SAND2011-1891C

Simulation-based Optimization Tutorial

Brian M. Adams

Navigating Chemical Compound Space for Materials and Bio Design: Tutorials

March 15 – 18, 2011

Institute for Pure and Applied Mathematics

Los Angeles, CA

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.

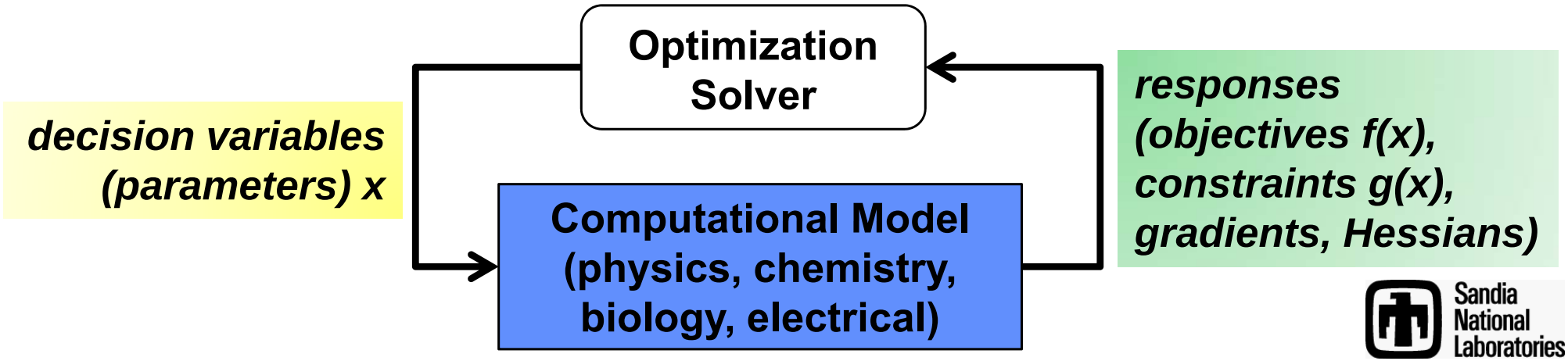
Outline
Simulation-based Optimization

A high-level introduction to performing optimization using computational models

- **Motivating application examples**
- **Challenges**
- **Discussion of calibration (parameter estimation)**
- **Survey of algorithms**
  - **Derivative-based local**
  - **Derivative-free local**
  - **Global**
- **Advanced approaches**
- **Discussion**

Simulation-based Optimization

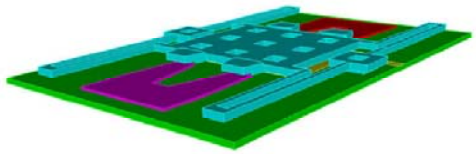
- **GOAL: Vary parameters of a simulation to extremize objectives, while satisfying constraints to find (or tune) the best design, estimate best parameters, analyze worst-case surety**
- **Mapping from decision variables to objectives and constraints is (at least partially) implicit; no explicit algebraic form**
- **Relationship is calculated by a "black box" computational model of target phenomenon (often loosely coupled to the solver)**
- **Optimization methods iteratively evaluate this simulation and adapt based on its outputs**
- **Generic, so flexible; can apply to nearly any simulation, but can be computationally costly**



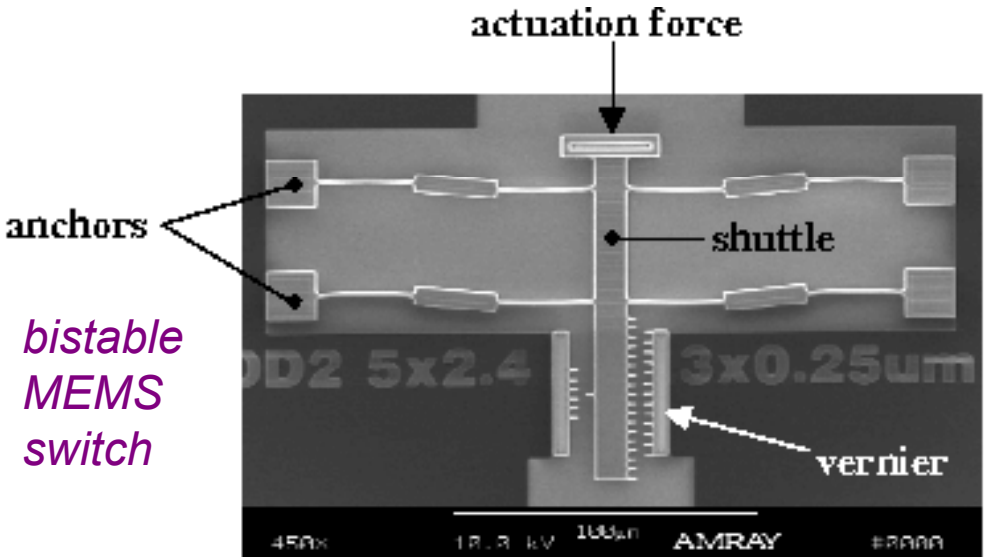
Examples in Brief: Optimization with Simulators

- Use a combined discrete event / differential equation model of a flu epidemic to determine optimal interventions to minimize mortality given limited drug resources
- Determine an aero shell case geometry to minimize weight and drag (as calculated by a fluids code) while maintaining sufficient strength and safety (calculated by solid mechanics simulator)
- Adjust atom positions to find minimum energy molecular configurations (may be algebraic)
- Optimally locate groundwater insertion and extraction wells to abate a hazardous underground plume using a flow simulator accounting for geophysical data
- Maximize traffic throughput given safety constraints as modeled by an urban traffic flow simulator

Shape Optimization of Compliant MEMS



- **Micro-electromechanical system (MEMS):** typically made from silicon, polymers, or metals; used as micro-scale sensors, actuators, switches, and machines
- **MEMS designs are subject to substantial variability** and lack historical knowledge base. Materials and micromachining, photo lithography, etching processes all yield uncertainty.
- Resulting part yields can be low or have poor cycle durability
- **Goal: shape optimize finite element model of bistable switch to…**
  - **Achieve prescribed reliability** in actuation force
  - Minimize sensitivity to uncertainties **(robustness)**

