SAND2006-4718C

The DAKOTA Toolkit and its use in Computational Experiments

Anthony A. Giunta (Product Manager)
Michael Eldred (Principal Investigator and Team Lead)
Laura Swiler

Sandia National Laboratories
Optimization and Uncertainty Estimation
Albuquerque, NM
87185-1318

lpswile@sandia.gov
505-844-8093

August 11-14, 2006
NSF-SAMSI Summer School on Computer Experiments

*Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Outline/Agenda

- DAKOTA history and background
- DAKOTA methods
  - Parameter study
  - Uncertainty quantification
  - Optimization
- DAKOTA input/output/script files
- Design of Computer Experiments
  - DACE
  - FSUDACE
  - LHS
- Engineering applications
- What’s new in DAKOTA version 4.0

What is DAKOTA? Executive Summary

- **DAKOTA: Design and Analysis toolKit for Optimization and Terascale Applications**
  - **Under development at SNL since 1994**
  - **State of the art tools for performing engineering "what if" studies:**
    - **Uncertainty quantification, sensitivity analysis, computer model calibration, design optimization, etc.**
    - **Extensive support for parallel computing – PCs to supercomputers**
  - **Works as a "black-box" with your simulation code(s):**
    - **Data transferred via file read/write operations**
    - **Works on LINUX/UNIX, Mac OS, Windows**
  - **In use at SNL, LLNL, LANL, ORNL, Navy, NASA, Lockheed-Martin, 3M, Kodak, Goodyear, etc. and at numerous universities**
  - **Freely available worldwide via GNU General Public License**
    - **~3000 downloads, approx several hundred "serious" users**
  - **DAKOTA team receives significant return on investment from external users:**
    - **Bug reports, compilations on new computer systems, suggestions for future R&D, research collaborations**
  - **DAKOTA enables sensitivity analysis, optimization, and uncertainty quantification w/ high-fidelity simulation tools on massively-parallel supercomputers.**

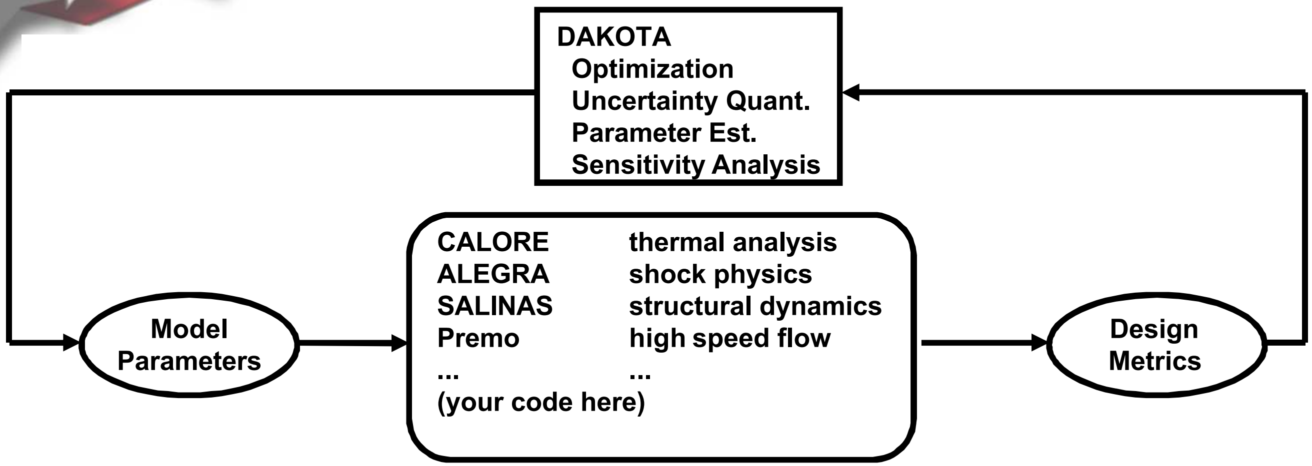
**roughly 500k lines of code total, with ~100k in DAKOTA "core"*

What is the Role of DAKOTA in Engineering/Science Applications?

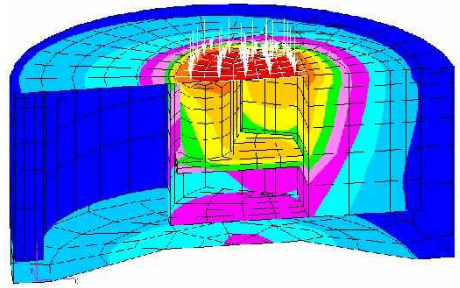
- **DAKOTA enables sensitivity analysis, optimization, and uncertainty quantification (UQ) to help answer "what if..." questions.**
  - What happens to my cost (or safety margin or performance level or ...) if I change parameter X?"
  - How reliable is my design?
  - How safe is my design?
  - What is the best design?

- **DAKOTA assists the analyst/designer in understanding and managing complex computer models.**
  - Automate typical "parameter variation" studies.
  - Discover/predict nonlinear interactions among many parameters.
    - Interactions that might be missed with traditional "change one parameter at a time" studies.
  - Support experimental testing efforts:
    - Examine many accident conditions with computer models, then physically test only a few of the worst-case conditions.

DAKOTA Overview

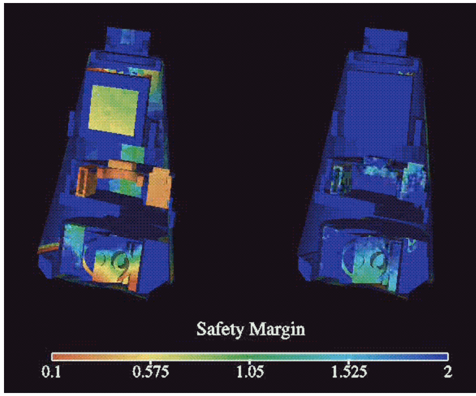


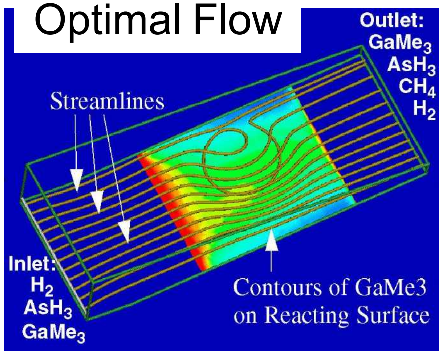
Worst Case Fire Safety



Structural Design

Nominal Optimized





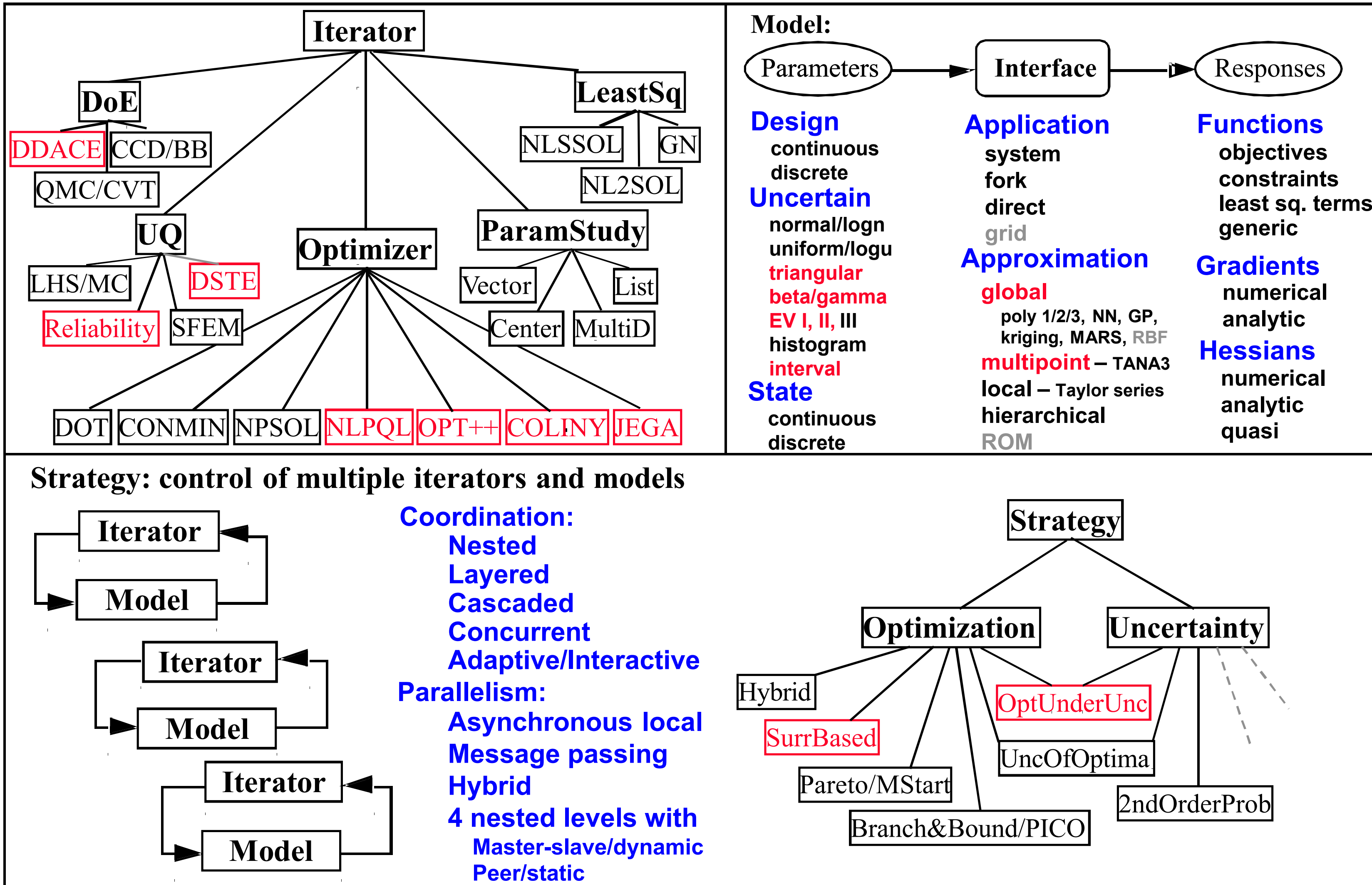
Technical themes

- **DAKOTA works with just about any simulation code.**
  - **Typically in "black box" mode with file reading/writing.**
  - **DAKOTA contains math & stats methods – no physics!**
- ***Exploit large-scale/massively parallel computing platforms***
- **Create and deploy state-of-the art methods for complex and expensive engineering simulations, e.g.:**
  - **Surrogate-based optimization & uncertainty quantification**
- ***Balance research goals with production software support***

Impact

- **Internal: extensive (and growing) use within SNL**
- **External: DOE labs, DoD labs, NASA, commercial & academic partners**

DAKOTA Framework



Sandia National Laboratories

Some Notation....

- **“Design parameters”: x1, x2, ..., xN**
  - **X’s can be real-valued**
  - **X’s can be discrete-valued (integer or real)**
  - **X’s can be uncertain (e.g., probability distributions)**
  - **X’s can be a mix of all of these**

- **Examples:**

  -123.4 <= x1 <= 567.8

  x1 is an element of [1.0, 1.2, 1.4. 1.6. 1.8, 2.0]

  x1 is an element of [-2, -1, 0, 1, 2]

  x1 has Normal distribution with a specific mean and standard deviation

- **The collection of all possible design parameter values defines the “parameter space.”**
  - **e.g., if x1 and x2 are both real-valued on [0,1], then the parameter space is the unit square**
  - **e.g., if x1 and x2 are both discrete-valued on [0, 0.5, 1.0], then the parameter space is a 3x3 grid of points on the unit square**

Some Notation....

