SAND2012-8608P

DAKOTA Advanced Topics: Interfacing and Parallelism

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.

Basic Steps to Using DAKOTA

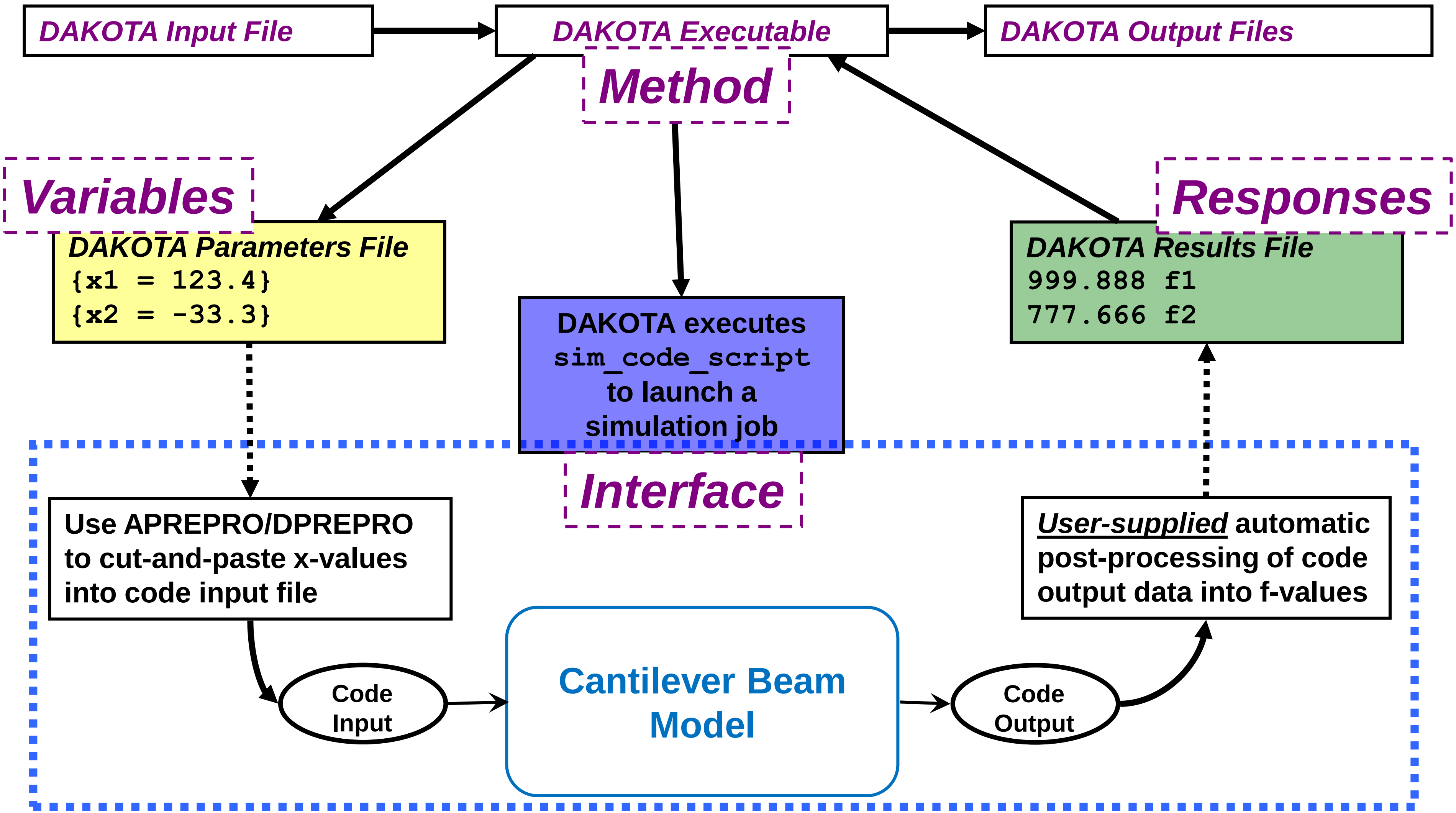
1. Define analysis goals; understand how DAKOTA helps and select a method to use
2. Access DAKOTA and understand help resources
3. **Workflow:** create an automated workflow so DAKOTA can communicate with your simulation (Advanced Topic)
   - Parameters to model, responses from model to DAKOTA
   - Typically requires scripting (Python, Perl, Shell, Matlab) or programming (C, C++, Java, Fortran)
   - Workflow usually crosscuts DAKOTA analysis types
4. **DAKOTA input file:** Jaguar GUI or text editor to configure DAKOTA to exercise the workflow to meet your goals
   - Tailor variables, methods, responses to analysis goals
5. Run DAKOTA: command-line; text input / output

Possible Directions

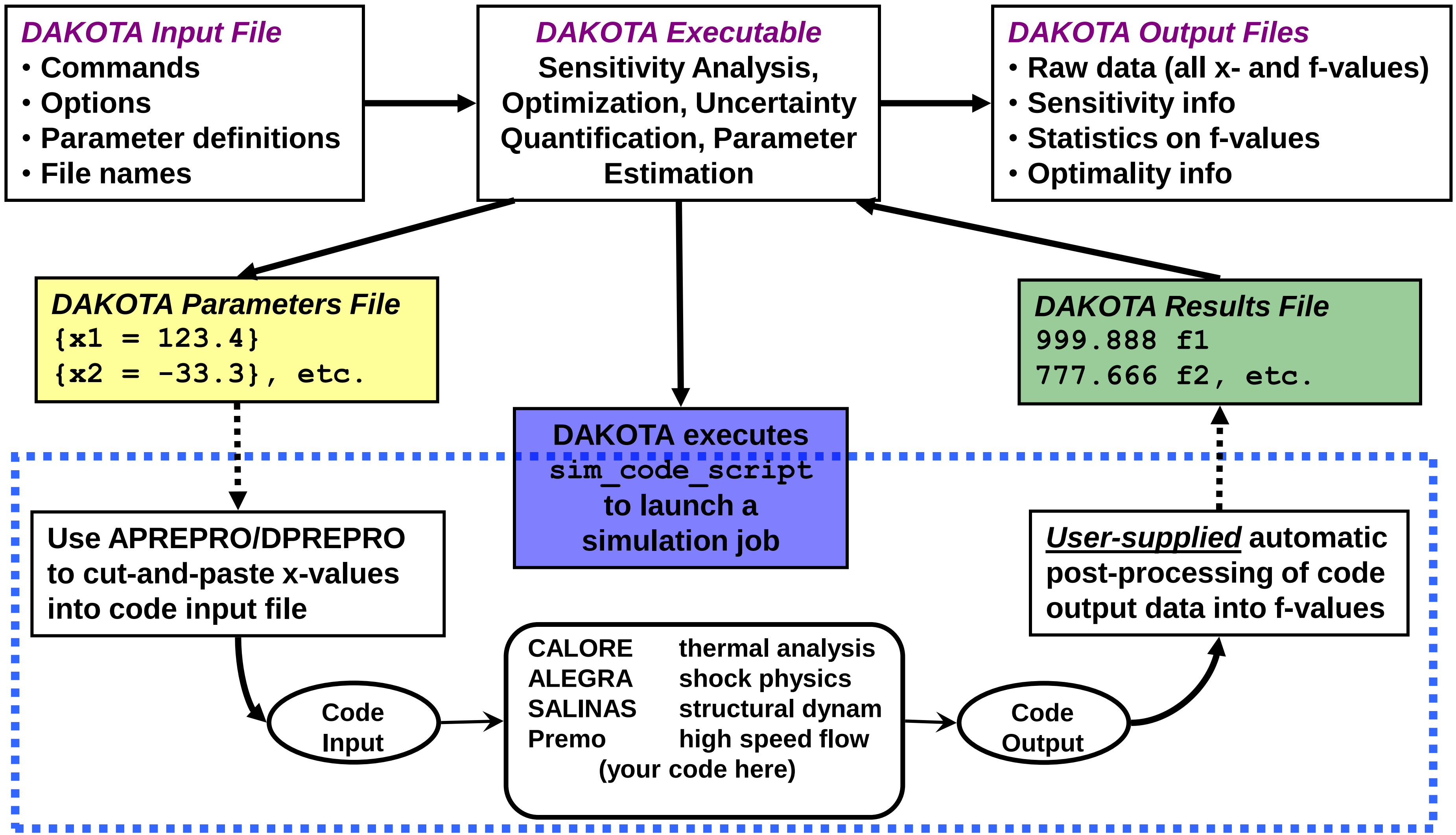
- See process of interfacing DAKOTA to a black-box application through file system
- See current state of DAKOTA library interface
- Understand MPI vs. local parallelism
- Understand modes of application parallelism (in queue, out of queue, serial, parallel apps)