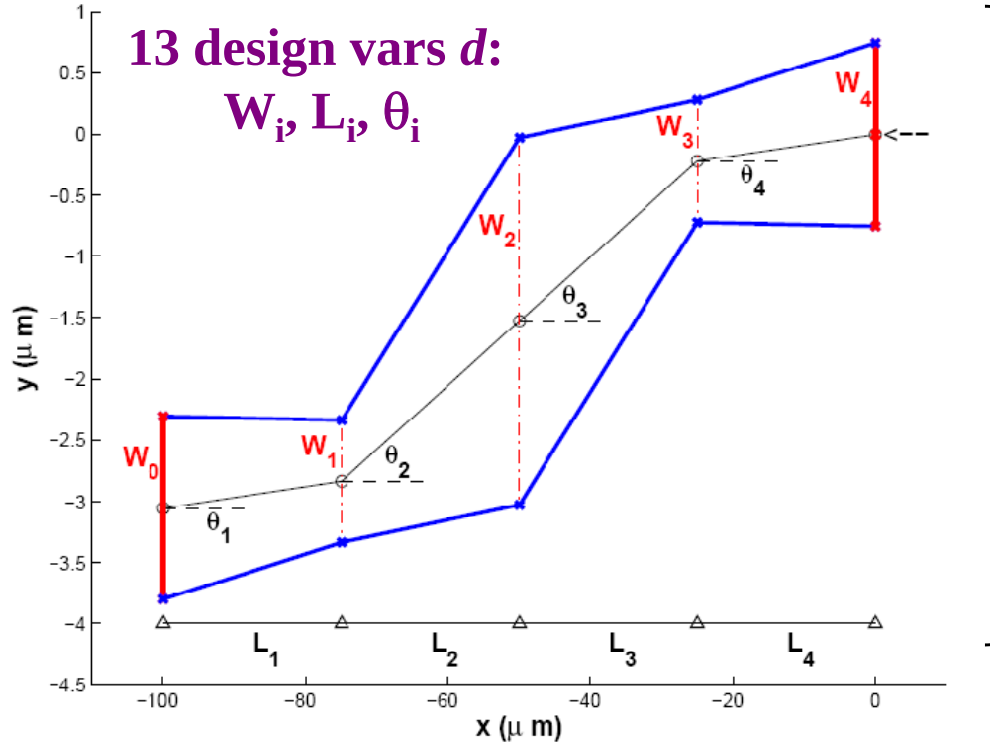


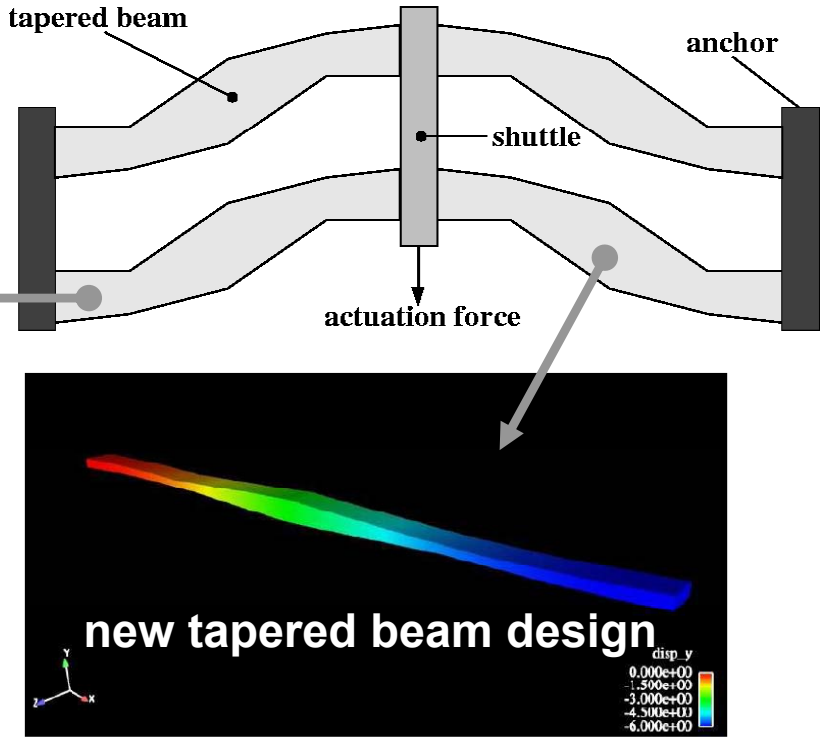
uncertainties to be considered (edge bias and residual stress)

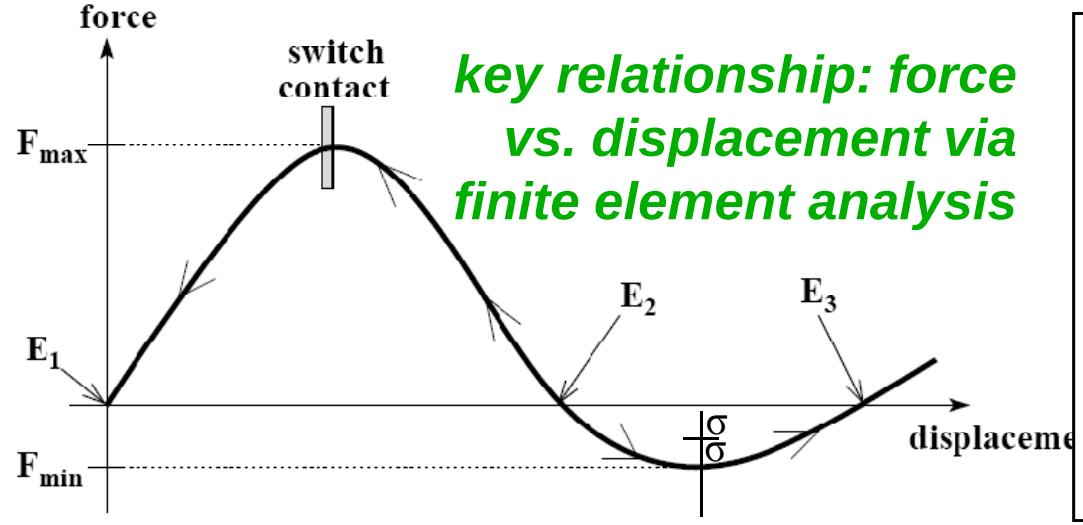
variable	mean	std. dev.	distribution
Δw	-0.2 μm	0.08	normal
S_r	-11 Mpa	4.13	normal