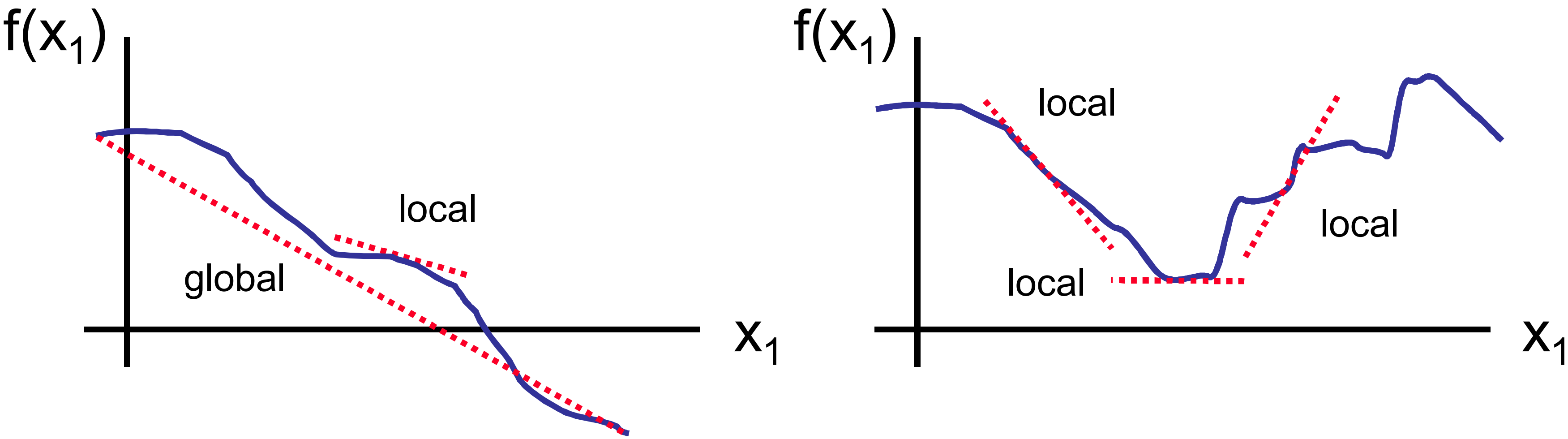
- "Response functions" are dependent on the design parameters.
  - aka: "figures of merit," "objective functions"
  - These functions capture the "goodness" of what you are studying: cost, weight, speed, penetration depth, discrepancy between simulation data & test data, etc.
  - For most real-world applications, the actual mathematical form of the response function is not known precisely, and the response is evaluated via a complex computer code.
    - In cases where the functional form is known, this knowledge can be exploited
  - These functions are real/discrete/uncertain depending on the form of the design parameters.
- Typically, a problem of interest has more than one response function:

  f1 = fcn1(x1,...); f2 = fcn2(x1,...); ...; fM = fcnM(x1,...)
  - Multiple objectives (minimize both cost & weight)
  - Objectives and constraints (minimize weight, subject to cost <= cost_limit)

Sandia National Laboratories

a1

Examples of Sensitivity Analysis



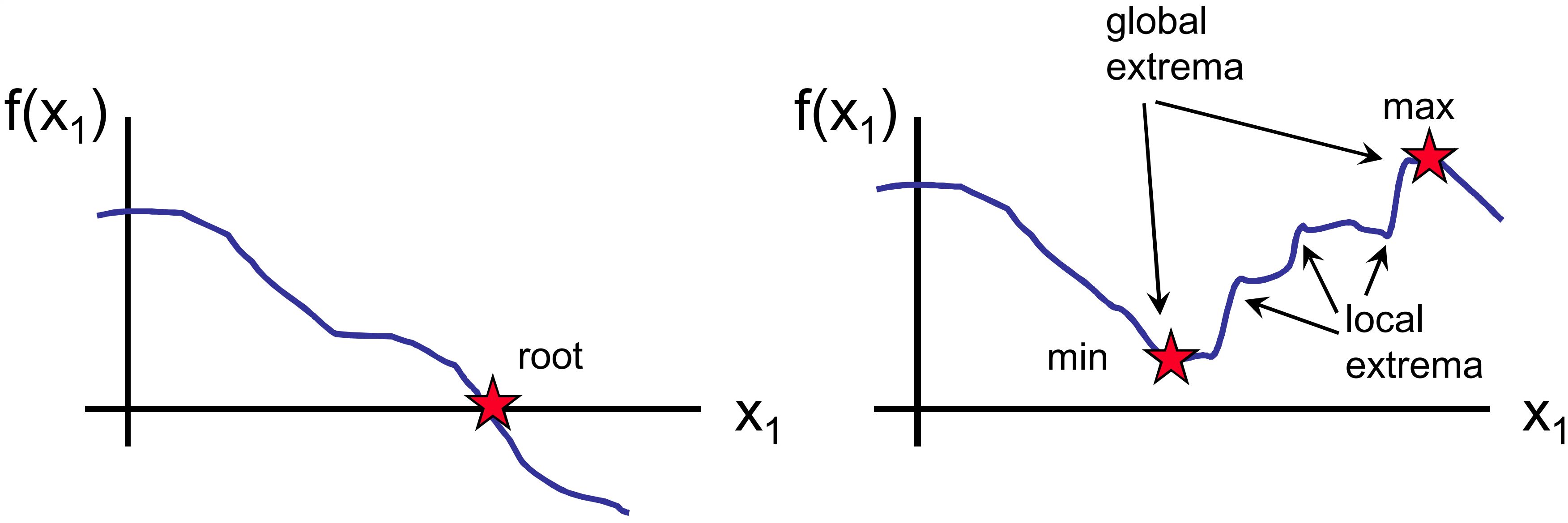
- **Sensitivity analysis examines variations in $f(x_1)$ due to perturbations in x_1**
  - **Local sensitivities are typically partial derivatives.**
    - **Given a specific x1, what is the slope at that point?**
  - **Global sensitivities are typically found via least squares.**
    - **What is the trend of the function over all values of x1?**

Sandia National Laboratories

Slide 9

a1 animate the local gradient lines
aagiunt, 4/17/2006

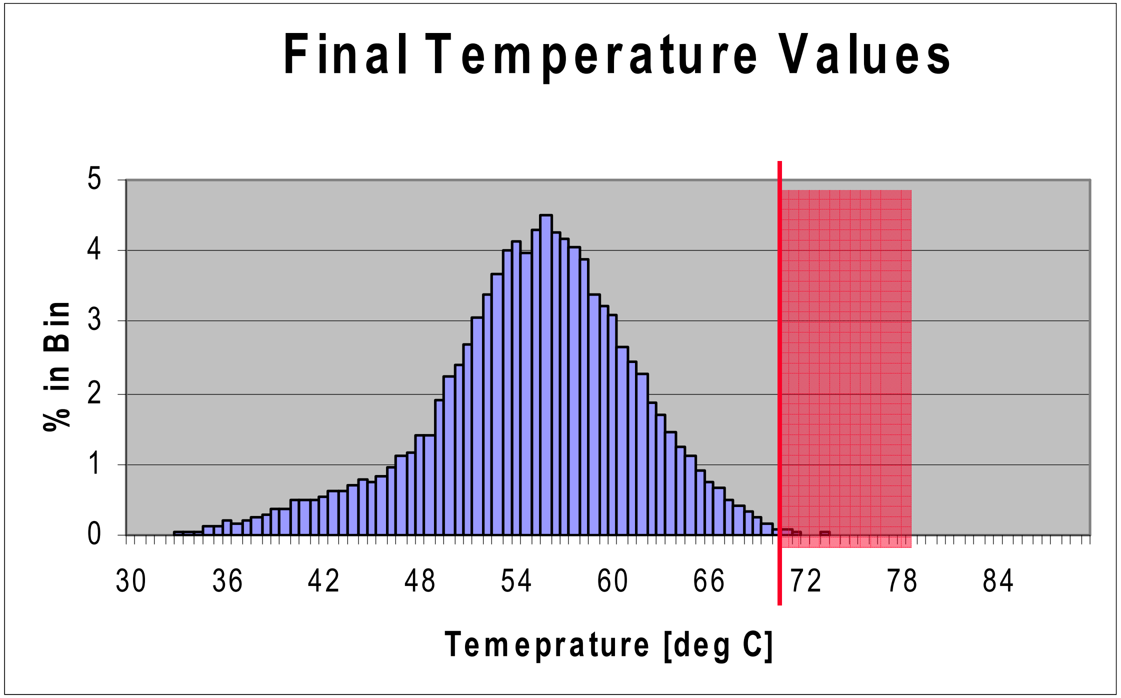
Examples of Optimization



- Optimization methods find the “zero” or “root” of a function and/or the extrema (min/max points) of a function
  - Some opt methods use gradient information to guide their search process
  - e.g., find the value of x1 where (f_simulation – f_test)=0, or, where (f_sim – f_test) is minimized.

Sandia National Laboratories

Example of Uncertainty Quantification



Note: DAKOTA does not fit a statistical distribution to the f-value data. This type of statistical analysis can be performed with statistical analysis software: JMP, Minitab, S-Plus, Matlab Stats Toolbox, etc.

- **UQ methods provide info on the statistics of the responses.**
  - **Correlations of f-values to x-values**
  - **Mean(f), StdDev(f), Probability(f >= f_critical)**
  - **Example: x = uncertain test condition, f = temperature**
  - **Choose many x-values, run code for each set of x-values, collect f-values, plot and analyze the f-values**

Sandia National Laboratories

Summary of DAKOTA's Utility

- **In many applications, we want to understand how changes in our x-values affect the f-values from our simulation code.**
- **Traditional process – user changes one x-value at a time, runs the simulation code, and observes the change in the f-value.**
  - **Tried and true, but, can miss important multi-parameter interactions.**
- **Computer-aided process – DAKOTA selects the x-values, runs the simulation code, collects all of the f-values, and presents "data" to user.**
  - **"data" = local/global sensitivity info**
  - **"data" = optimization info on x- and f-values**
  - **"data" = probability & statistical info on f-values**
  - ***Then, the user decides how to proceed – new x-value ranges, focus in on x-values of interest, etc.***
  - ***The user is still in control of the analysis/design process!***

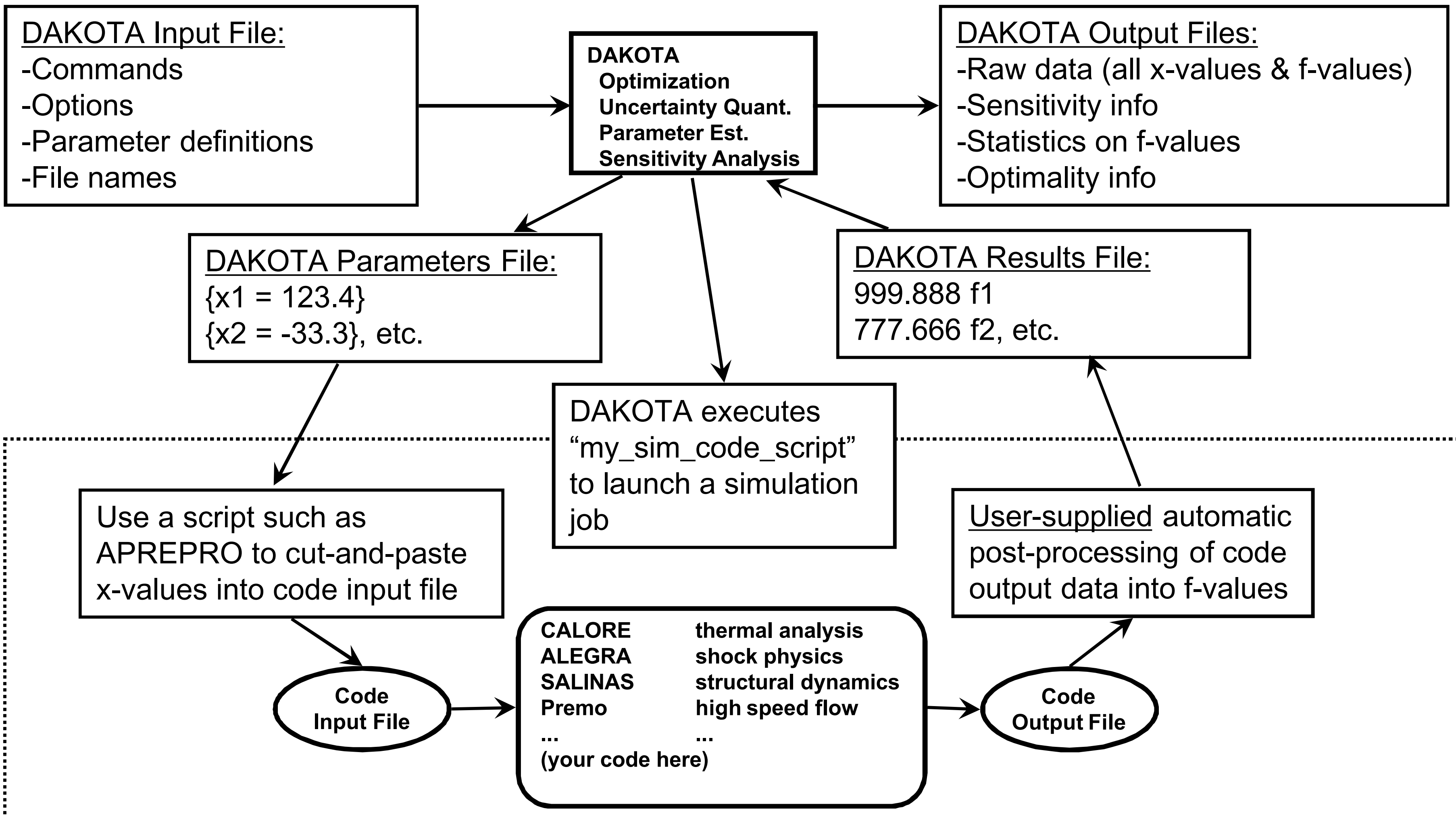
Sandia National Laboratories

DAKOTA Execution & Info Flow

- **What files go into DAKOTA?**
- **What files come out of DAKOTA?**
- **How does DAKOTA interact with my simulation code?**

Sandia National Laboratories

DAKOTA Execution & Info Flow



What is the Format of DAKOTA's "results.out" File?