- From DAKOTA 101:
  - Matlab, Python interfacing
  - DAKOTA as a library
  - Basics of HPC at SNL

Interface communicates through file system and user-supplied script



DAKOTA Execution & Info Flow



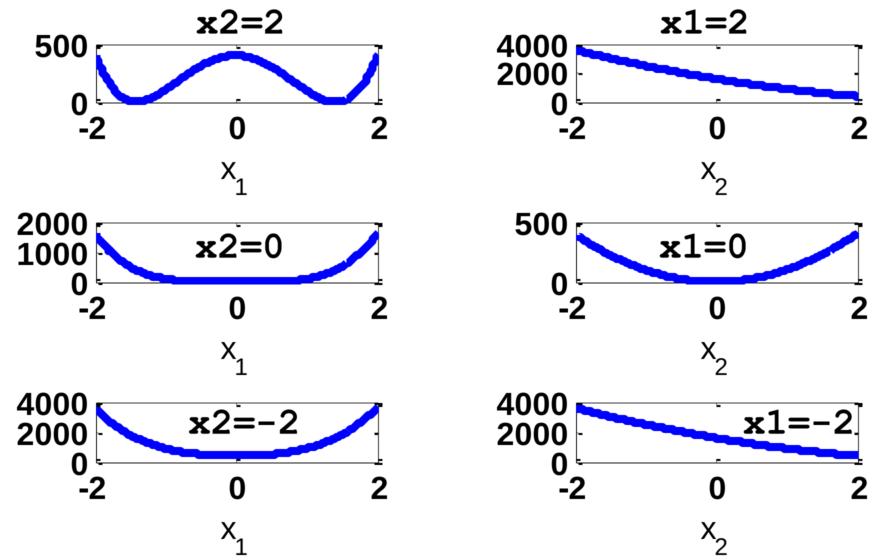
Application Stand-in: Rosenbrock "Banana" Function

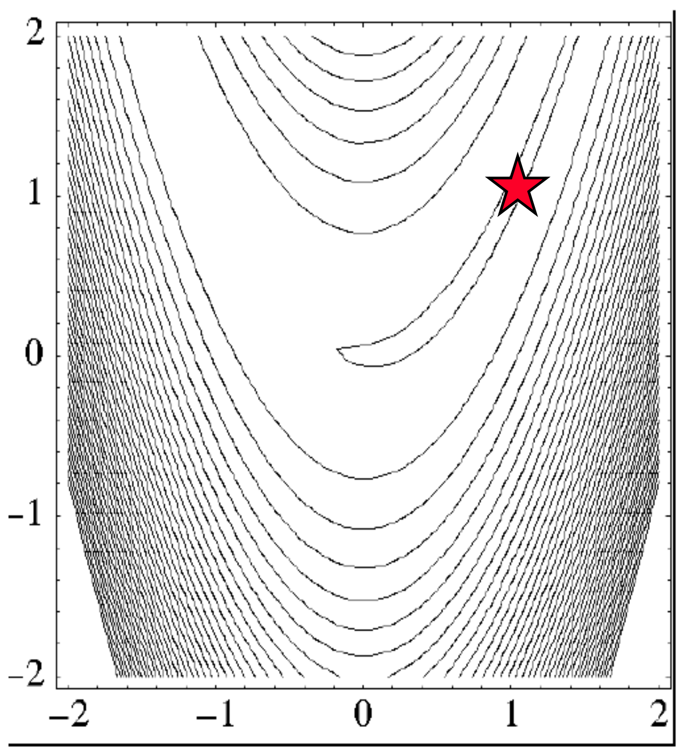
$f(x1,x2) = 100*(x2-x1*x1)^2 + (1-x1)^2$

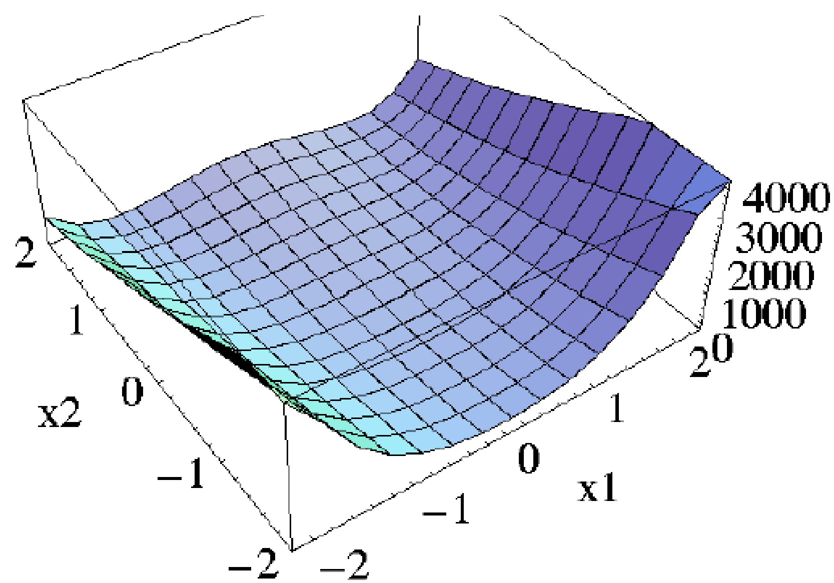
$-2 \le x1 \le 2$

$-2 \le x2 \le 2$

Minimum: $f(x1,x2) = f(1,1) = 0.0$







Demo: Rosenbrock as a "black box"

- Locate example in `Dakota/examples/script_interfaces/generic`
- Described in DAKOTA 5.2 User's Manual 18.1
- Explore top-down (DAKOTA down to application and back)
- Since you're familiar with your application, may want to build from application up