Sandia National Laboratories

MEMS Switch Design: Geometry Optimization





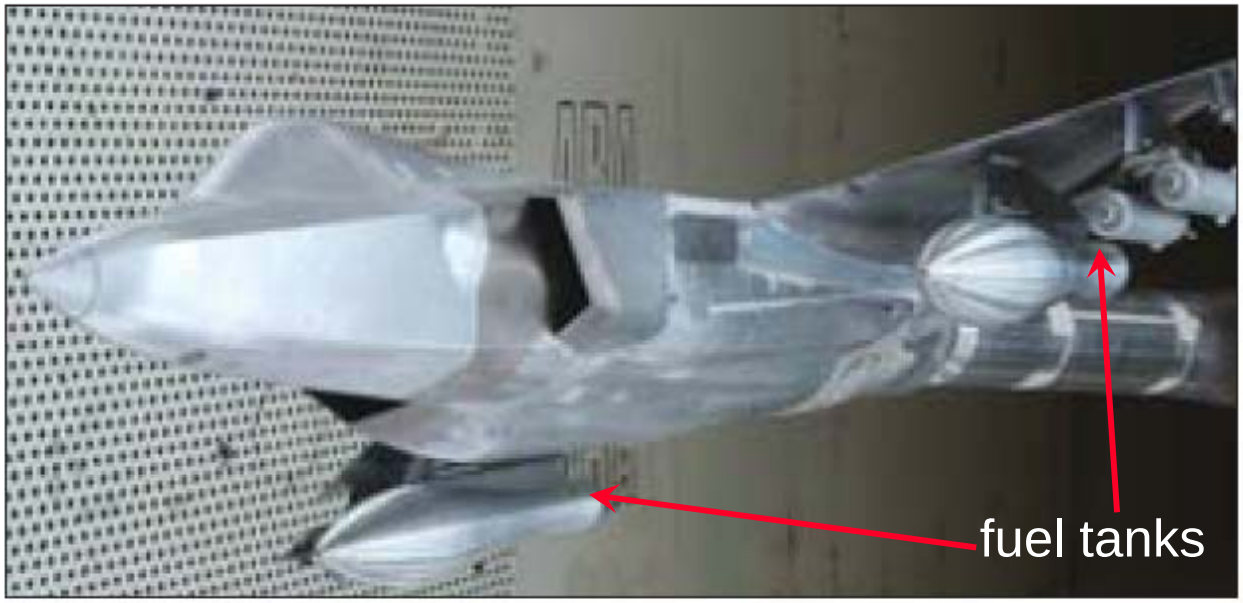


key relationship: force vs. displacement via finite element analysis

Typical design specifications:

- actuation force F_{min} reliably 5 μN
- bistable ($F_{max} > 0$, $F_{min} < 0$)
- maximum force: $50 < F_{max} < 150$
- equilibrium E2 < 8 μm
- maximum stress < 1200 MPa

Optimization for Lockheed-Martin F-35 External Fuel Tank Design



This wind tunnel model of F-35 features an optimized external fuel tank.

F-35: stealth and supersonic cruise
~ $20 billion cost
~ 2600 aircraft (USN, USAF, USMC, UK & other foreign buyers)

LM CFD code:

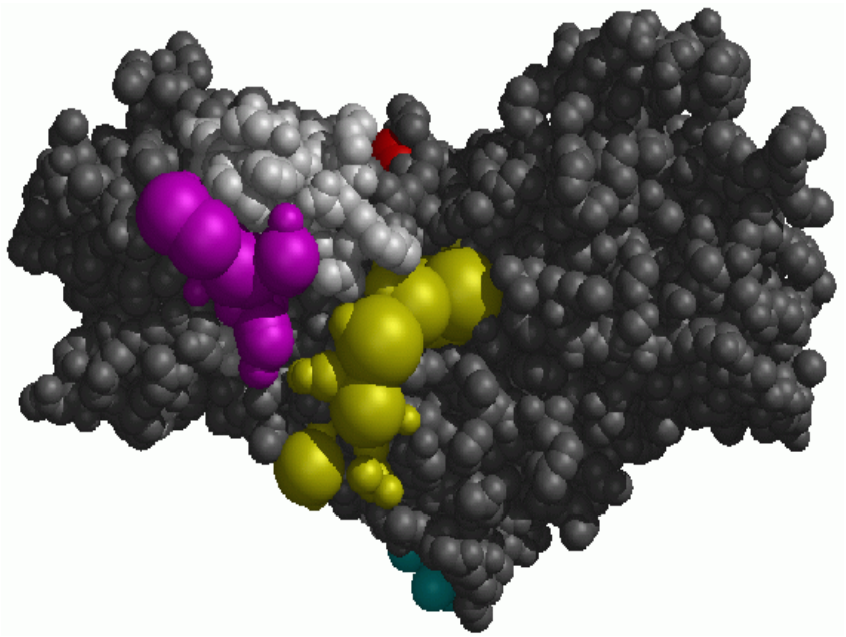
- ***Expensive*: 8 hrs/job on 16 processors**
- **Fluid flow around tank *highly sensitive* to shape changes**