- **This question comes up a lot.**
- **Answer:**
  - **List the f-values in order, one value per line in the file:**
    f1
    f2
    ...
    fM
- **Note:**
  - **You can add a text label after the function value (on the same line) to help you keep track of the f-values**
- **If your code generates gradients of the f-values and/or Hessian values (matrix of 2^{nd} derivatives), DAKOTA can use this info. See the DAKOTA Users Manual, or contact me, for more info.**

Sandia National Laboratories

Questions to Consider Before Using DAKOTA

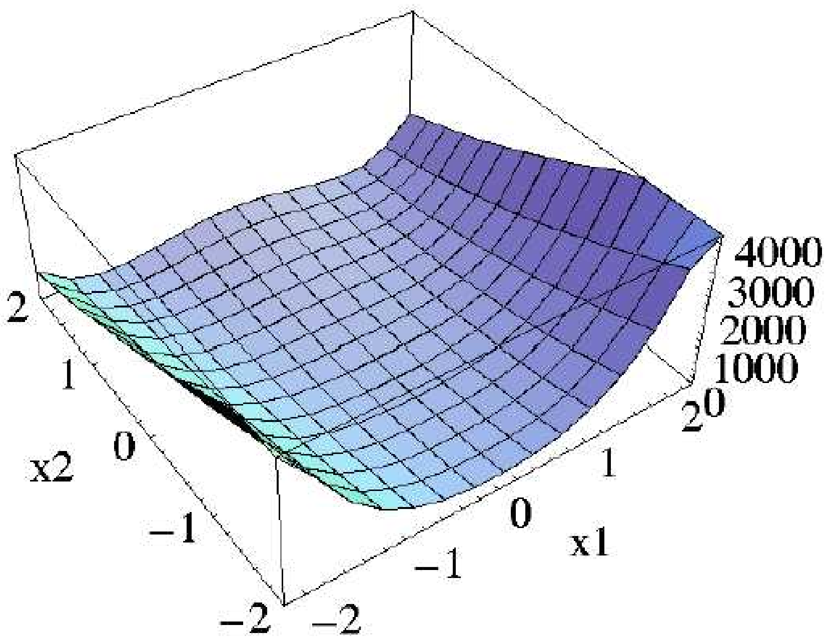
1. What do I want to find out?
   - Sensitivity study? Optimization study? UQ study?

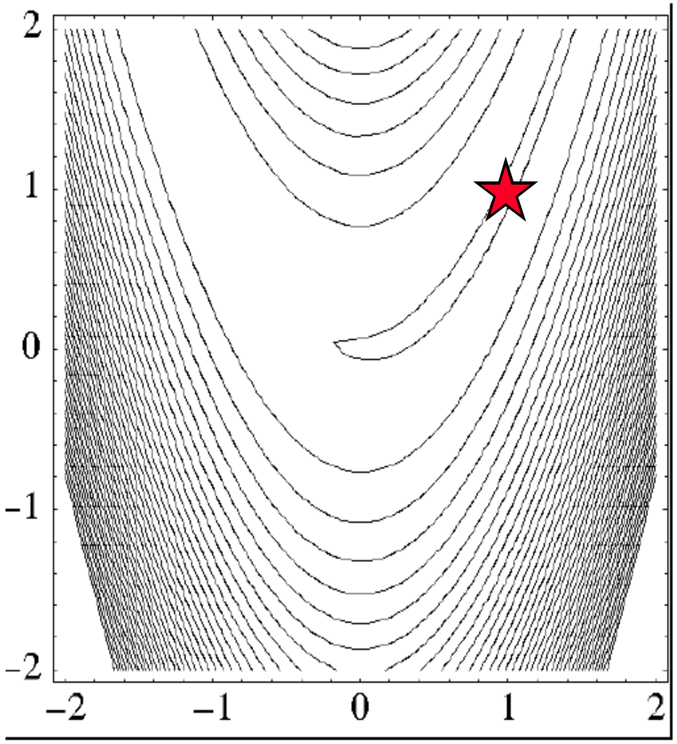
2. How many runs of my simulation code can I afford?
   - 10’s, 100’s, 1000’s, more?
   - How many processors per simulation code run?

3. Where am I going to run the simulation code?
   - On my PC? On my Mac?
   - On my Linux/Sun/SGI/IBM workstation?
   - On a network of workstations?
   - On a Linux/Sun/SGI/IBM cluster?
   - On a special supercomputer?

Of these, #1 and #2 are the most critical!

DAKOTA Examples: Rosenbrock Function





f(x1,x2) = 100*(x2-x1*x1)^2 + (1-x1)^2

-2 <= x1 <= 2

-2 <= x2 <= 2

Optimum point: (x1,x2) = (1,1); f(1,1) = 0.0

Sandia National Laboratories

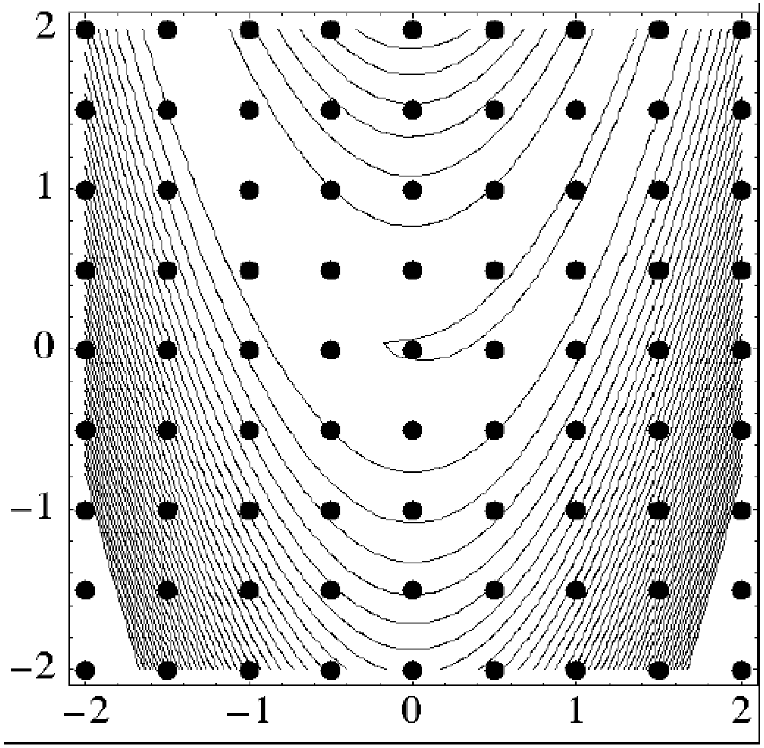
Parameter Study Methods

- Goal of a parameter study:
  - Sensitivity of f-values due to small x-value changes
  - Sensitivity of f-values due to large x-value changes
- Multidimensional parameter study:
  - Good if your simulation code is not expensive
  - Good if your # of design parameters "N" is small
  - Pros: readily amenable to 2-D and 3-D plotting
  - Cons: doesn't scale well with large "N"
- Vector parameter study:
  - Good if your simulation code is expensive and/or if your # of design parameters is large

Sandia National Laboratories

Multidimensional Parameter Study

- **Example: 9 equally-spaced samples on x1 and x2 intervals.**
  – **Yields 81 samples on a regular grid**
- **Great for plotting!**
  – **Generates a table of data for Matlab, Excel, Mathematica, etc.**
- **Flexible - could have used a different # of samples for x1 and x2.**
- **Warning: this method scales, at best, as # samples = 2^N:**
  – **N = # of parameters, and the "2" comes from sampling at the endpoints of each parameter interval:**
  – **For N=2, there are 4 samples**
  – **For N=3, there are 8 samples**
  – **For N=5, there are 32 samples**
  – **For N=10, there are 1024 samples**
  – **For N=20, there are ~1 million samples***

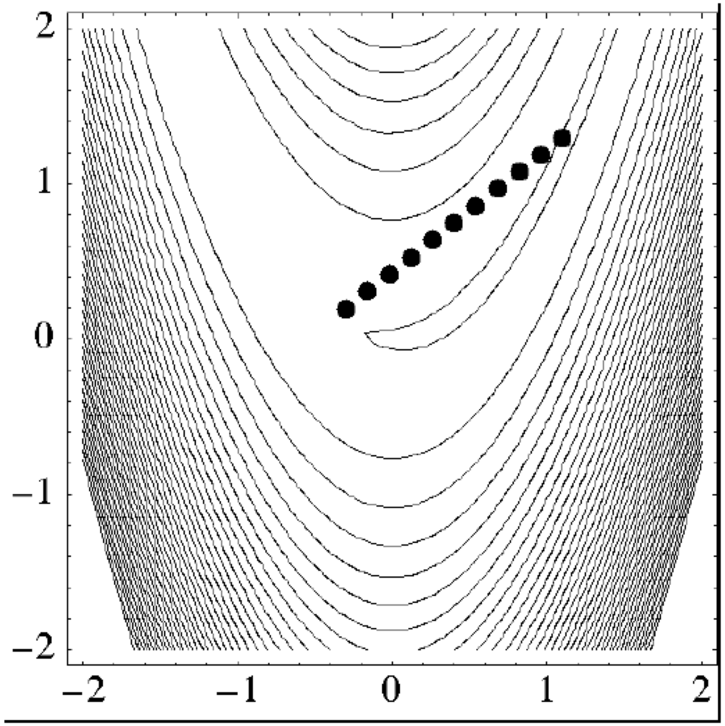


This is the "curse of dimensionality" – more info on this coming in the UQ section.

Sandia National Laboratories

Vector Parameter Study

- **Example: 11 equally-spaced samples along a vector in the x1-x2 parameter space.**
  - **User defines the starting point, ending point, and number of samples**
  - **In 2-D, this is like "walking" from point A to point B.**
  - **In 3-D, this is like "flying" from point A to point B.**
- **This method is not especially useful with N=2, but is very useful in N>2.**
  - **With big steps, this provides some global trend info on f-values.**
  - **With small steps, this provides some local trend info on f-values (quasi derivatives).**



Example Input/Output: Parameter Studies

```
# DAKOTA Users Manual Ch 2 example problems
# using Rosenbrock's function

strategy,                                   \
        single_method                       \
        tabular_graphics_data

method,                                     \
        multidim_parameter_study            \
          partitions = 8 8                  \

model,                                      \
        single

variables,                                  \
        continuous_design = 2               \
          cdv_initial_point     0.0   0.0       \
          cdv_lower_bounds   -2.0   -2.0      \
          cdv_upper_bounds    2.0    2.0      \
          cdv_descriptors     'x1'   'x2'      \

interface,                                  \
       system                               \
          analysis_driver = 'rosenbrock'        \
           parameters_file = 'params.in'        \
           results_file    = 'results.out'

responses,                                  \
        num_objective_functions = 1         \
        no_gradients                        \
```

Parameter Study Input

%eval_id	x1	x2	obj_fn
1	-2	-2	3609
2	-1.5	-2	1812.5
3	-1	-2	904
4	-0.5	-2	508.5
5	0	-2	401
6	0.5	-2	506.5
7	1	-2	900
8	1.5	-2	1806.5
9	2	-2	3601
10	-2	-1.5	3034
11	-1.5	-1.5	1412.5
12	-1	-1.5	629
13	-0.5	-1.5	308.5
14	0	-1.5	226
15	0.5	-1.5	306.5

Parameter Study Output

Sandia National Laboratories

Optimization Methods

- Optimization problem formulation
- Key decision criteria for optimization method selection
- Examples of optimization methods
  - Gradient-based methods
  - Non-gradient pattern search
  - Non-gradient genetic algorithms
- Summary

Sandia National Laboratories

Optimization Problem Formulation

Minimize:	**f1(x1, ...,xN)***	Objective function
Subject to:	**-1.0 <= f2 <= 1.0** **...** **-1.0 <= fM <= 1.0**	Linear or nonlinear constraints
	0.0 <= x1 <= 1.0 **...** **0.0 <= xN <= 1.0**	Bound constraints

***Note: in practice, we can have multiple f-values in the objective function (aka "multiobjective optimization")**

Sandia National Laboratories

Optimization Problem Lingo*

Unconstrained problem –neither bound constraints nor linear/nonlinear constraints

Bound-constrained problem – bound constraints only, no linear/nonlinear constraints

Linearly-constrained problem – has bound constraints, and the f-value constraints are linear with respect to the x-values

- in this case, the f-value constraints can be written via a linear system of equations

Nonlinearly-constrained problem* – has bound constraints, and the f-value constraints are nonlinear w.r.t. the x-values

*These are just the most basic problem definition terms

**Most engineering opt problems I encounter are nonlinearly-constrained.

Getting Ready for an Optimization Study via DAKOTA

- **Key decision criteria:**
  - Smoothness of the sensitivity study data – both large scale and small scale trends
  - Expense of the simulation
  - Types of constraints in your optimization problem
  - Optimization goal: local optimization or global optimization?