Interfacing to Your Simulation (Assuming Text-based I/O)

1. **Annotate your input file to create template**
   `{ stress }` `{ alpha1 }`
2. **Create a representative DAKOTA `params.in` file in aprepro format (see User's 11.6) and test:**
   `dprepro params.in analysis.in.template analysis.in`
3. **Verify commands to run application with `analysis.in`**
4. **Determine how to automatically extract results of interest (direct application to export, shell commands, python, perl, visual basic, etc.) to create `results.out` (see User's 13.2)**
5. **Assemble into a script, e.g., `run_analysis.sh`; test script with sample params.in:**
   `./run_analysis.sh params.in results.out`
6. **Test with a simple DAKOTA input deck, e.g., parameter study**

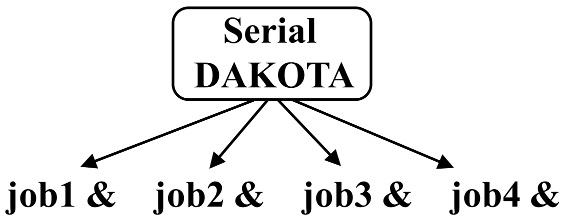
Parallelism

- **See Application Parallelism slides shipped in Dakota/examples/parallelism**

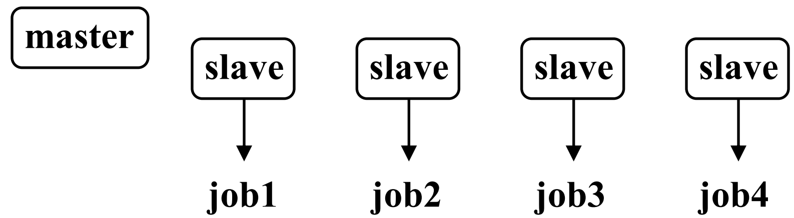
Parallelism from a computing platform perspective

Nested parallel models support large-scale applications and architectures.

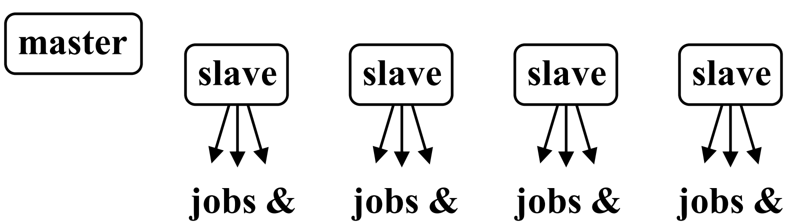
1. SMP/multiprocessor workstations: Asynchronous (external job allocation)



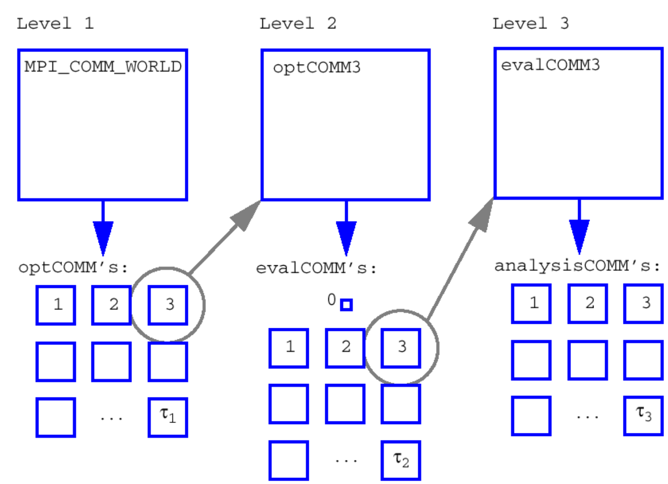
2. Cluster of workstations: Message-passing (internal job allocation)



3. Cluster of SMP's: Hybrid (service/compute model)



4. MPP: Internal MPI partitions (nested parallelism)



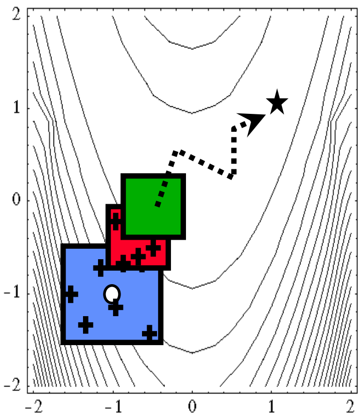
Parallelism from an algorithmic perspective

1. *Algorithmic coarse-grained parallelism*: independent fn. Evaluations performed concurrently:
   - Gradient-based (e.g., finite difference gradients, speculative opt.)
   - Nongradient-based (e.g., GAs, PS, Monte Carlo)
   - Approximate methods (e.g., DACE)
   - Concurrent-method strategies (e.g., parallel B&B, island-model GAs, OUU)
2. *Algorithmic fine-grained parallelism:* computing the internal linear algebra of an opt. algorithm in parallel (e.g., large-scale opt., SAND)
3. *Function evaluation coarse-grained parallelism:* concurrent execution of separable simulations within a fn. eval. (e.g., multiple loading cases)
4. *Function evaluation fine-grained parallelism:* parallelization of the solution steps within a single analysis code (e.g., SALINAS, MPSalsa)

DAKOTA Advanced Topics: Hybrid and Advanced Algorithms

http://dakota.sandia.gov

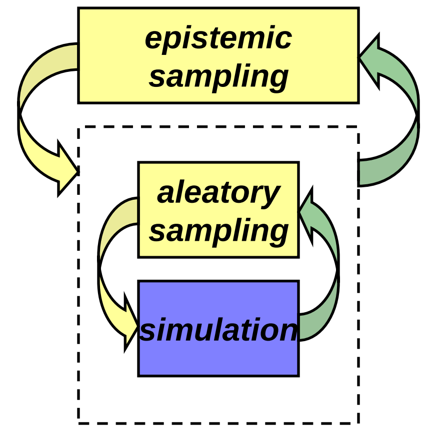
Opportunities for Mixing and Matching Methods



Strategies ***(general nesting, layering, sequencing and recasting facilities)*** *combine methods to enable advanced studies:*

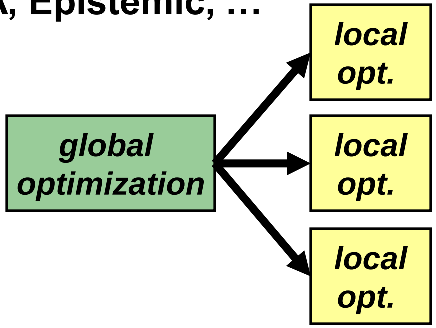
- **opt within opt (multilevel opt & hierarchical MDO)**
- **UQ within UQ (second-order probability)**
- **UQ within opt (OUU) and NLS (MCUU)**
- **opt within UQ (uncertainty of optima)**

with and without surrogate model indirection



Optimization

- **Surrogate-based: data fit, multifidelity, ROM**
- **Mixed integer nonlinear programming (MINLP): PEBBL (parallel branch and bound)**
- **Optimization under uncertainty**
  - **TR-SBOUU, RBDO (Bi-level, Sequential)**
  - **MCUU, PC-BDO, EGO/EGRA, Epistemic, …**
- **Hybrids (e.g., global/local)**
- **Pareto set**
- **Multi-start**
- **Multilevel methods**