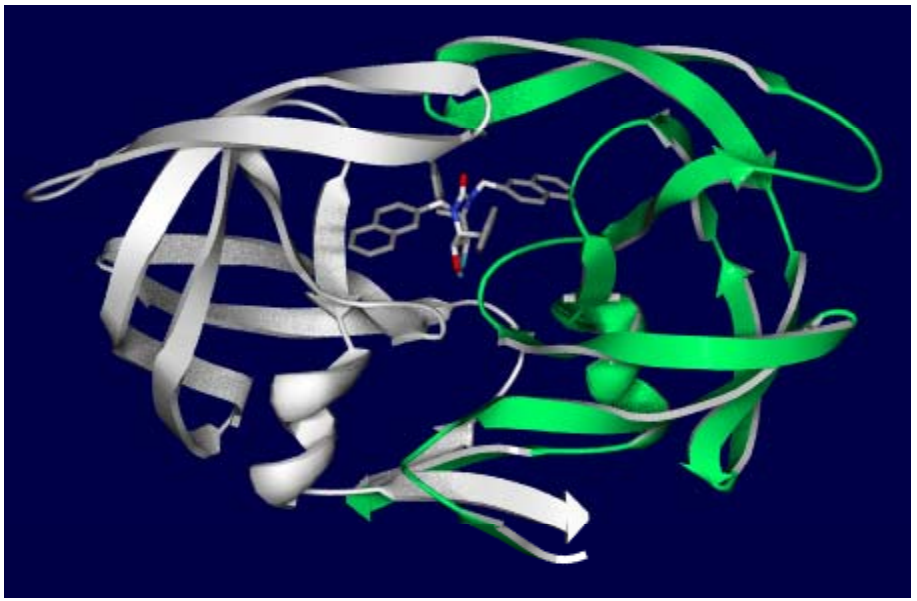
"Lockheed Martin Aeronautics conducted a trade study for the F-35 Joint Strike Fighter (JSF) aircraft to design the external fuel tank for improved performance, store separation, and flutter. **CFD was used in conjunction with Sandia National Laboratories' Dakota optimization code to determine the optimal shape of the tank that minimizes drag for maximum range and minimizes yawing moment for separation of adjacent stores.** Data obtained at several wind tunnel facilities verified the predicted performance of the new aeroshaped, compartmented tank for separation and flutter, as well as acceptable characteristics for loads, stability, and control." -- Dec. 2004 *Aerospace America*, p. 22

Sandia National Laboratories

Drug Docking (Courtesy Bill Hart)

- Problem: find the optimal binding for a small ligand in a binding site



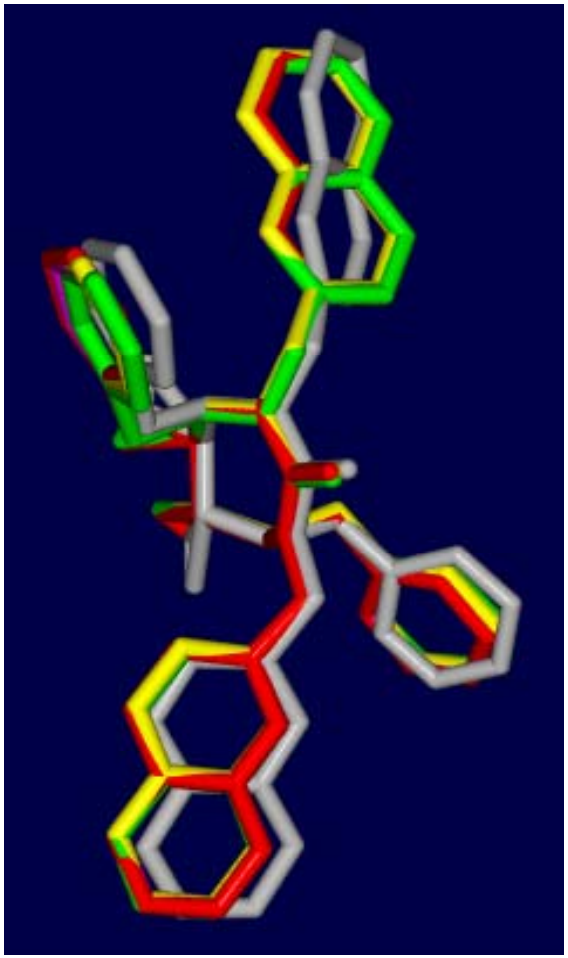


- Application: lead compound development
- Impact: limit lab experimentation required to develop new drugs
- Approach: heuristic global optimization of flexible docking empirical potential functions

Sandia National Laboratories

Flexible Drug Docking in AutoDock Software

- Idea: dock a small flexible ligand into a static binding site



Search dimensions (decision variables)

- Cartesian coordinates (3 dimensions)
- Quarternion (unit vector plus rotation angle)
- Torsion angles

Formulation

- Nonsmooth energy potential
  - Standard energy model: electrostatic, van de Waals, Hydrogen bonds, etc
  - Trilinear interpolation of atomic interactions
- Simple bound constraints
- Unit-length quarternion constraint

- Note: Binding constraints or ring constraints possible

Sandia National Laboratories

Typical Challenges for Simulation-based Optimization

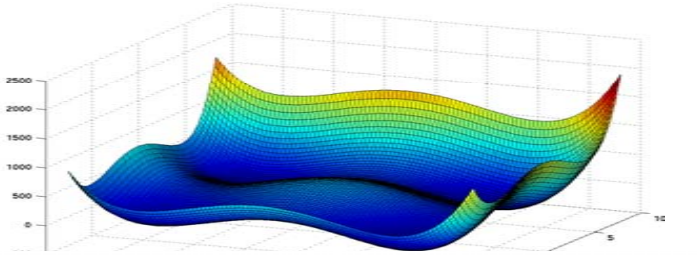
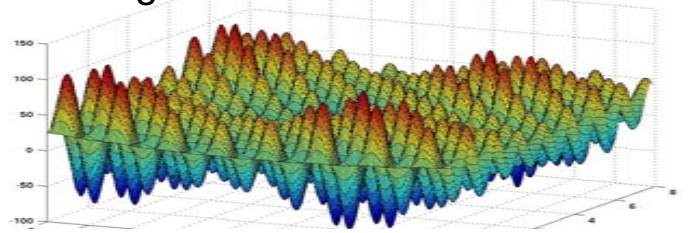
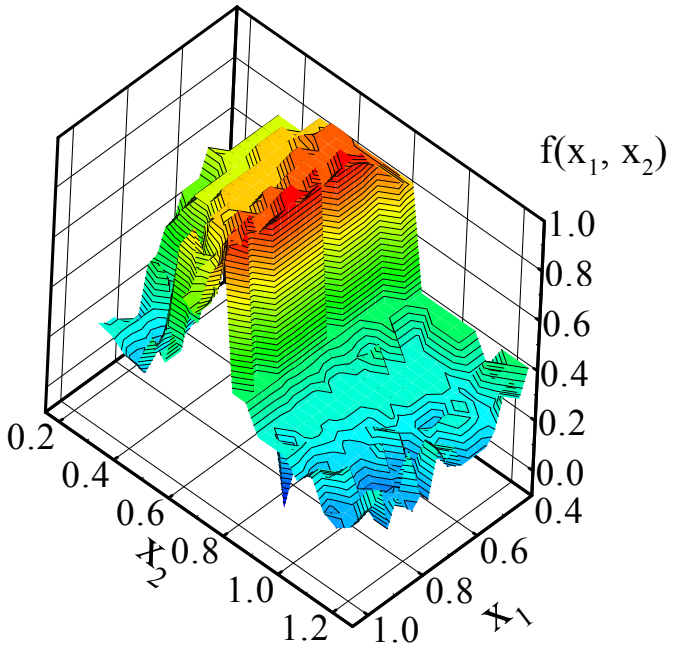