- **Extreme Case #1 – all optimization methods will work:**
  - F-values have smooth local and global trends
  - Can afford millions of simulation code runs
  - No constraints
  - Local optimization is the goal

- **Extreme Case #2 – no optimization methods will work:**
  - F-values have nonsmooth local and global trends
  - Can afford, at most, two simulation code runs
  - Many nonlinear constraints (that are also nonsmooth)
  - Global optimization is the goal

Recipe for an Optimization Study via DAKOTA

Here is my basic optimization method decision tree:

Unconstrained or bound-constrained problems:

- **Smooth and cheap: just about any method will work, gradient-based methods will be fastest**
- **Smooth and expensive: gradient-based methods**
- **Nonsmooth and cheap: non-gradient methods such as pattern search (local opt), genetic algorithms (global opt), DIRECT (global opt), or surrogate-based optimization (quasi local/global opt)**
- **Nonsmooth and expensive: surrogate-based optimization (SBO)**

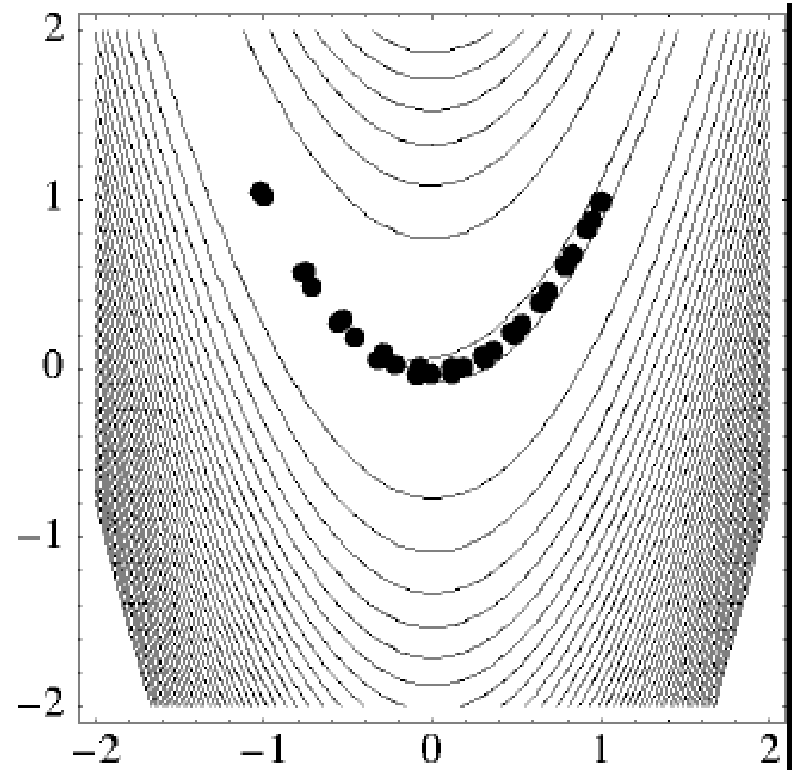
Nonlinearly-constrained problems:

- **Smooth and cheap: gradient-based methods**
- **Smooth and expensive: gradient-based methods**
- **Nonsmooth and cheap: non-gradient methods w/ penalty functions, SBO**
- **Nonsmooth and expensive: SBO**

Sandia National Laboratories

Examples of Various Optimization Methods

- Gradient-based optimization method
- Started at (x1,x2) = (-1.0, 1.2)
- Search algorithm follows the general descent direction "around the bend" of the Rosenbrock function.
- Gradient-based optimization is very efficient: ~30-100 evaluations of the f-values needed to find the minimum.
  - Performance varies between different algorithms



Sandia National Laboratories

Example Input/Output: Optimization

```
# DAKOTA Users Manual Ch 2 example problems
# using Rosenbrock's function

strategy,                              \
        single_method                  \
        tabular_graphics_data

method,                                \
          optpp_q_newton               \
          max_iterations = 50          \
          convergence_tolerance = 1e-4 \

model,                                 \
        single

variables,                             \
        continuous_design = 2          \
          cdv_initial_point    0.0   0.0     \
          cdv_lower_bounds    -2.0  -2.0     \
          cdv_upper_bounds     2.0   2.0     \
          cdv_descriptors     'x1'  'x2'     \

interface,                             \
       system                          \
          analysis_driver = 'rosenbrock'     \
           parameters_file = 'params.in'     \
           results_file    = 'results.out'

responses,                             \
        num_objective_functions = 1    \
        analytic_gradients             \
        no_hessians
```

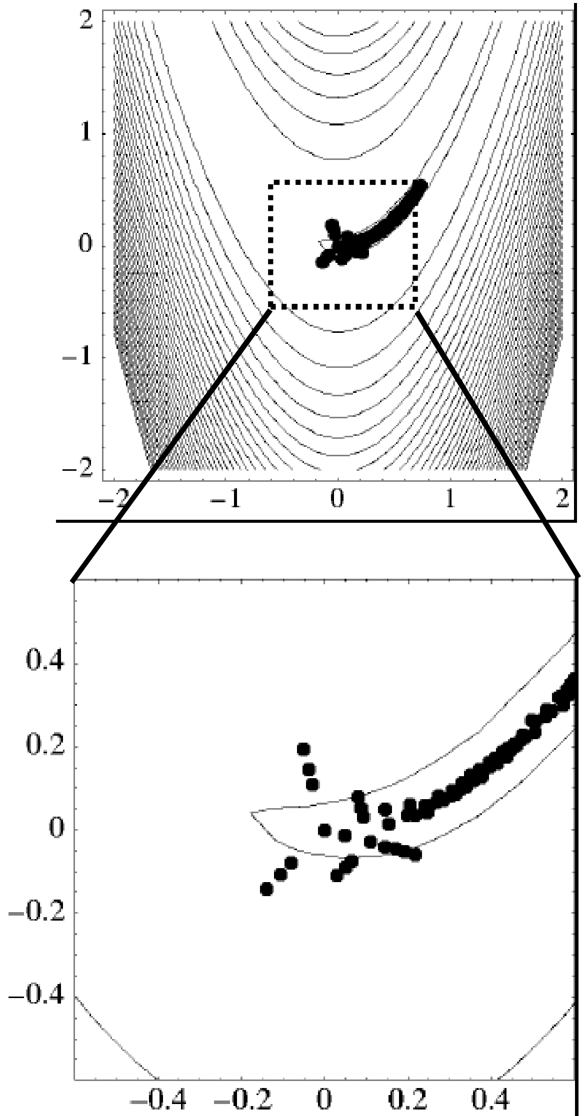
Optimization Input

```
------------------------------
Begin Function Evaluation   39
------------------------------
Parameters for function evaluation 39:
                      1.0013529993e+00 x1
                      1.0026805414e+00 x2

(rosenbrock /tmp/file3O5jSa /tmp/fileN42ob5)

Active response data for function evaluation 39:
Active set vector = { 2 }
 [  1.3635895231e-02 -5.4575642309e-03 ] obj_fn gradient

<<<<< Iterator optpp_q_newton completed.
<<<<< Function evaluation summary: 39 total (39 new, 0
duplicate)
<<<<< Best parameters          =
                      1.0013529993e+00 x1
                      1.0026805414e+00 x2
<<<<< Best objective function  =
                      1.9050696347e-06
<<<<< Best data captured at function evaluation 38
<<<<< Single Method Strategy completed.
DAKOTA execution time in seconds:
  Total CPU        =       0.05
```

Optimization Output

Examples of Various Optimization Methods

- **Pattern search optimization (non-gradient method)**
- **Started at (x1,x2) = (0,0)**
- **Search algorithm has made some progress toward the minimum after generating ~2000 f-values, but still not converged to the minimum.**
- **Pattern search is a great method, just not for this type of optimization problem.**

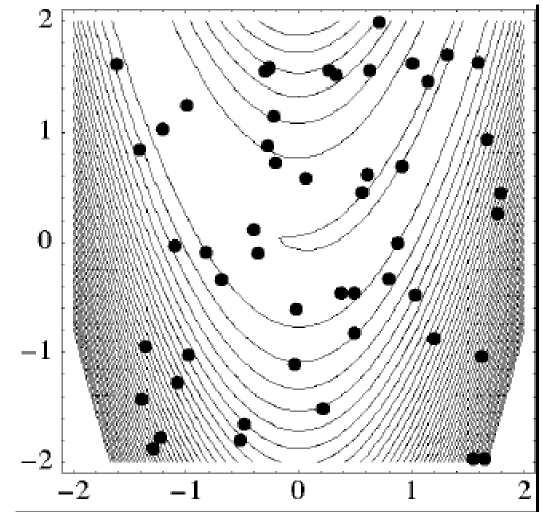


Sandia National Laboratories

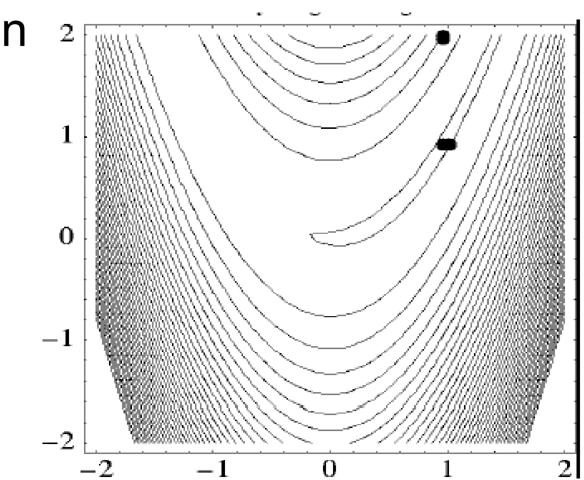
Examples of Various Optimization Methods

- **Genetic algorithm (GA) optimization (non-gradient method)**
- **Started with 50 random points in the parameter space**
- **GA search algorithm run to generate 10,000 f-values. 46 of the 50 samples have settled close to the true optimum**
- **GA is a great method, just not for this type of optimization problem.**

Initial population (50 random samples)



Final population (46 of 50 near minimum)



Sandia National Laboratories

Summary on Optimization Methods

- Selecting the right optimization method that matches the particular attributes of your problem is critical, especially if your simulation code is expensive!

- You won't have a good idea of the best optimization method UNLESS you perform some local and global sensitivity studies BEFORE you start optimizing.
  - Many users are blinded by past experience:
    - "I used method XYZ in grad school, therefore...."
    - "I read about genetic algorithms in the Porcelain Press, so they must be good...."

Uncertainty Quantification Methods

- **UQ methods in DAKOTA include:**
  - **DACE methods**
  - **Sampling methods**
  - **Analytic reliability methods**
  - **Epistemic uncertainty methods**

- **Sampling methods have two main uses:**
  - **Sensitivity analysis studies when the uncertain parameters only have bounds (intervals).**
  - **Statistical studies when the uncertain parameters have well-established probability distributions.**

DACE Methods

- Design of Computer Experiments
  - Usually, we do not assume distributional forms for the inputs
  - DDACE – Developed at SNL-CA (Monica Martinez-Canales)
    - Orthogonal arrays
    - Central Composite
    - Box-Behnken
    - Grid sampling
    - LHS and pure MC
    - Orthogonal LHS
    - Can calculate main effects for OAs
    - Can use in Variance Based Decomposition, Quality Metrics
  - FSUDACE – Developed by Florida State University, Max Gunzburger and John Burkardt
    - Halton sequences
    - Hammersley sequences
    - Centroidal Voronoi tesselation
    - Can "Latinize" these methods
    - Can use in Variance Based Decomposition, Quality Metrics
    - Fair amount of control in terms of where you want to start the sequence, what prime bases are used, etc.