Uncertainty

- **Second order probability**
- **Uncertainty of optima**

Nonlinear least squares

- **Surrogate-based calibration**
- **Model calibration under uncertainty**

Need to think of relationships between DAKOTA input blocks

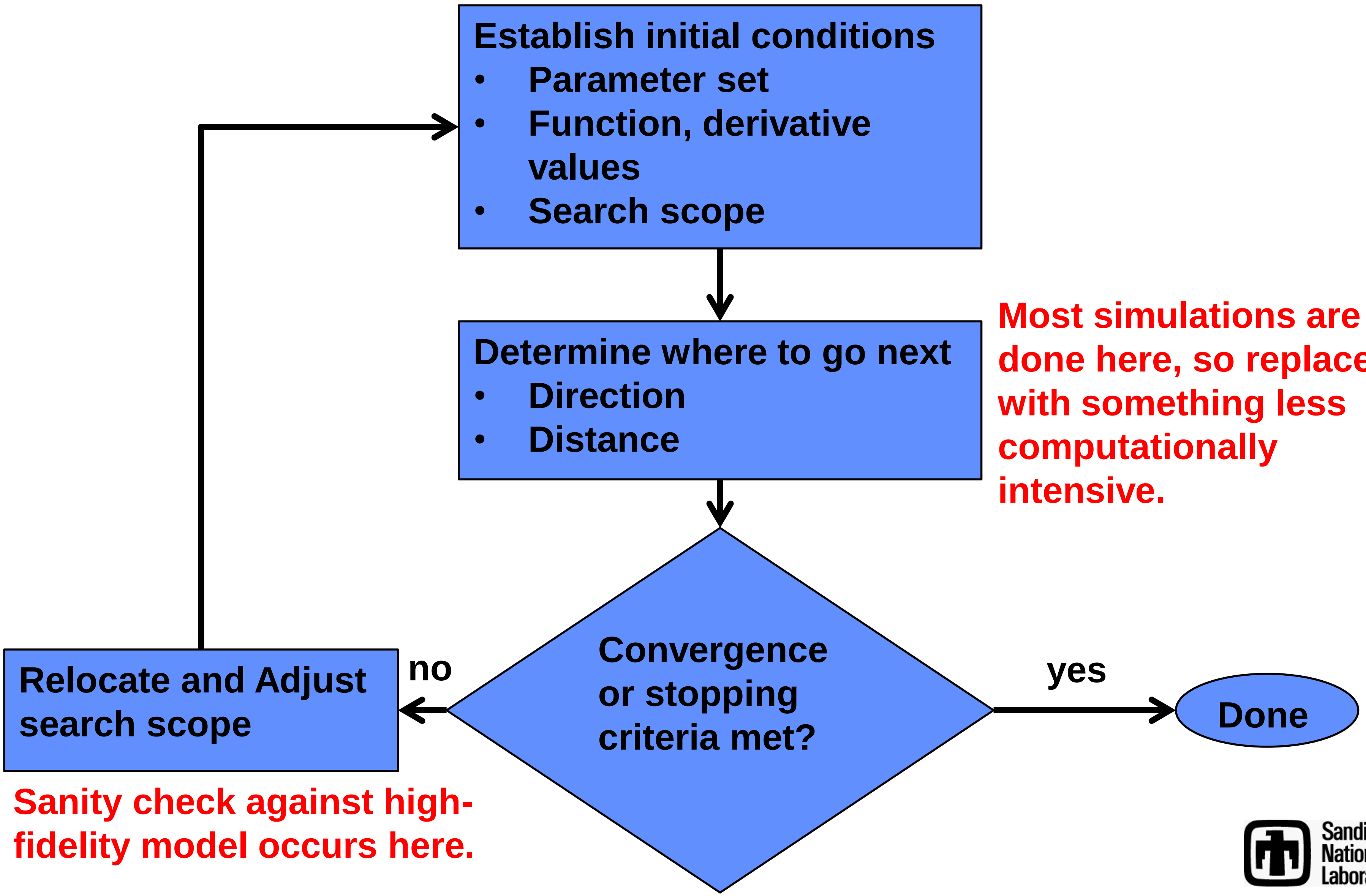
- Strategy
  - Consists of a method or set of methods

- Method
  - Operates on a model

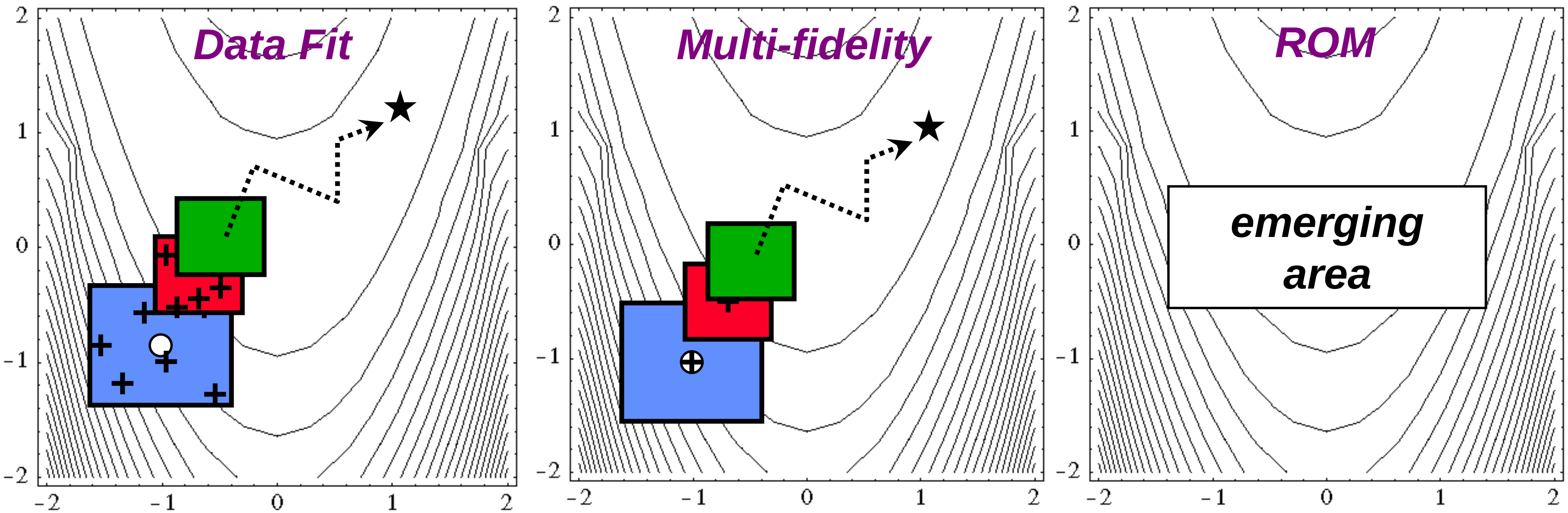
- Model has
  - Variables/parameters
  - Responses
  - Interface

There may be more than one of these in a DAKOTA input file.