Image credits: John Siirola





In science and engineering problems of interest, we typically have:

- **no explicit function for $f(x_1,x_2)$**
  - **can't leverage algebraic structure**
- **limited number of evaluations/samples**
  - **expensive to evaluate $f(x_1,x_2)$ (long runtime even on many processors)**
  - **simulation may fail (hidden constraints)**
- **noisy / non-smooth**
  - **can't reliably estimate derivatives**
- **local extrema, non-convex**
  - **globally optimal solutions challenging**

Considerable research has been done to mitigate these issues.

Sandia National Laboratories

Special Optimization Case: Calibration (Parameter Estimation)

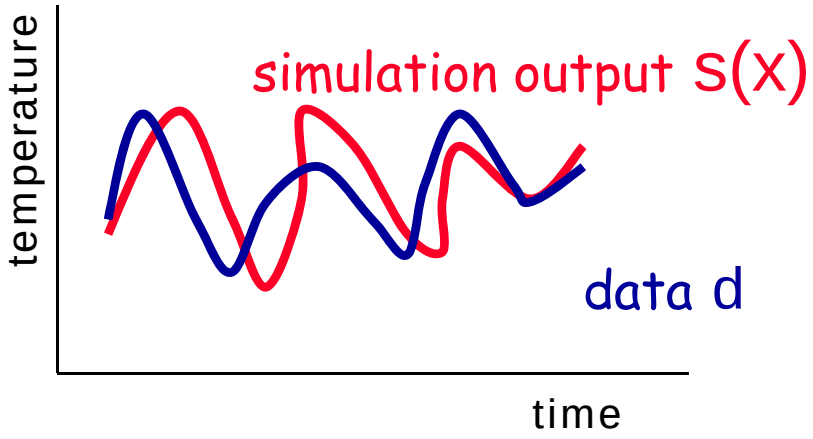
- **Calibration: Adjust model parameters x to maximize agreement with a set of experimental data.**
- **A.K.A. parameter estimation, parameter identification, systems identification, nonlinear least-squares, inverse problem.**

Minimize

$$f(x) = \sum_{i=1}^{n} (s_i(x) - d_i)^2$$

simulation output that depends on x

given data



Sandia National Laboratories

Why use calibration?

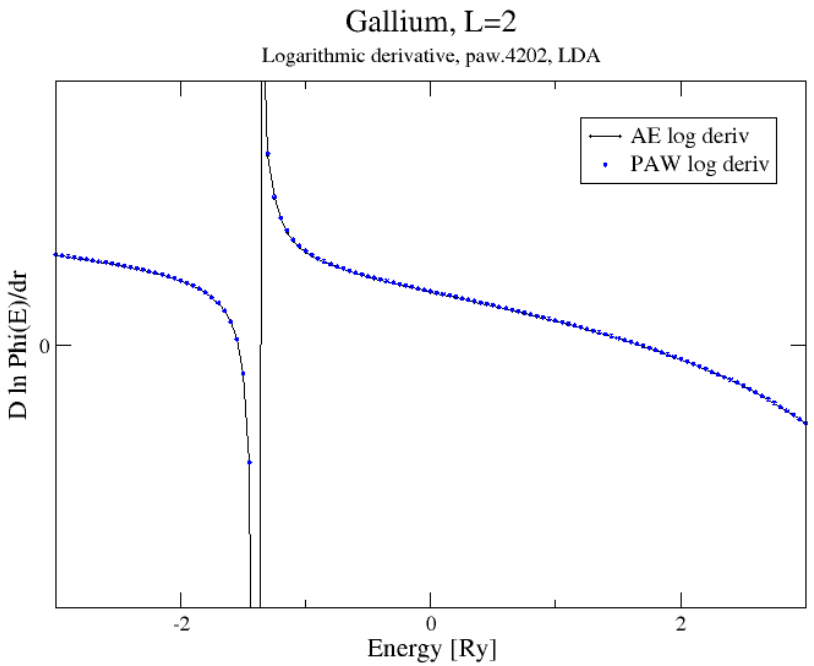
- Tune a model to experimental or trusted simulation data to
  - ensure sufficient simulation code predictive capability
  - decrease the amount of info lost due to using a model instead of the "truth" (minimize discrepancy)
  - gain understanding of design space
  - find parameters yielding improved model robustness

- Calibration is not validation! Separate data should be used to assess whether a calibrated model is valid.

- Calibration (inverse) problems often suffer from non-uniqueness or lack of identifiability of some parameters

Calibration of Projector-Augmented Wave (PAW) functions

- Ryan Snow (UC Davis) and Alan Wright (SNL)
- Use genetic algorithms with SOCORRO and ATOMPAW
- PAW functions (in contract to NCPs) permit a smaller basis set to be used for most atoms in the periodic table, more accurate electron scattering features, and explicit treatment of more atoms
- SOCORRO can use the PAW method, but is limited by a lack of publicly available PAW functions, which are typically determined through hand calibration
- Calibrate based on
  - electron scattering properties of the PAW functions,
  - computed properties of various materials,
  - efficiency of the SOCORRO calculations used to obtain the material properties.