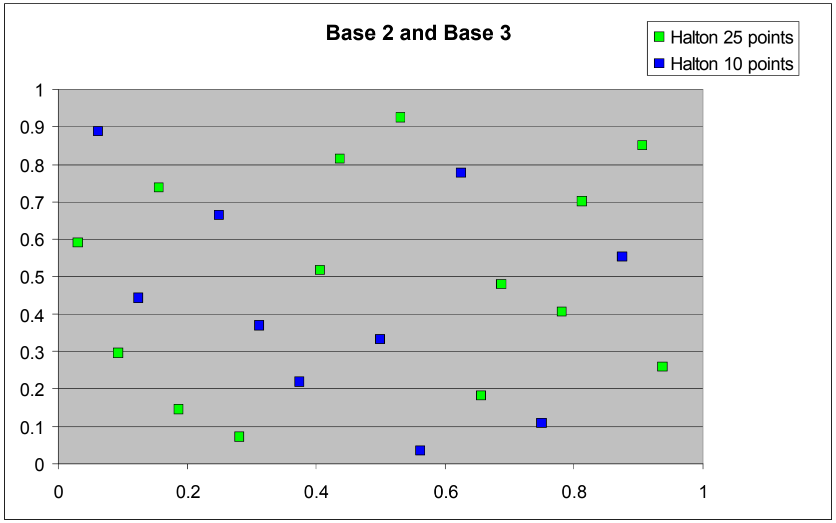
Quasi Monte Carlo Methods

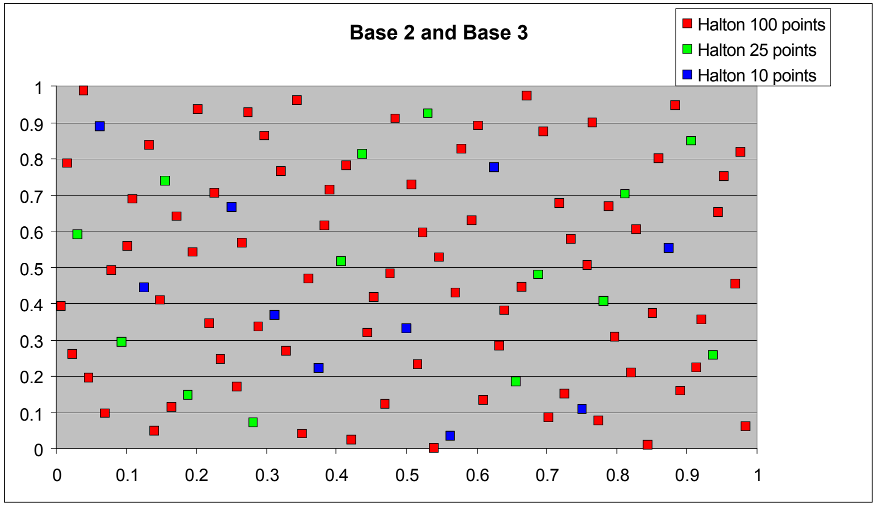
- Quasi-Monte Carlo sequences are deterministic sequences determined by a series of prime bases. They are designed to produce uniform random numbers on the interval [0,1].
- E.g., Halton sequence:

Sample Number	Base 2	Base 3	Base 5	Base 7
1	0.5000	0.3333	0.2000	0.1429
2	0.2500	0.6667	0.4000	0.2857
3	0.7500	0.1111	0.6000	0.4286
4	0.1250	0.4444	0.8000	0.5714
5	0.6250	0.7778	0.0400	0.7143
6	0.3750	0.2222	0.2400	0.8571
7	0.8750	0.5556	0.4400	0.0204
8	0.0625	0.8889	0.6400	0.1633
9	0.5625	0.0370	0.8400	0.3061
10	0.3125	0.3704	0.0800	0.4490

Sandia National Laboratories

Example: Halton Set



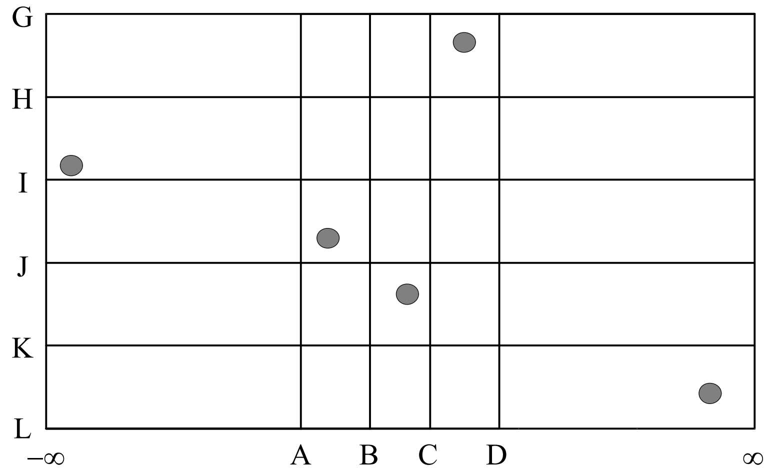


Quasi Monte Carlo Methods

- The quasi-Monte Carlo (QMC) and Centroidal Voronoi Tesselation (CVT) methods are designed with the goal of low discrepancy.
- Discrepancy refers to the nonuniformity of the sample points within the hypercube. Discrepancy is defined as the difference between the actual number and the expected number of points one would expect in a particular set B (such as a hyper-rectangle within the unit hypercube), maximized over all such sets.
- Low discrepancy sequences tend to cover the unit hypercube reasonably uniformly.
- CVT does very well volumetrically, however the lower-dimension (such as 1-D) projections of CVT can have high discrepancy.
- CVT developed as part of unstructured meshing techniques for irregular domains
- Centroidal Voronoi diagrams subdivide arbitrarily shaped domains into arbitrary numbers of nearly uniform subvolumes
- Sample points returned are the centers of the Voronoi region
- CVT good choice for high dimensional sampling

Sampling Methods

– **Latin Hypercube Sampling**
  - Distribution types supported: normal, lognormal, uniform, loguniform, triangular, gamma, gumbel, frechet, weibull, histogram, interval
  - Also can specify plain MC
  - Correlations between inputs supported with Iman and Conover's restricted pairing algorithm



A Two-Dimensional Representation of One Possible LHS of size 5 Utilizing X1 (normal) and X2 (uniform)

Variance Based Decomposition

$$V(y) = \sum_{i=1}^{n} V_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} V_{ij} + \cdots + V_{12\cdots n},$$

$V_i = V(E(Y| X_i))$,
$V_{ij} = V(E(Y| X_i, X_j) - V_i - V_j$
$V_{ijk} = V(E(Y| X_i, X_j, X_k) - V_i - V_j - V_k - V_{ij} - V_{jk} - V_{ik}$

$$S_j = \frac{V(E(Y \mid X_j))}{V(Y)}$$

- Variance-based decomposition methods involve Monte Carlo sampling of the “inner loop” to calculate $E(Y| X_i = x^*)$ and additional Monte Carlo sampling of the “outer loop” to calculate the variance $V(E(Y| X_i))$. This is very expensive in terms of number of function evaluations.
- Various methods (Saltelli, McKay replicated LHS, Morris, FAST) have been developed to calculate SA indices.
- Advantage of VBD over correlation analysis is that VBD better captures non-monotonic relationships.

Sandia National Laboratories

Example Input/Output: LHS

```
# DAKOTA Users Manual Ch 2 example problems
# using Rosenbrock's function

strategy,                                   \
        single_method                       \
        tabular_graphics_data

method,                                     \
        nond_sampling                       \
        samples = 200 seed = 1734           \
        sample_type lhs                     \
        response_levels = 100.0

model,                                      \
        single

variables,                                  \
      normal_uncertain = 2                  \
        nuv_means  0.0  0.0      \
        nuv_std_deviations  1.0   2.0       \
        nuv_descriptor         'x1'  'x2'

interface,                                  \
        system                              \
          analysis_driver = 'rosenbrock'          \
          parameters_file = 'params.in'           \
          results_file    = 'results.out'

responses,                                  \
        num_response_functions = 1          \
        no_gradients                        \
        no_hessians
```

Sample Input

```
Statistics based on 200 samples:

Moments for each response function:
response_fn_1:  Mean = 6.68699e+02  Std. Dev. = 1.01401e+03  Coeff. of
Variation = 1.51639e+00

95% confidence intervals for each response function:
response_fn_1:  Mean = ( 5.27307e+02, 8.10091e+02 ), Std Dev = (
9.23420e+02, 1.12446e+03 )

Probabilities for each response function:
Cumulative Distribution Function (CDF) for response_fn_1:
     Response Level  Probability Level  Reliability Index
     --------------  -----------------  -----------------
   1.0000000000e+02   3.0500000000e-01

Simple Correlation Matrix between input and output:
                     x1           x2 response_fn_1
          x1  1.00000e+00
          x2  3.69447e-03  1.00000e+00
response_fn_1 -1.23484e-01 -3.35515e-01  1.00000e+00

Partial Correlation Matrix between input and output:
             response_fn_1
          x1 -1.29767e-01
          x2 -3.37645e-01

Simple Rank Correlation Matrix between input and output:
                     x1           x2 response_fn_1
          x1  1.00000e+00
          x2 -3.09488e-02  1.00000e+00
response_fn_1  1.85300e-02 -3.03560e-01  1.00000e+00

Partial Rank Correlation Matrix between input and output:
             response_fn_1
          x1  9.59218e-03
          x2 -3.03183e-01
```

Sample Output

Sandia National Laboratories

UQ with Reliability Methods

Mean Value Method

$$\mu_g = g(\mu_{\mathbf{x}})$$

$$\sigma_g = \sum_i \sum_j Cov(i,j) \frac{dg}{dx_i}(\mu_{\mathbf{x}}) \frac{dg}{dx_j}(\mu_{\mathbf{x}})$$

$$\bar{z} \rightarrow p, \beta \begin{cases} \beta_{cdf} = \dfrac{\mu_g - \bar{z}}{\sigma_g} \\ \beta_{ccdf} = \dfrac{\bar{z} - \mu_g}{\sigma_g} \end{cases}$$

$$\bar{p}, \bar{\beta} \rightarrow z \begin{cases} z = \mu_g - \sigma_g \bar{\beta}_{cdf} \\ z = \mu_g + \sigma_g \bar{\beta}_{ccdf} \end{cases}$$

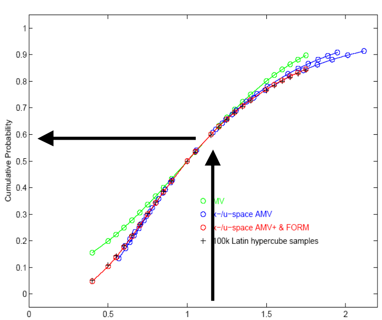
Rough statistics

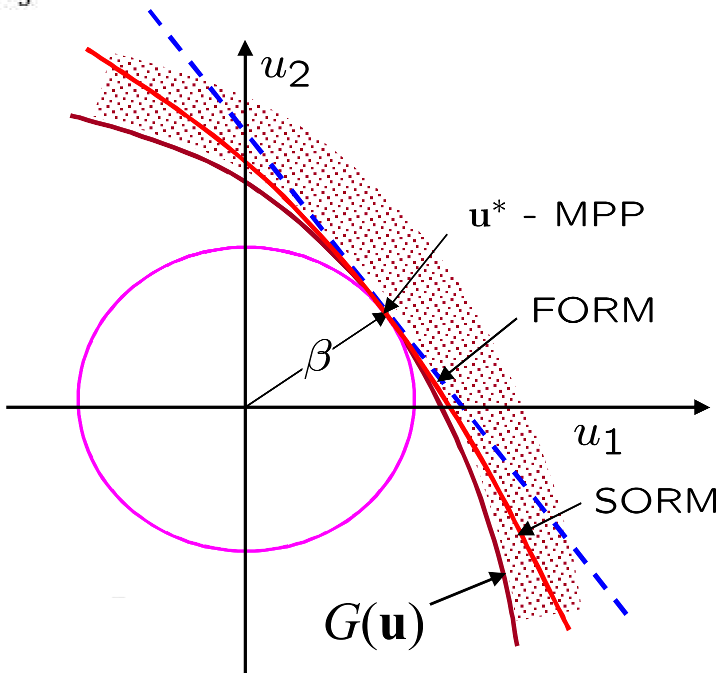
MPP search methods

Reliability Index Approach (RIA)

$$\text{minimize} \quad \mathbf{u}^T \mathbf{u}$$
$$\text{subject to} \quad G(\mathbf{u}) = \bar{z}$$