Structure of surrogate-based (or multi-fidelity) optimization



Trust Region Surrogate-Based Minimization



Data fit surrogates

- Global: polynomials, splines, neural network, Kriging, RBFs
- Local: 1st/2nd-order Taylor

Data fits in SBO

- Smoothing: extract global trend
- DACE: limited # design vars
- Must balance local consistency with global accuracy

Multifidelity surrogates:

- Coarser discretizations, looser conv. tols., reduced element order
- Omitted physics: e.g., Euler CFD, panel methods

Multifidelity SBO

- HF scale better w/ des. vars.
- Requires smooth LF model
- May require design mapping
- Correction quality is crucial

ROM surrogates:

- Spectral decomposition
- POD/PCA w/ SVD
- KL/PCE (random fields, stochastic processes)

ROMs in SBO

- Key issue: parametrize (extended or spanning ROM)
- Otherwise like data fit case

Many Types of Data-Fit Surrogates

Polynomials are accurate in small regions and smooth noisy data.

linear

$$\hat{f}(\mathbf{x}) \approx c_0 + \sum_{i=1}^{n} c_i x_i$$

quadratic

$$\hat{f}(\mathbf{x}) \approx c_0 + \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n}\sum_{j\geq i}^{n} c_{ij} x_i x_j$$

cubic

$$\hat{f}(\mathbf{x}) \approx c_0 + \sum_{i=1}^{n} c_i x_i + \sum_{i=1}^{n}\sum_{j\geq i}^{n} c_{ij} x_i x_j + \sum_{i=1}^{n}\sum_{j\geq i}^{n}\sum_{k\geq j}^{n} c_{ijk} x_i x_j x_k$$

Splines can represent complex multi-modal surfaces and smooth noisy data.

$$\hat{f}(\mathbf{x}) = \sum_{m=1}^{M} a_m B_m(\mathbf{x})$$

↑ truncated power basis functions

Gaussian processes are good predictors of mean and variance but can suffer from ill conditioning.

$$\hat{f}(\underline{x}) \approx \underline{g}(\underline{x})^T \underline{\beta} + \underline{r}(\underline{x})^T \underline{\underline{R}}^{-1}(\underline{f} - \underline{\underline{G}}\,\underline{\beta})$$

↑ trend ↑ correlation

Correction terms can be applied to surrogates for improved accuracy.

additive

$$\hat{f_{hi_\alpha}}(\mathbf{x}) = f_{lo}(\mathbf{x}) + \alpha(\mathbf{x})$$

multiplicative

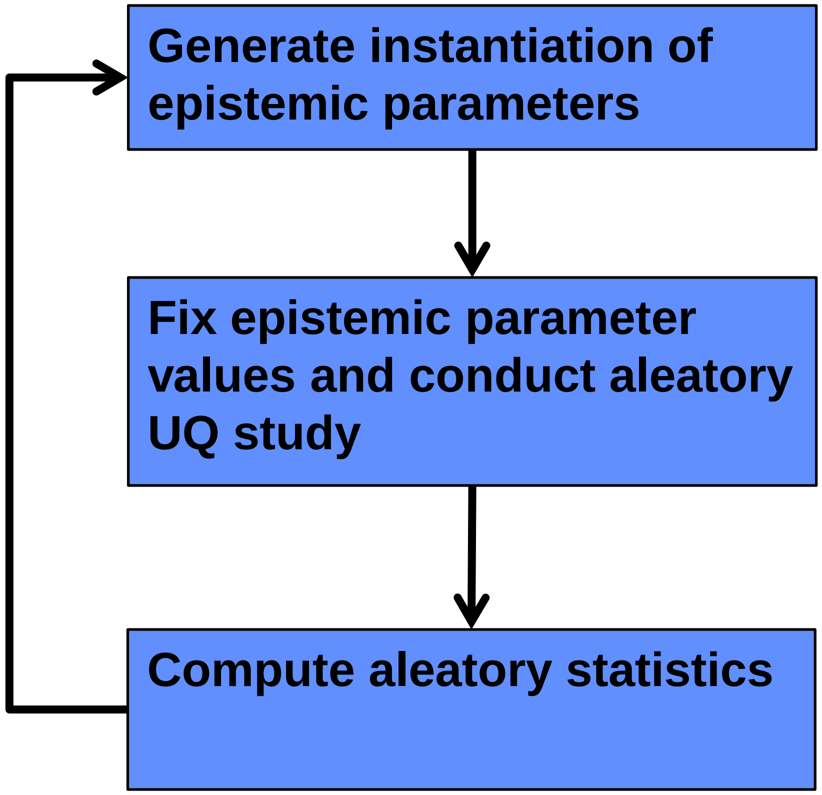
$$\hat{f_{hi_\beta}}(\mathbf{x}) = f_{lo}(\mathbf{x})\beta(\mathbf{x})$$

convex combination

$$\hat{f_{hi_\gamma}}(\mathbf{x}) = \gamma \hat{f_{hi_\alpha}}(\mathbf{x}) + (1-\gamma)\hat{f_{hi_\beta}}(\mathbf{x})$$

Sandia National Laboratories

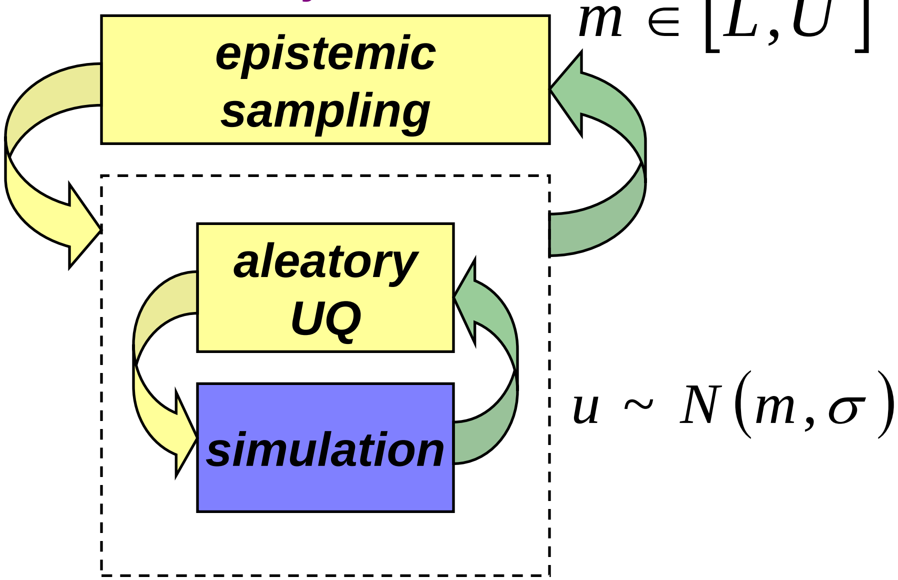
Structure of mixed (or nested) uncertainty quantification