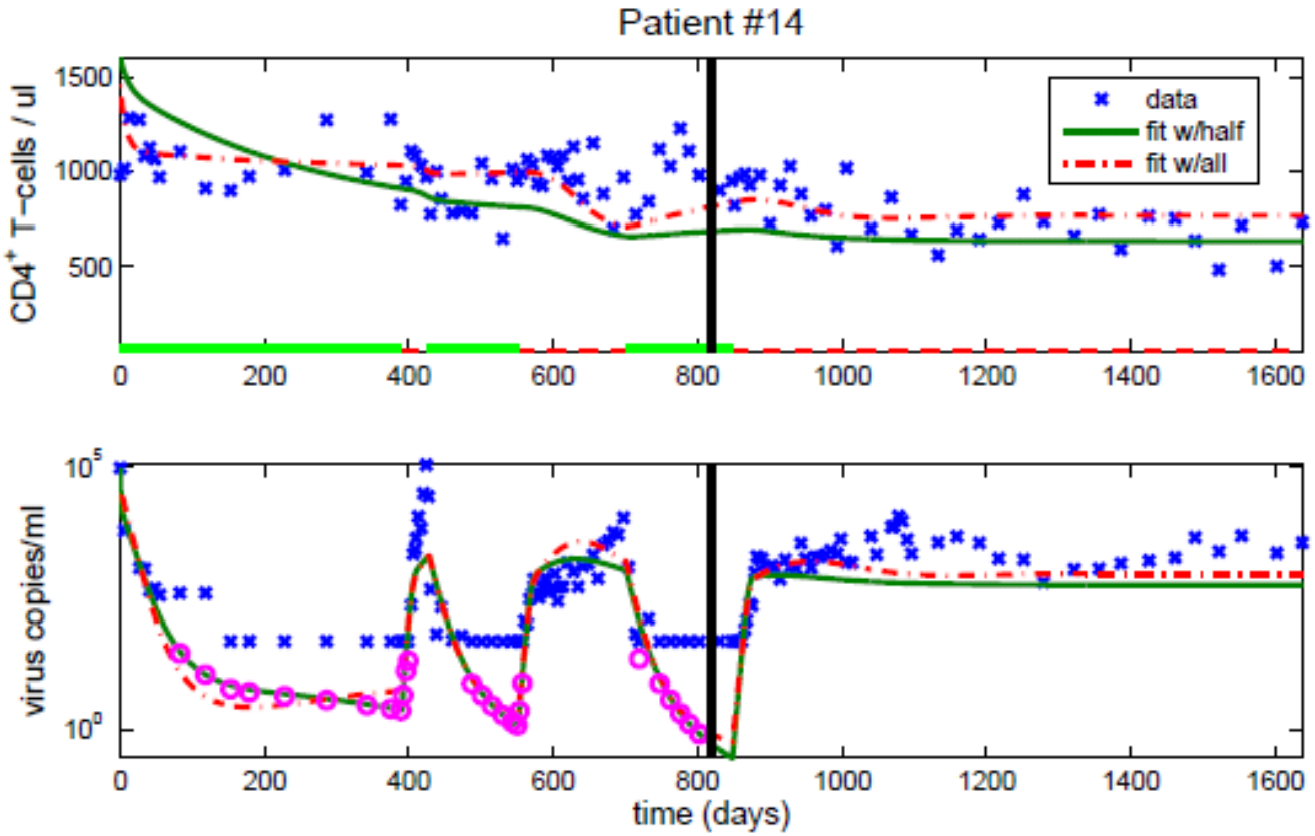


Optimal PAW function for gallium determined by coupling DAKOTA to the SOCORRO density functional theory (DFT) code.

