and Engineering Chemistry*, **Vol. 20, No. 3**, pp.234-240, 1928.

Updated Parameter Distributions

- Lead coefficient θ_1 now lower: 0.004 from 0.023.
- Reynolds exponent θ_2 now larger: 0.99 from 0.8
- Prandlt exponent θ_3 slightly larger: 0.41 from 0.4.

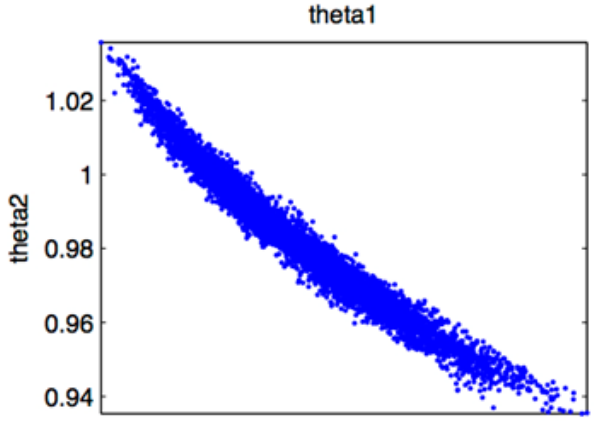




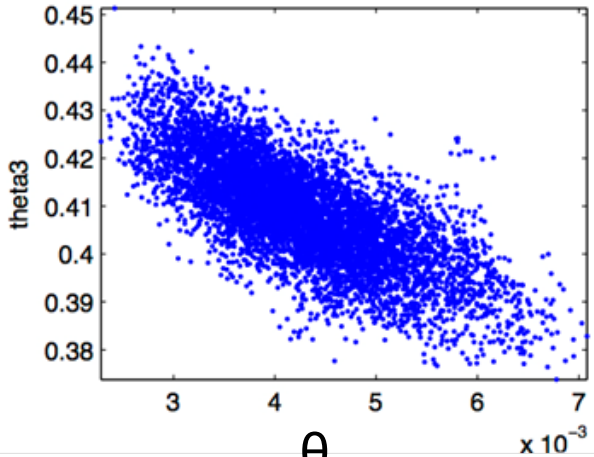


- (Only used 1 of 13 data sets informing Dittus Bolter)
- Most importantly, have uncertainty estimates

Joint Samples for the Parameters

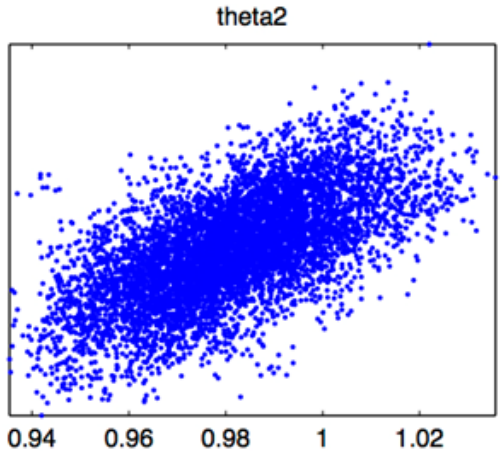


θ_2



θ_3

θ_1



θ_2

- Bayesian analysis reveals correlations among parameters
- If lead coefficient increases, Reynold's exponent must decrease
- Defines a 3-D surface indicating combinations of parameters that best match the data