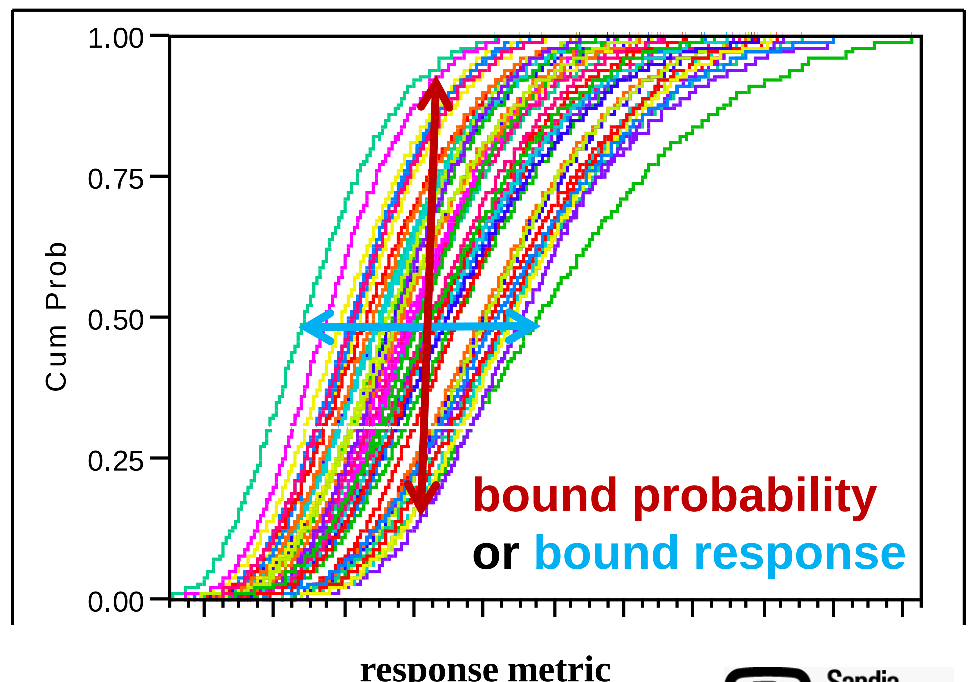


Epistemic UQ: Nested (“Second-order”)Approaches

- **Propagate over epistemic and aleatory uncertainty, e.g., UQ with bounds on the mean of a normal distribution (hyper-parameters)**
- **Typical in regulatory analyses (e.g., NRC. WIPP)**
- **Outer loop: epistemic (interval) variables, inner loop UQ over aleatory (probability) variables; *potentially costly, not conservative***
- ***If treating epistemic as uniform, do not analyze probabilistically!***

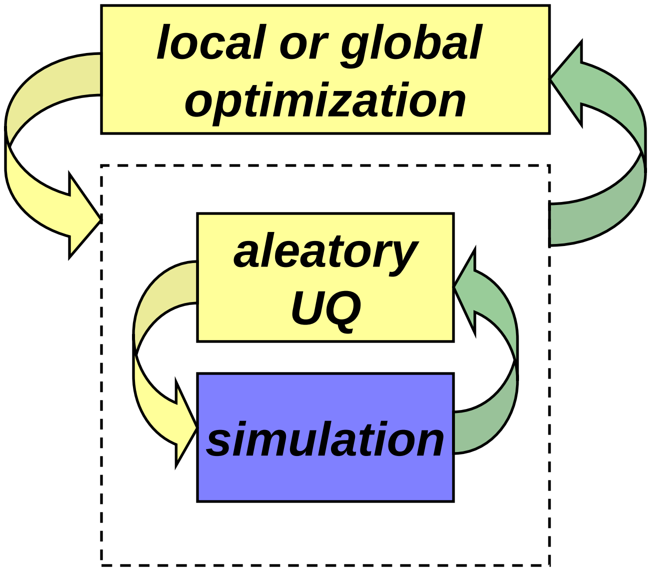
50 outer loop samples:
50 aleatory CDF traces





“Envelope” of CDF traces represents response epistemic uncertainty

Interval Estimation Approach (Probability Bounds Analysis)



- *Propagate intervals through simulation code*
- **Outer loop: determine interval on statistics, e.g., mean, variance**
  - **global optimization problem: find max/min of statistic of interest, given bound constrained interval variables**
  - **use EGO to solve 2 optimization problems with essentially one Gaussian process surrogate**
- **Inner loop: Use sampling, PCE, etc., to determine the CDFs or moments with respect to the aleatory variables**

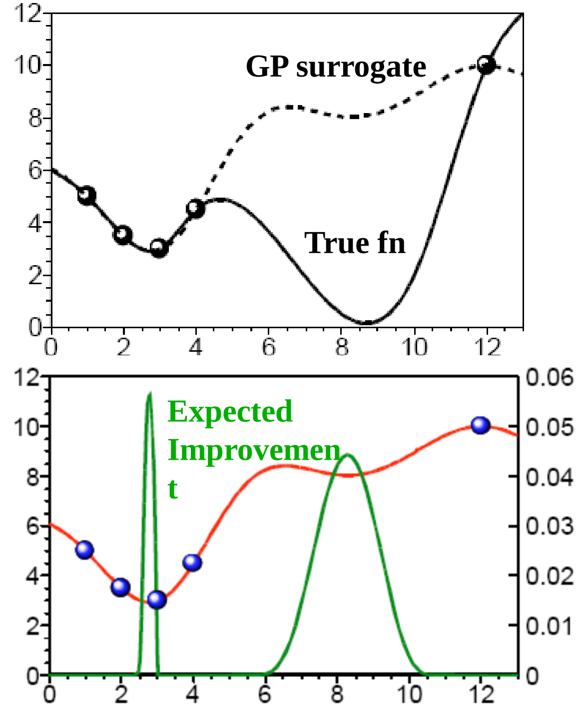
$$\min_{u_E} \quad f_{STAT}(u_A \mid u_E)$$
$$u_{LB} \le u_E \le u_{UB}$$
$$u_A \sim F(u_A ; u_E)$$

$$\max_{u_E} \quad f_{STAT}(u_A \mid u_E)$$
$$u_{LB} \le u_E \le u_{UB}$$
$$u_A \sim F(u_A ; u_E)$$

Sandia National Laboratories

Efficient Global Optimization

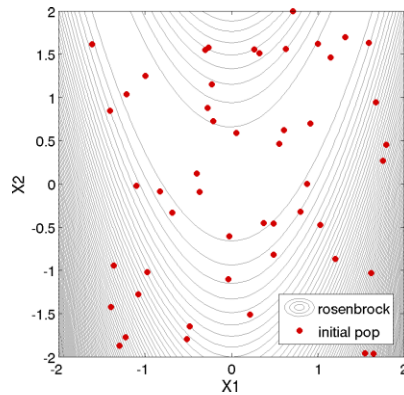
- **Technique due to Jones, Schonlau, Welch**
- **Build global Gaussian process approximation to initial sample**
- **Balance global exploration (add points with high predicted variance) with local optimality (promising minima) via an "expected improvement function"**