Find min dist to *G* level curve
Used for fwd map *z* → *p*/*β*

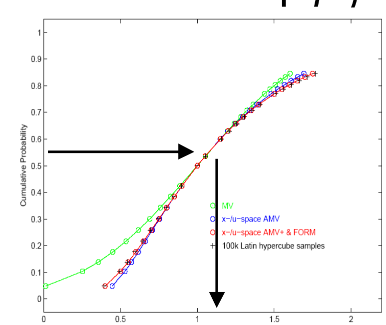




Performance Measure Approach (PMA)

$$\text{minimize} \quad \pm G(\mathbf{u})$$
$$\text{subject to} \quad \mathbf{u}^T \mathbf{u} = \bar{\beta}^2$$

Find min *G* at *β* radius
Used for inv map *p*/*β* → *z*

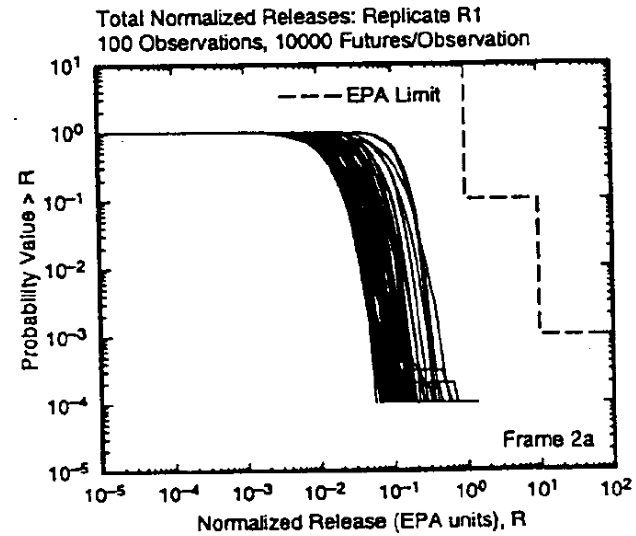


Nataf x → u: $\Phi(z_i) = F(x_i)$
$\mathbf{z} = \mathbf{L}\mathbf{u}$

Epistemic UQ

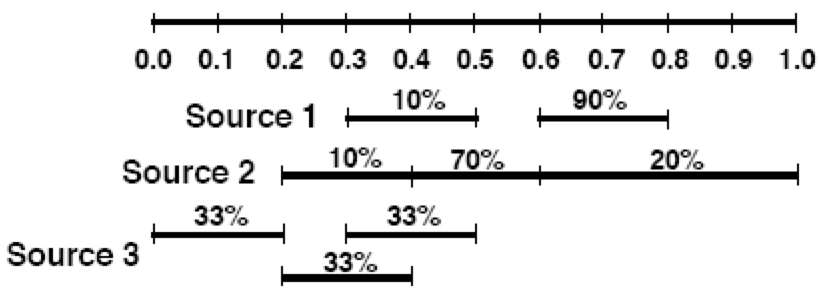
Second-order probability

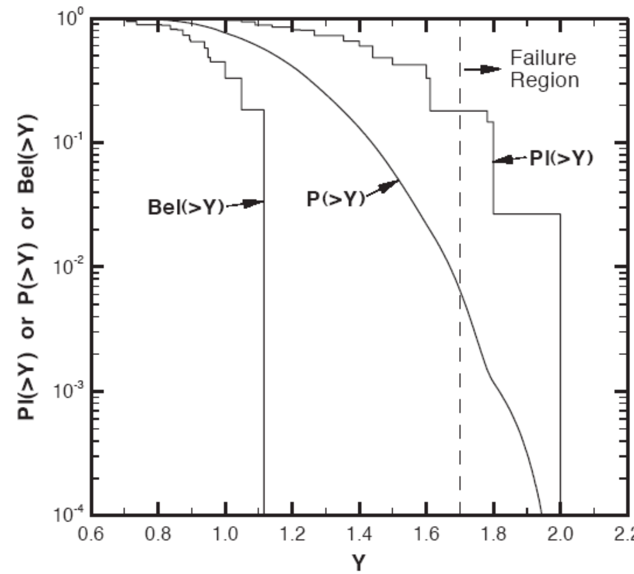
- Two levels: distributions/intervals on distribution parameters
- Outer level can be epistemic (e.g., interval)
- Inner level can be aleatory (probability distrs)
- Strong regulatory history (NRC, WIPP).



Dempster-Shafer theory of evidence

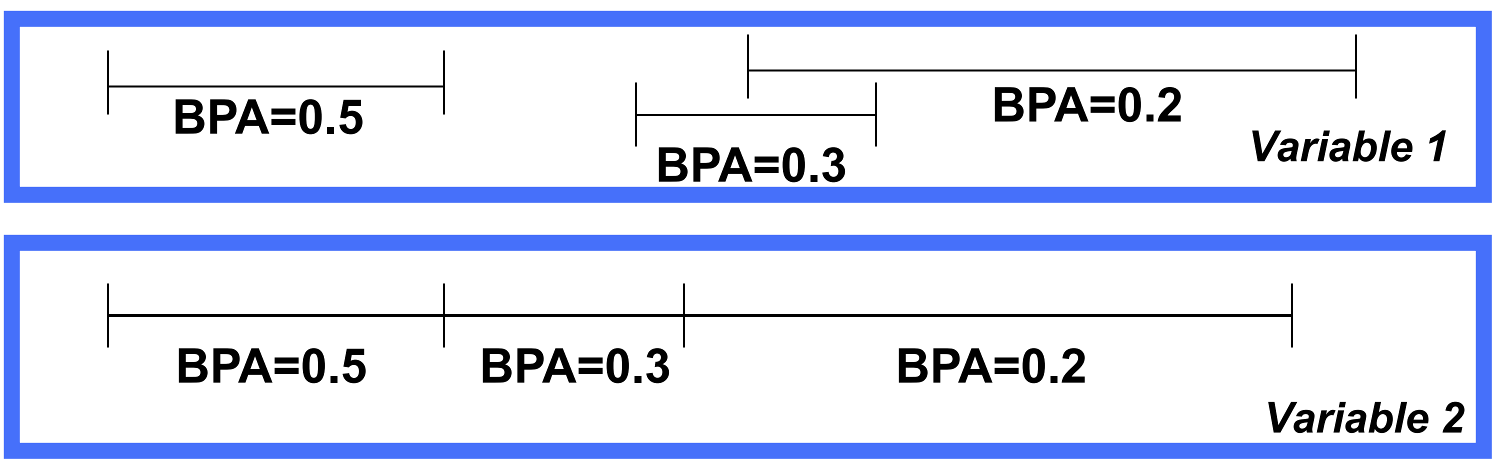
- Basic probability assignment (interval-based)
- Solve opt. problems (currently sampling-based) to compute belief/plausibility for output intervals





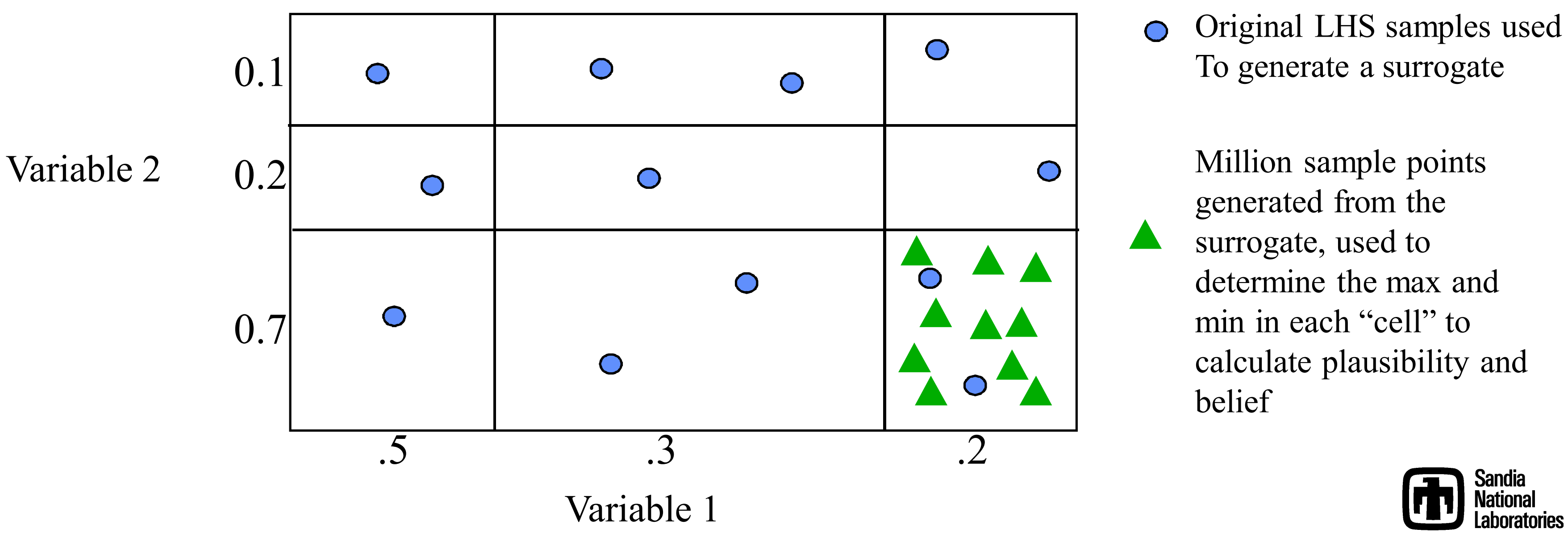
Epistemic Uncertainty Quantification

- Epistemic uncertainty refers to the situation where one does not know enough to specify a probability distribution on a variable
- Sometimes it is referred to as subjective, reducible, or lack of knowledge uncertainty
- The implication is that if you had more time and resources to gather more information, you could reduce the uncertainty
- Initial implementation in DAKOTA uses Dempster-Shafer belief structures. For each uncertain input variable, one specifies “basic probability assignment” for each potential interval where this variable may exist.
- Intervals may be contiguous, overlapping, or have “gaps”



Epistemic Uncertainty Quantification

- **Look at various combinations of intervals. In each joint interval "box", one needs to find the maximum and minimum value in that box (by sampling or optimization)**
- **Belief is a lower bound on the probability that is consistent with the evidence**
- **Plausibility is the upper bound on the probability that is consistent with the evidence**
- **Order these beliefs and plausibility to get CDFs**
- **Draws on the strengths of DAKOTA**
  - **Requires surrogates**
  - **Requires sampling and/or optimization for calculation of plausibility and belief within each interval "cell"**
  - **Easily parallelized**



UQ Needs for the next 5 years

- Quantify "extrapolative" confidence
  - Requires the use of response surface methods which handle uncertainty
- Sampling for stochastic processes
  - Sampling of random fields (in space and/or time, possibly non-stationary and non-Gaussian), not just random variables.
- Intrinsic / Analytic UQ capability
  - Expand the role of expansion methods such as Polynomial chaos
  - Many issues remain about the set of points on which to construct the basis for different distribution types, the type of integration method, etc.
- Efficient (e.g. surrogate) methods for higher order moments and tail statistics
  - Better quantification of surrogate accuracy
- Adaptive Experimental Design
  - Importance Sampling, Adaptive OAs
- Efficient sensitivity analysis
- Epistemic UQ
  - Capability to combine aleatory and epistemic uncertainty in one analysis
- UQ treatment in multi-fidelity and/or hierarchical models
  - Efficiency issue
  - More important, dealing with uncertainty at different time or length scales across simulations

DAKOTA Team Contact Info

- Web site:
  - http://endo.sandia.gov/DAKOTA
  - Email: dakota@sandia.gov
  - User's Manual, Reference Manual, Developers Manual - online

- Team Members
  - Mike Eldred, Principal Investigator (R&D)
  - Tony Giunta, Product Manager (applications & training)
  - Shane Brown, Support Manager (software issues)
  - Laura Swiler
  - Brian Adams
  - Danny Dunlavy
  - Dave Gay
  - Bill Hart
  - Jean-Paul Watson
  - Many other technical contributors (SNL-CA, SNL-NM, academia,...)
  - Scott Mitchell, Dept. 1411 manager
  - Marty Pilch, Dept. 1533 manager

Parting Thoughts....