Sandia National Laboratories

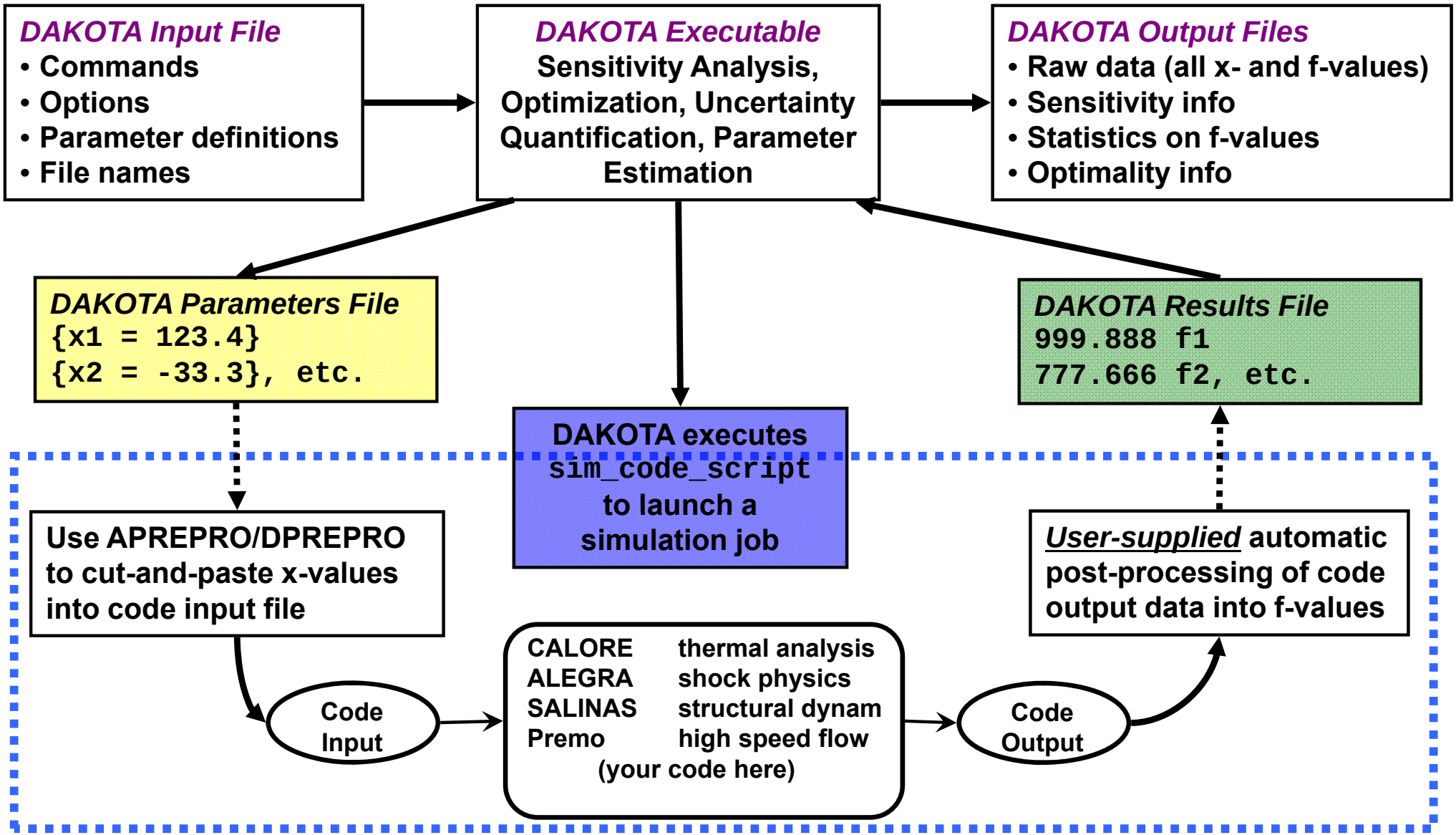
HIV Model Calibration

- **Calibrate HIV model (system of ODEs) to patient data to predict long-term outcomes**



Sandia National Laboratories

Flexibility: Typical Loose-Coupled Workflow (DAKOTA example)



Sandia National Laboratories

Flexibility

- Simulation-based optimization is (largely) agnostic to the details of the computational model
- With many frameworks, once you've developed an interface to the simulation, you can readily try several algorithms

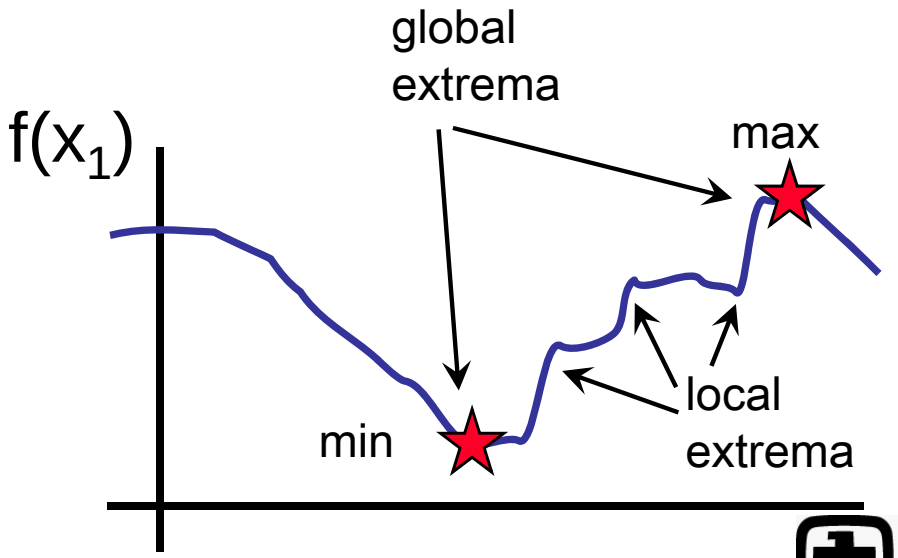
- To specify an optimization problem and solution technique, need to consider:
  - optimization goal
  - formulation of objective and constraints
  - selection of solver

- Need to be able to distill simulation output into a form acceptable to the solver (typically a handful of scalars)

Problem Formulation: Goal

Potential optimization goals:

- **local optimization: incremental improvement suffices; there are no better designs/configurations nearby**
  - **example: make minor changes to car shape to reduce drag, resulting in huge cost savings, but retaining essential design of car**
- **global optimization: must find global minimum (best scenario) at any cost (often considerably greater)**
  - **explore all feasible support structure designs to produce the most reliable bridge**
- **Importance of constraints**
  - **feasibility then optimality**
  - **optimality with relaxed constraints**



Problem Formulation: Objectives and Constraints

Information with which to configure the solver:

Minimize:	$f(x_1, ...,x_N)$	***Objective function(s)****
Subject to:	$g_{LB} \leq g(x) \leq g_{UB}$	***Nonlinear inequality constraints***
	$h(x) = h_E$	***Nonlinear equality constraints***

(Metrics above are typically implicit: computed by/extracted from a simulation code)

(Algebraic metrics below are typically specified directly to an optimization solver)

$A_I x \leq b_I$	***Linear inequality constraints***
$A_E x = b_E$	***Linear equality constraints***
$x_{LB} \leq x \leq x_{UB}$	***Bound constraints***