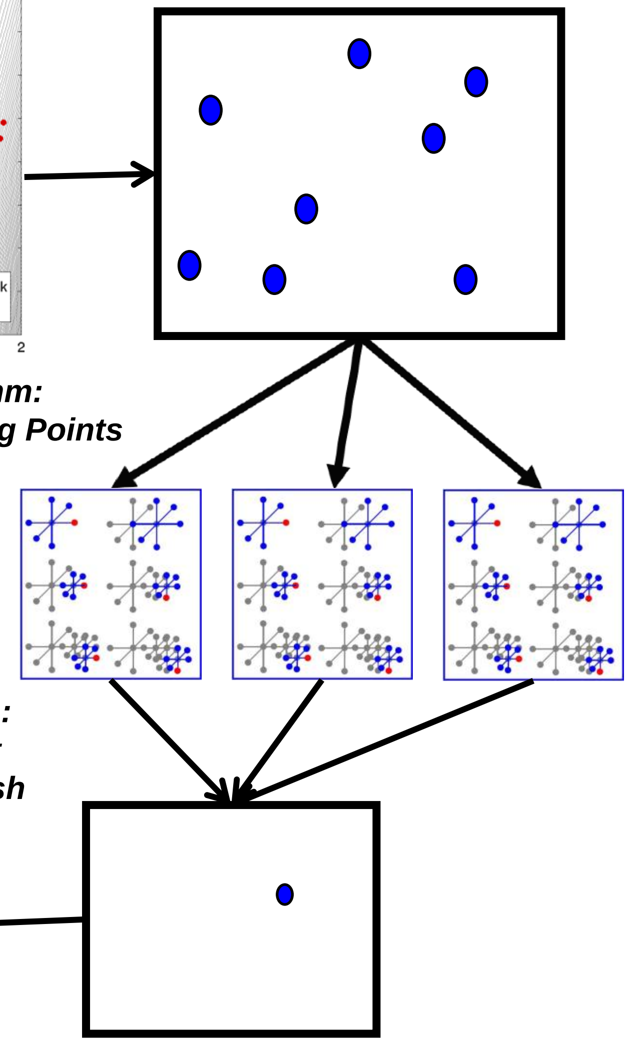
From Jones, Schonlau, Welch, 1998

Hybrid Optimization

```
strategy,
  graphics
  hybrid sequential
    method_list = 'GA' 'PS' 'NLP'
method,
  id_method = 'GA'
  model_pointer = 'M1'
  coliny_ea
    seed = 1234
    population_size = 10
    verbose output
method,
  id_method = 'PS'
  model_pointer = 'M1'
  coliny_pattern_search stochastic
    seed = 1234
    initial_delta = 0.1
    threshold_delta = 1.e-4
    solution_accuracy = 1.e-10
    exploratory_moves basic_pattern
    verbose output
method,
  id_method = 'NLP'
  model_pointer = 'M2'
  optpp_newton
    gradient_tolerance = 1.e-12
    convergence_tolerance = 1.e-15
    verbose output
model,
  id_model = 'M1'
  single
    variables_pointer = 'V1'
    interface_pointer = 'I1'
    responses_pointer = 'R1'
model,
  id_model = 'M2'
  single
    variables_pointer = 'V1'
    interface_pointer = 'I1'
    responses_pointer = 'R2'
variables,
  id_variables = 'V1'
  continuous_design = 2
    initial_point     0.6     0.7
    upper_bounds      5.8     2.9
    lower_bounds      0.5    -2.9
    descriptors       'x1'    'x2'
interface,
  id_interface = 'I1'
  direct
    analysis_driver=   'text_book'
responses,
  id_responses = 'R1'
  num_objective_functions = 1
  no_gradients
  no_hessians
responses,
  id_responses = 'R2'
  num_objective_functions = 1
  analytic_gradients
  analytic_hessians
```



Evolutionary Algorithm: Generates Multiple Starting Points for Pattern Search



Pattern Search Ensemble: Generates Starting Point for Newton Method to finish

Newton Method

Multi-Objective Optimization

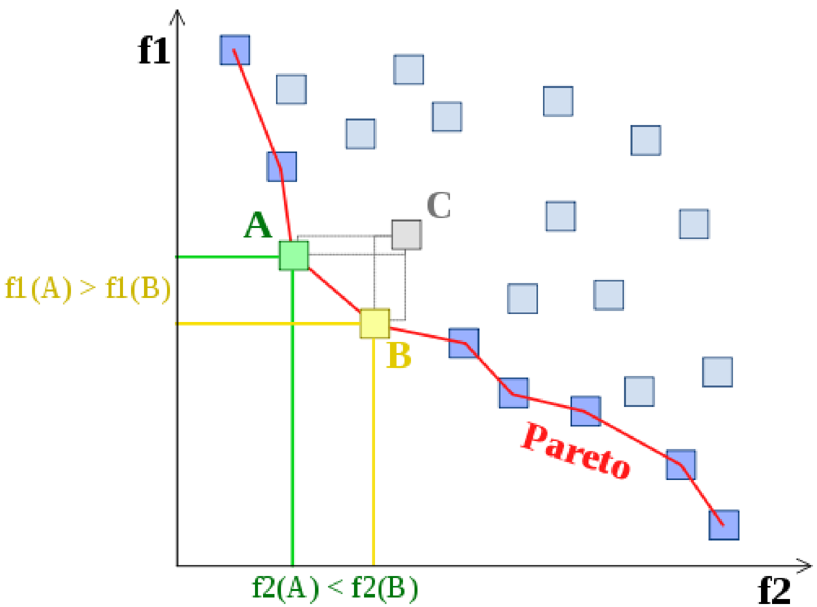


Image from http://en.wikipedia.org/wiki/Pareto_efficiency

May want tradeoffs between multiple objectives.

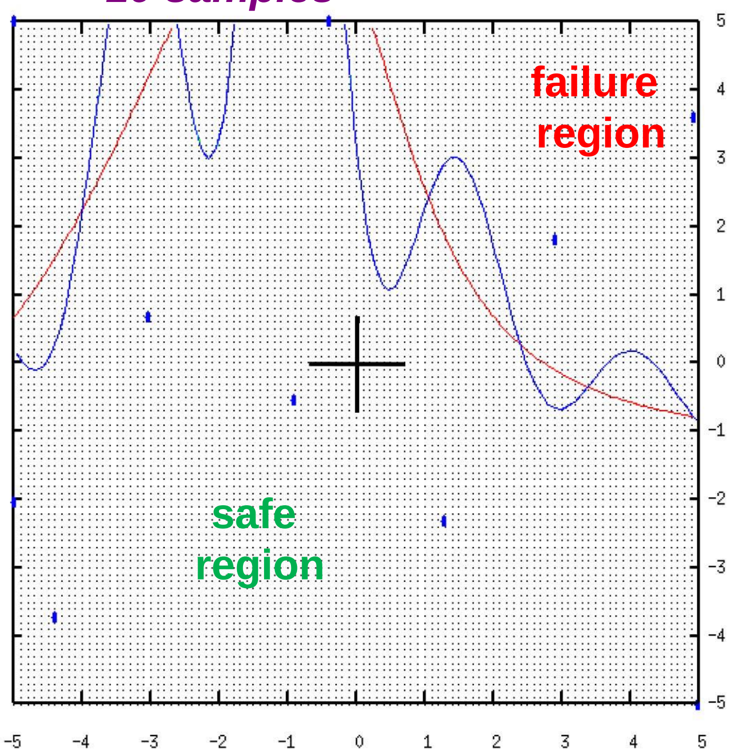
```
strategy,
  single_method
  tabular_graphics_data
method,
  optpp_q_newton
    output verbose
    convergence_tolerance = 1.e-8
variables,
  continuous_design = 2
    initial_point     0.9     1.1
    upper_bounds      5.8     2.9
    lower_bounds      0.5    -2.9
    descriptors       'x1'    'x2'
interface,
  system asynchronous
    analysis_driver=  'text_book'
responses,
  num_objective_functions = 3
  multi_objective_weights = .7 .2 .1
  analytic_gradients
  no_hessians
```