Recipe for a UQ Study via DAKOTA

UQ Study*

- Define appropriate parameter ranges and/or probability distributions.
- Use the Latin hypercube sampling method to generate Order(N to N^2) samples of the x-values.
  - The minimum # samples you can get away with is N+1
  - One rule of thumb is: # samples >= 0.5*(N+1)*(N+2)
  - Another rule of thumb is: ~15-30 samples for each N (i.e., # samples = 15*N or 30*N)
  - Another rule of thumb is: # samples = 1/3 of your simulation run budget, so that you can save the other 2/3 runs for follow-on studies

- Examine the f-value correlation data and basic statistical data generated by DAKOTA.
  - These correlations are the "global" linear trends in the f-values.
  - Often they are useful in finding a worst-case or best-case combination of x-values.

- Perform a more detailed statistical analysis of the f-values (contact me or a SNL stats expert for pointers).

- ***Only use DAKOTA's f-value probability estimates if you have well-founded knowledge about the probability distributions on the x-values.***

*Note: There are other more advanced UQ methods (e.g., Dempster-Shafer theory, polynomial chaos theory) in DAKOTA, but these are yet ready for "production" use.

Sandia National Laboratories

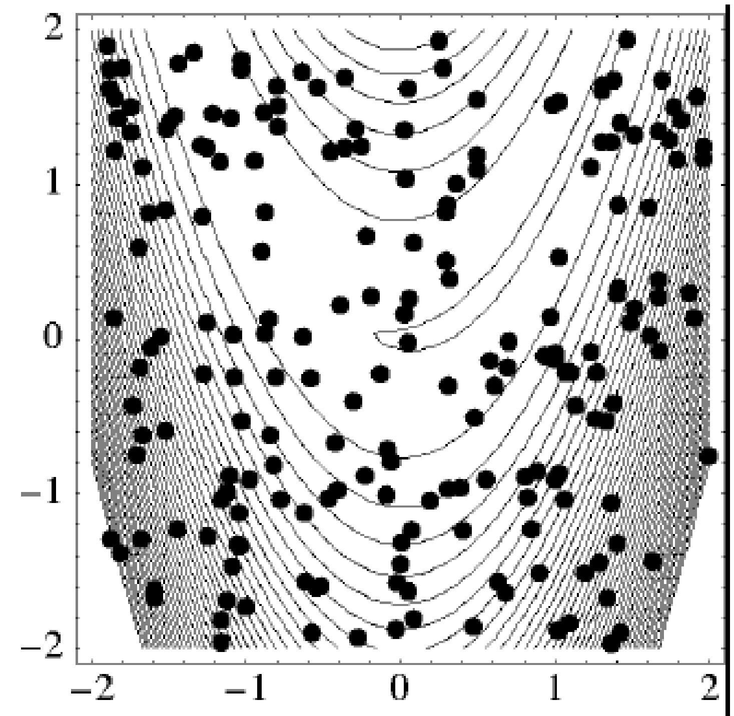
Statistical Analysis via UQ Methods

- The UQ statistical analysis leverages the fact that we know the probability distributions of the design parameters.
- Same approach as with the sensitivity analysis:
  - Assign specific probability distributions to the design parameters in the dakota input file
  - Select 0.3*K samples (see "recipe" guidelines) via DAKOTA's Latin hypercube sampling method
  - Run DAKOTA and examine the following:
    - Correlation data for f-values to x-values
    - Minimum and maximum f-values
    - Probability data on f-values, e.g., Prob(f>f_critical)
  - Repeat the LHS study with another 0.3*K samples

Sandia National Laboratories

Sensitivity Analysis via UQ Methods

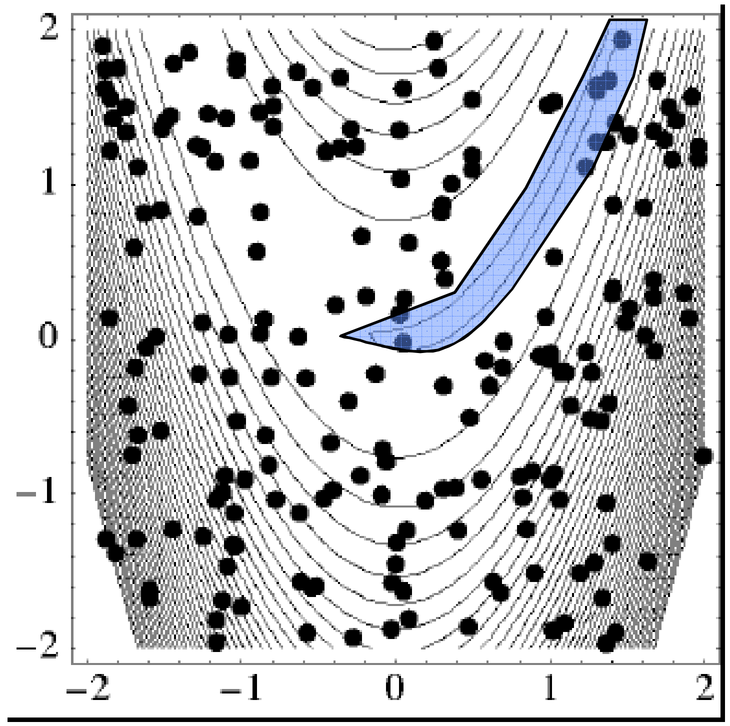
- **200 random samples in the parameter space:**
  - **-2 <= x1 <= 2**
  - **-2 <= x2 <= 2**
  - **** We only know the bounds, and not any probabilities on x1 and x2.***
- **DAKOTA produces a correlation matrix with data on:**
  - **Correlation of f with x1**
  - **Correlation of f with x2**
  - **Correlation of x1 with x2 (should be nearly zero!)**
- **DAKOTA produces a column oriented output data file: 200 rows by 3 columns; the columns are x1, x2, and f**
  - **Analyze this data file in a commercial statistics software package to get min, max, and trend data**



Sandia National Laboratories

Statistical Analysis via UQ Methods

- **200 random samples in the parameter space:**
  - **-2 <= x1 <= 2**
  - **-2 <= x2 <= 2**
  - *** But now we are told that x1 and x2 have uniform probability distributions**
- **DAKOTA produces a correlation matrix with data on:**
  - **Correlation of f with x1**
  - **Correlation of f with x2**
  - **Correlation of x1 with x2 (should be nearly zero!)**
- **DAKOTA produces statistics on the f-values (e.g., Prob(f<4) = 0.049)**
- **DAKOTA produces a column oriented output data file: 200 rows by 3 columns; the columns are x1, x2, and f**
  - **Analyze this data file in a commercial statistics software package to get min, max, etc.**



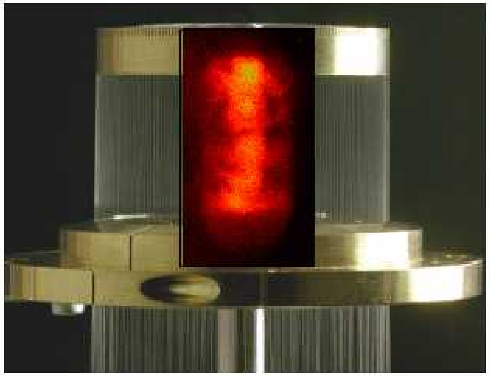
Sandia National Laboratories

DAKOTA Applications

Typical SNL DAKOTA Applications

Design Optimization - Use numerical optimization methods to find the best design.

- Load spreader plate design with Pronto
- Weapon component design with Salinas
- **F-35 fuel tank design (with Lockheed-Martin)**

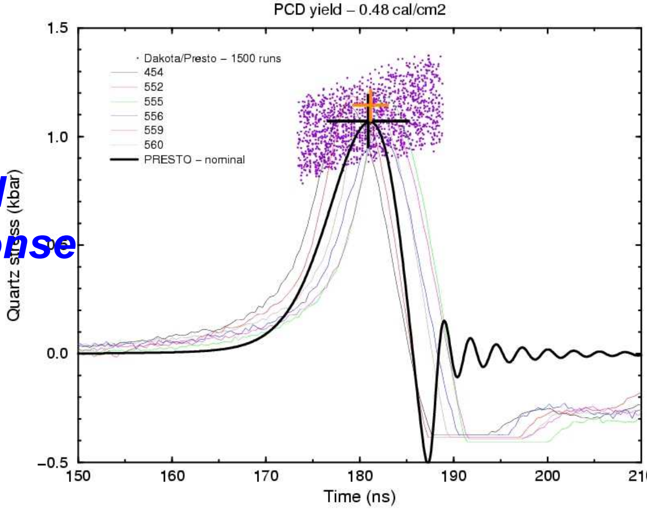


Parameter Estimation – Use numerical optimization methods to calibrate computer models to match experimental test data.

- Heterogeneous material parameter identification w/ Presto
- **Hypervelocity flyer plate calibration w/ ALEGRA**
- Molecular force field parameter identification w/ Towee
- Material parameter identification w/ CTH

Uncertainty Quantification – Propagate variability and uncertainty on code inputs & compute output response probabilities.

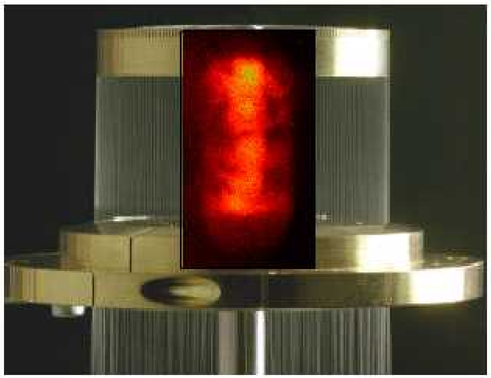
- Circuit transient response w/ Xyce
- Thermal environment uncertainties w/ CALORE
- **Material stress-strain response w/ Presto**
- **Underground target response modeling**