Efficient Global Reliability Analysis: GP Surrogate + MMAIS (B.J. Bichon)

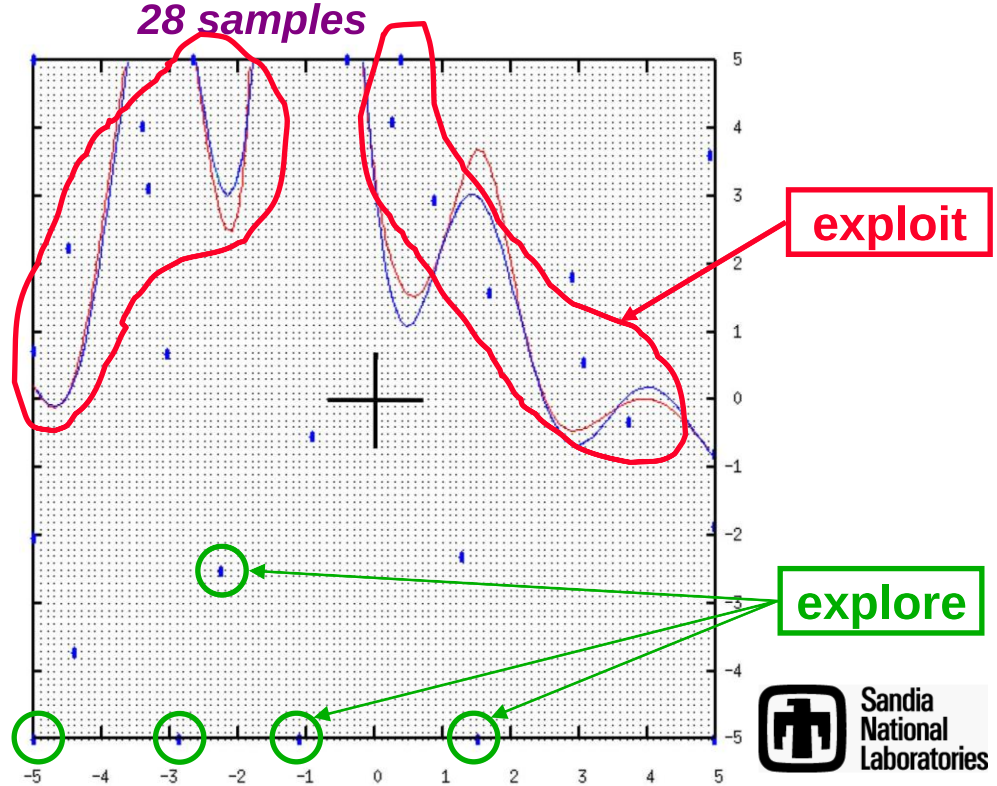
- **Apply an EGO-like method to the equality-constrained optimization problem**
- **In EGRA, an expected feasibility function balances exploration with local search near the failure boundary to refine the GP**
- **Cost competitive with best MPP search methods, yet better probability of failure estimates; addresses nonlinear and multimodal challenges**

Gaussian process model (level curves) of reliability limit state with

10 samples



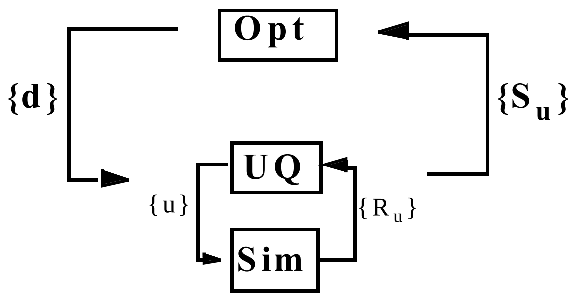
28 samples



Sandia National Laboratories

Optimization Under Uncertainty

Rather than design and then post-process to evaluate uncertainty…
actively design optimize while accounting for uncertainty/reliability metrics $s_u(d)$, e.g., mean, variance, reliability, probability:



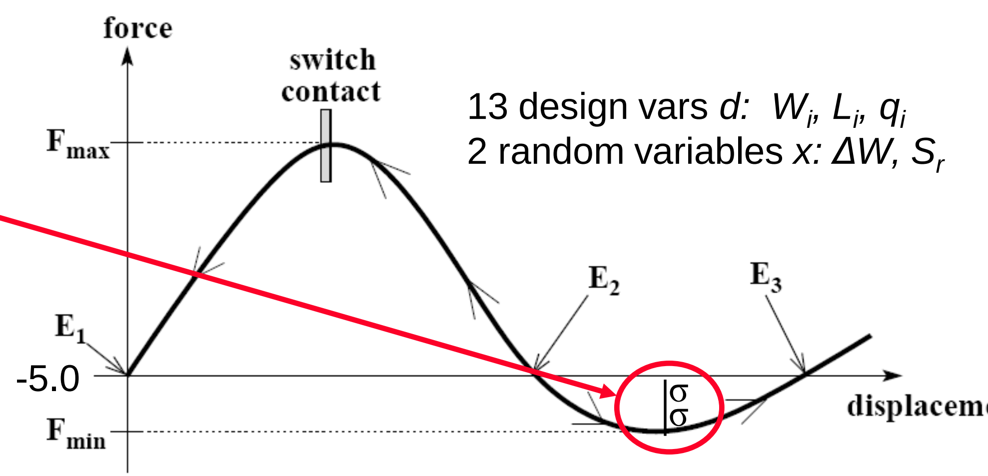
(nested paradigm)

$$
\begin{aligned}
\min \quad & f(d) + W s_u(d) \\
\text{s.t.} \quad & g_l \leq g(d) \leq g_u \\
& h(d) = h_t \\
& d_l \leq d \leq d_u \\
& a_l \leq A_i\, s_u(d) \leq a_u \\
& A_e\, s_u(d) = a_t
\end{aligned}
$$

Bistable switch problem formulation (Reliability-Based Design Optimization):

simultaneously reliable and robust designs

$$
\begin{array}{llcll}
\text{max} & & & \mathsf{E}\left[F_{min}(\mathbf{d},\mathbf{x})\right] & \\
\text{s.t.} & 2 & \leq & \beta_{ccdf}(\mathbf{d}) & \\
 & 50 & \leq & \mathsf{E}\left[F_{max}(\mathbf{d},\mathbf{x})\right] & \leq 150 \\
 & & & \mathsf{E}\left[E_2(\mathbf{d},\mathbf{x})\right] & \leq 8 \\
 & & & \mathsf{E}\left[S_{max}(\mathbf{d},\mathbf{x})\right] & \leq 3000
\end{array}
$$



13 design vars *d*: W_i, L_i, q_i
2 random variables *x*: ΔW, S_r