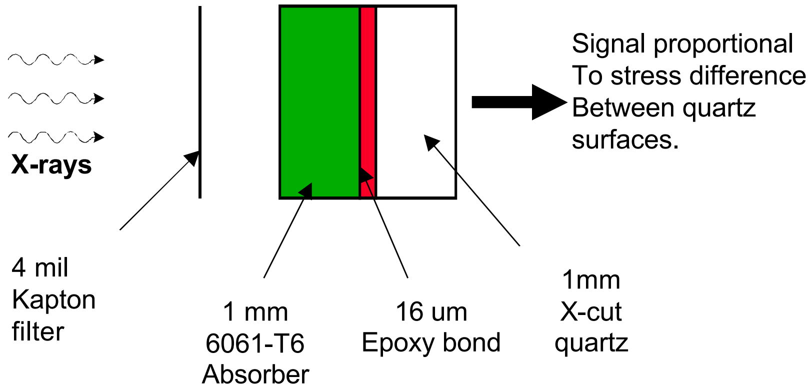
Sandia National Laboratories

DAKOTA UQ Study:
Presto Simulations vs. Z-Accelerator Data

Tungsten wire array & Z pinch

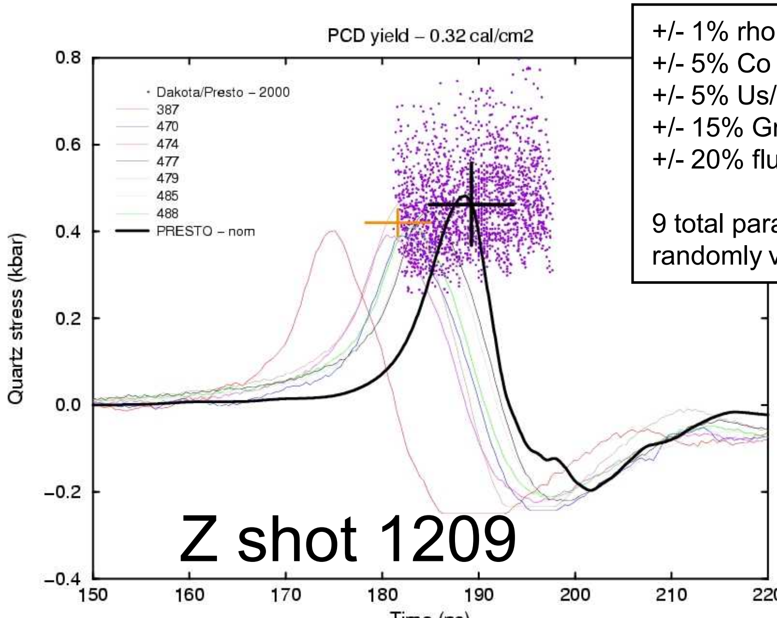


X-Ray Induced Thermomechanical Shock Modeled w/ Presto



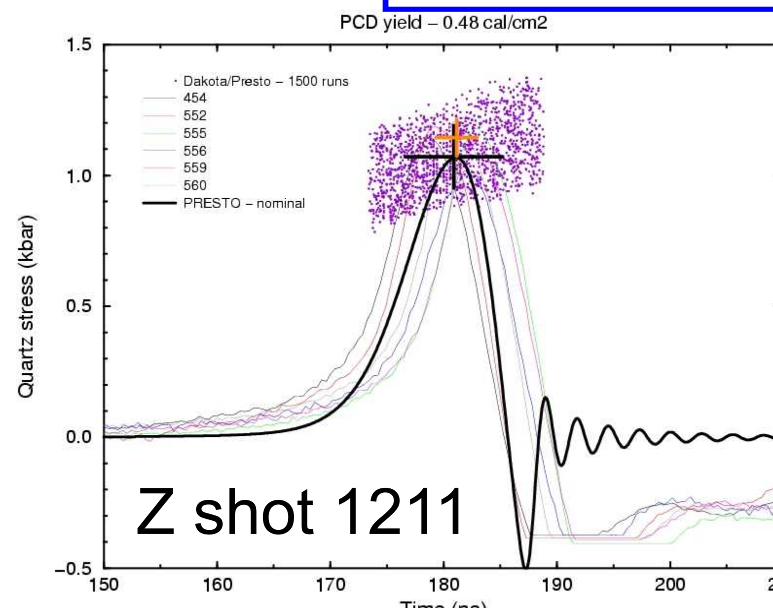
<u>Summary:</u>

- UQ study on Presto sims. of thermomechanical shock.
- DAKOTA generated 2000 Presto runs; run on 1500's network of workstations
- Compared Presto vs. Z Shot μ±1σ uncertainty bands.
- First-ever UQ study gives info on design margins. Need for Presto model improvement was identified.
- Contacts: Tony Giunta, 1533 & Joel Lash, 1514



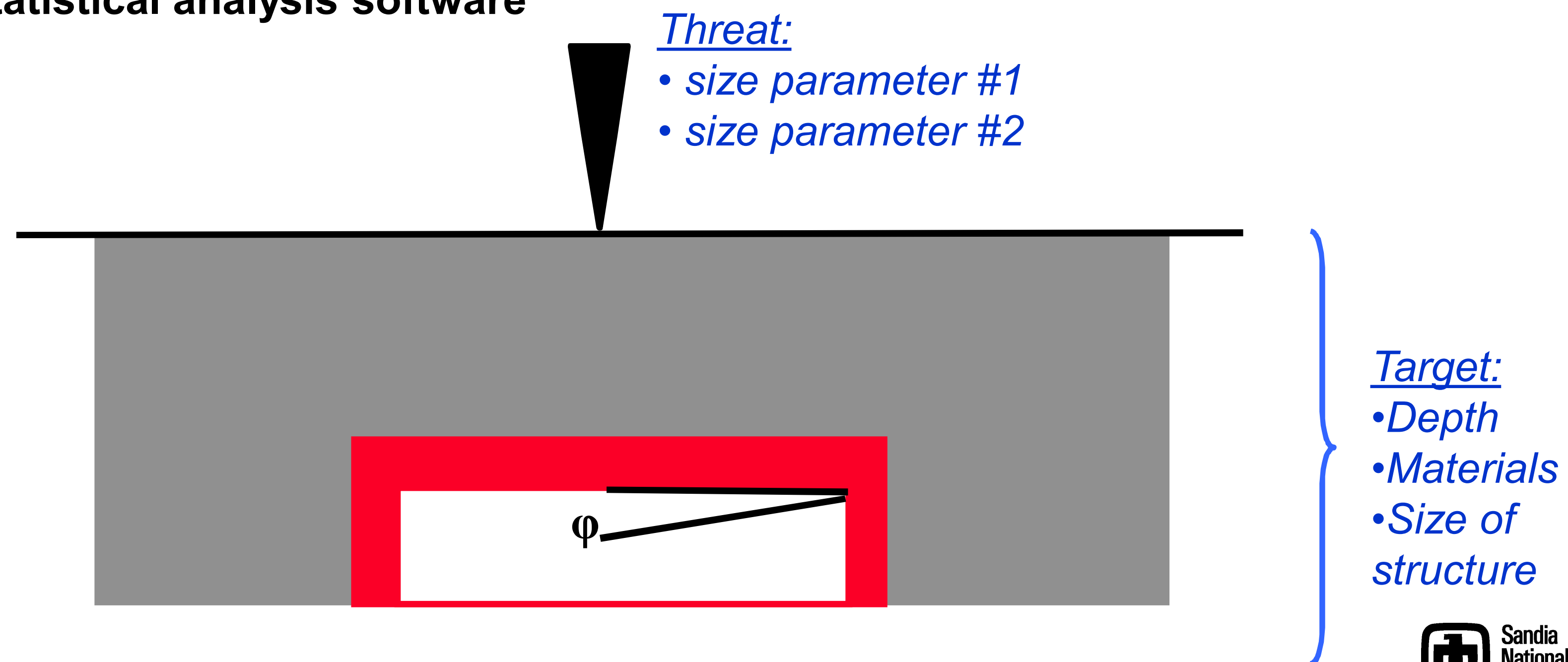
+/- 1% rho
+/- 5% Co
+/- 5% Us/Up slope
+/- 15% Gruneisen
+/- 20% fluence

9 total parameters randomly varied



DAKOTA UQ Study: Underground Target Defeat (1 of 4)

- **Scenario: underground target with an external threat**
- **Goal: Assess uncertainty in target response due to uncertainties in target construction and threat characteristics**
- **9 parameters that describe target & threat uncertainty**
  - **Each parameter has uncertainty specified by an interval**
- **Response: deflection angle (φ) of target roof at mid-span**
- **Tools: Sandia shock physics code; DAKOTA UQ/optimization tool; JMP statistical analysis software**

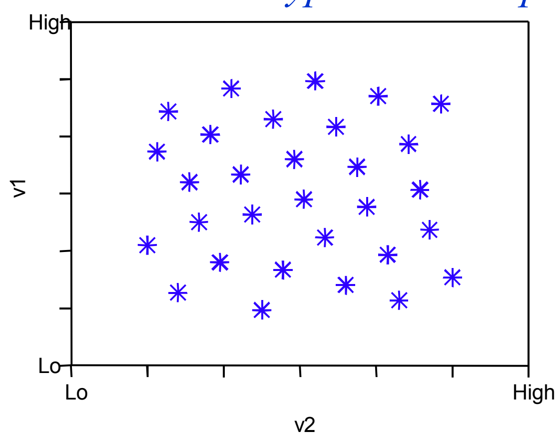


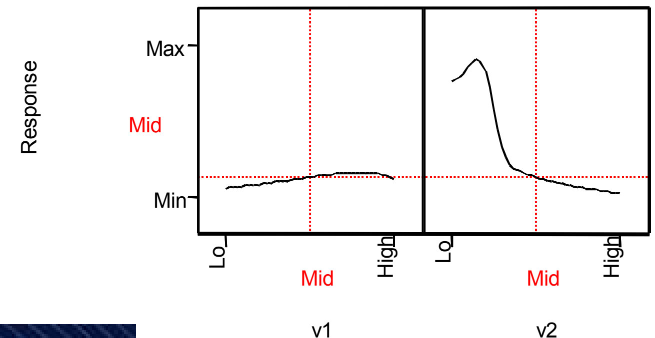
Sandia National Laboratories

DAKOTA UQ Study: Underground Target Defeat (2 of 4)

- **Example: 2-parameter sampling via DAKOTA; simulations run on a Linux cluster; data analysis via JMP yields response trends vs parameters variations**

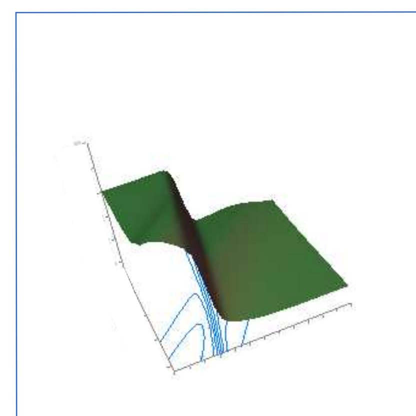
DAKOTA Latin Hypercube sampling





The Statistical Discovery Software

Statistical Analysis & Graphics



Sims run on a LINUX Cluster

Summary of Fit

RSquare	0.866299
RSquare Adj	0.823758
Root Mean Square Error	#.#
Mean of Response	#.#
Observations (or Sum Wgts)	30

Analysis of Variance

Source	DF	Sum of Square	Mean Square	F Ratio
Model	7	5343.9158	763.417	20.3638
Error	22	824.7562	37.489	Prob > F
C. Total	29	6168.6720		<.0001

DAKOTA UQ Study: Underground Target Defeat (3 of 4)

- **Actual Study:**
  - **9 parameters of interest**
  - **70 Latin Hypercube samples uniformly distributed in the 9-dimensional parameter space: 70 simulations run on various Sandia Linux clusters**
  - **Data analysis & visualization in JMP: stepwise regression on mixed 2nd & 3rd order polynomial models, plus neural network models.**

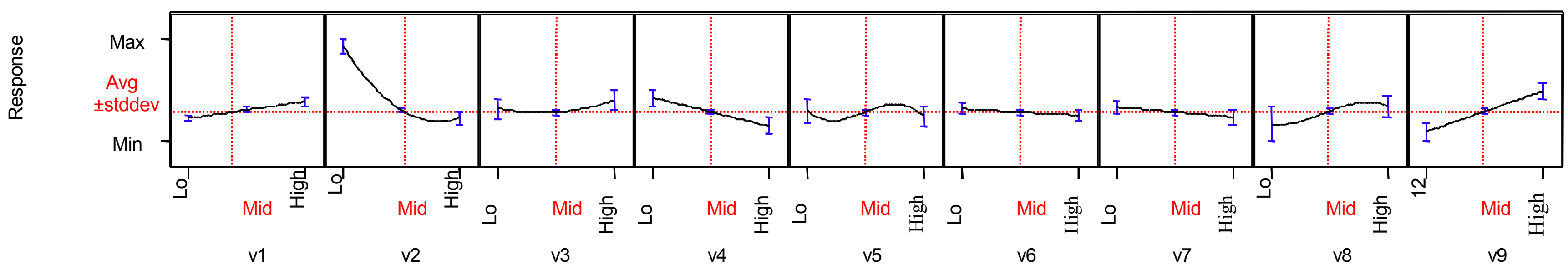
Summary of Fit

RSquare	0.848663
RSquare Adj	0.825564
Root Mean Square Error	###.###
Mean of Response	###.###
Observations (or Sum Wgts)	220

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Ratio
Model	29	36483.299	1258.04	36.7406
Error	190	6505.842	34.24	Prob > F
C. Total	219	42989.141		<.0001

- **Results from these 70 code runs motivated additional code runs concentrated in several [v1,v2] parameter subspaces.**
  - **150 additional code runs**
  - **220 total code runs**



DAKOTA UQ Study: Underground Target Defeat (4 of 4)

- Take home messages:
  - This statistical design/analysis approach yielded new insights:
    - Some parameters though to be important actually were not important
    - We found some multi-parameter interactions that were not obvious
  - We are exploiting the statistical tools in DAKOTA and in JMP to obtain greater insights out of our simulation runs.
  - These same statistical tools can be applied to many projects underway at SNL.

What’s New for DAKOTA version 4.0?

- **User Interface:**
  - New DAKOTA input file syntax (pros & cons)
  - New graphical user interface “JAGUAR”
  - New manuals (including a version 4.0 User’s Manual!!!)
- **Optimization:**
  - New non-gradient methods for both local and global optimization
  - General penalty function capability for handling constraints in non-gradient optimization methods
- **Uncertainty Quantification:**
  - Several new probability distribution types (triangular, beta, gamma, gumbel, frechet, etc.)
  - Dempster-Shafer theory method operational

Summary

- **DAKOTA Capabilities:**
  - Both time-tested and state-of-the-art methods for sensitivity analysis, optimization, and uncertainty quantification.
- **DAKOTA Software Support:**
  - Working to improve manuals & user support.
  - Lots of one-on-one training in FY05 and FY06
  - Working to pass on "philosophy" of sensitivity/Opt/UQ studies to SNL staff.
- **Goal:**
  - DAKOTA to become a widely used tool at Sandia, within the DOE/NNSA Tri-Laboratory community, and with key industrial partners.
    - Much progress, but more to be done

Sandia National Laboratories