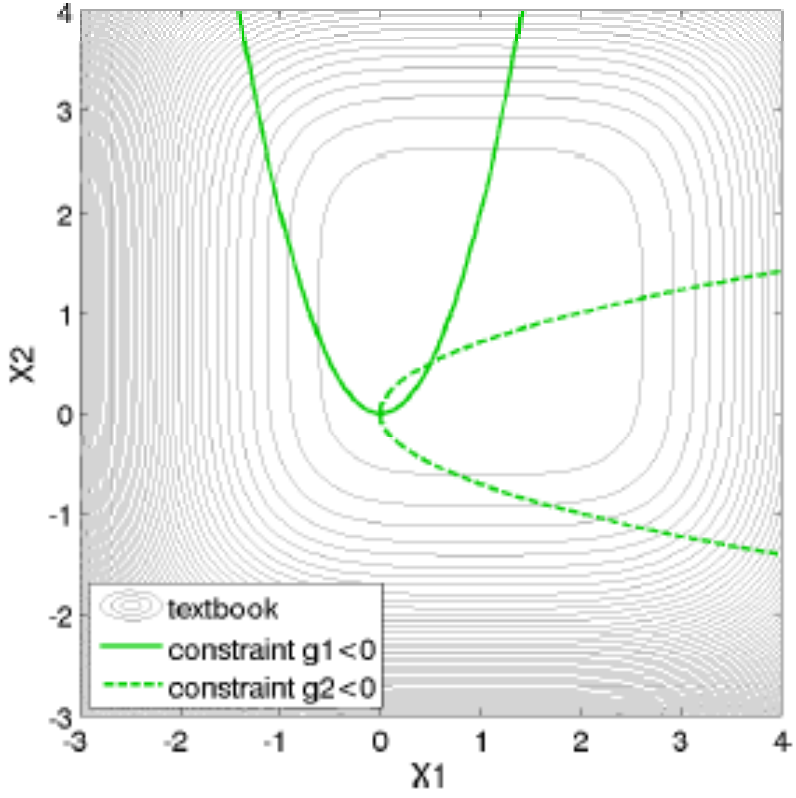
**** In practice, multiple f-values can comprise the objective function ("multi-objective optimization"), and there can be multiple constraints of each type.***

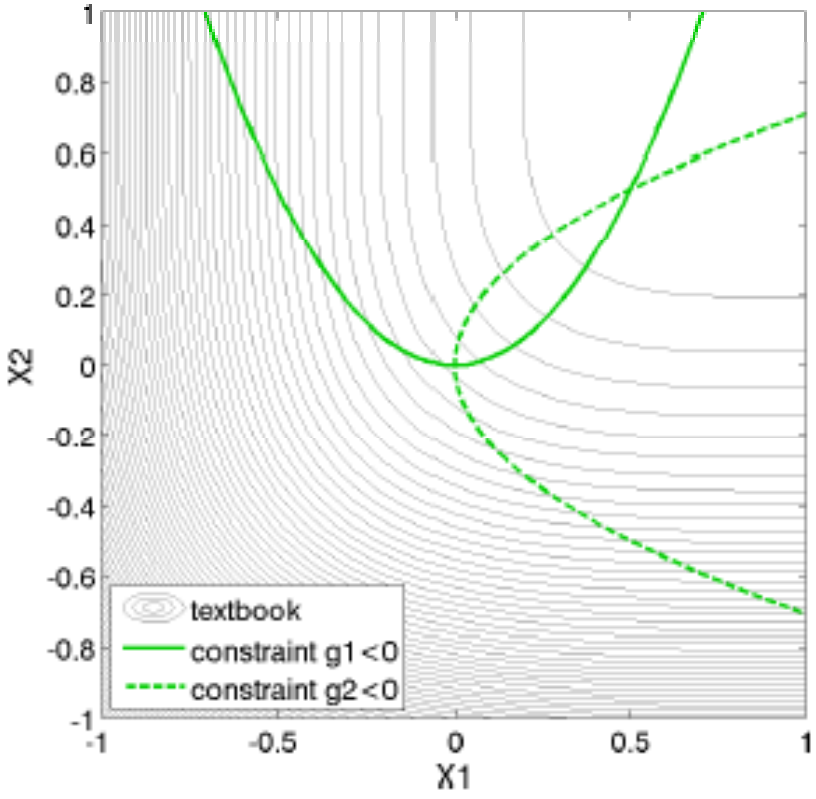
Sandia National Laboratories

Nonlinear Constraints Implicitly Bound the Search Space

- **Nonlinear constraints: "text book" problem**
- **In black-box problems, solver can only gain information about the constraints by evaluating the model**

$$
\begin{aligned}
\texttt{minimize} \quad & f = (x_1 - 1)^4 + (x_2 - 1)^4 \\
\texttt{subject to} \quad & g_1 = x_1^2 - \frac{x_2}{2} \leq 0 \\
& g_2 = x_2^2 - \frac{x_1}{2} \leq 0 \\
& 0.5 \leq x_1 \leq 5.8 \\
& -2.9 \leq x_2 \leq 2.9
\end{aligned}
$$





Sandia National Laboratories

Constraint Progression

Listed in order of (typically) increasing algorithm complexity and computational cost needed to solve.

- **Unconstrained problem: neither bound constraints nor linear/ nonlinear constraints**
- **Bound-constrained problem: bound (variable space x) constraints only (no linear/nonlinear constraints)**
- **Linearly-constrained problem: constraints are linear with respect to the x-variables (may also have bound constraints)**
- **Nonlinearly-constrained problem: the g(x) and h(x) constraints, nonlinear w.r.t. the x variables, are present (may also have bound constraints)**
  perhaps most typical in engineering and science applications

Typically, it is important to specify constraints as specifically as possible, e.g., don't specify a linear constraint as nonlinear if the solver supports linear constraints.

Considerations when Choosing an Optimization Method

Key considerations

- Trend and smoothness (perform local and global sensitivity analysis)
- Simulation expense
- Constraint types present
- Goal: local optimization (improvement) or global optimization (best possible)
- Variable types present (real, integer, categorical)
- Any special structure, e.g., quadratic objective, highly linearly constrained

Unconstrained or bound-constrained problems

- Smooth and cheap: nearly any method but gradient-based 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, though direct search works too
- Smooth and expensive: gradient-based methods
- Nonsmooth and cheap: non-gradient methods w/ penalty functions, SBO
- Nonsmooth and expensive: SBO

Outline
Simulation-based Optimization

A high-level introduction to performing optimization using computational models

- Motivating application examples
- Challenges
- Discussion of calibration (parameter estimation)
- **Survey of algorithms**
  - **Derivative-based local**
  - **Derivative-free local**
  - **Global**
- **Advanced approaches**
- **Discussion**

Gradient-based Local Algorithms

- **For smooth problems over continuous variables *x*, including with many variables and constraints, derivative-based methods can be very efficient**
- **Require reliable derivatives of objectives and any nonlinear constraints w.r.t. decision variables; calculate via**
  - **analytic evaluation (calculate derivatives by hand or symbolically and code them into the simulation)**
  - **finite differences via approximations like (no code modification needed and most optimizers can do automatically)**

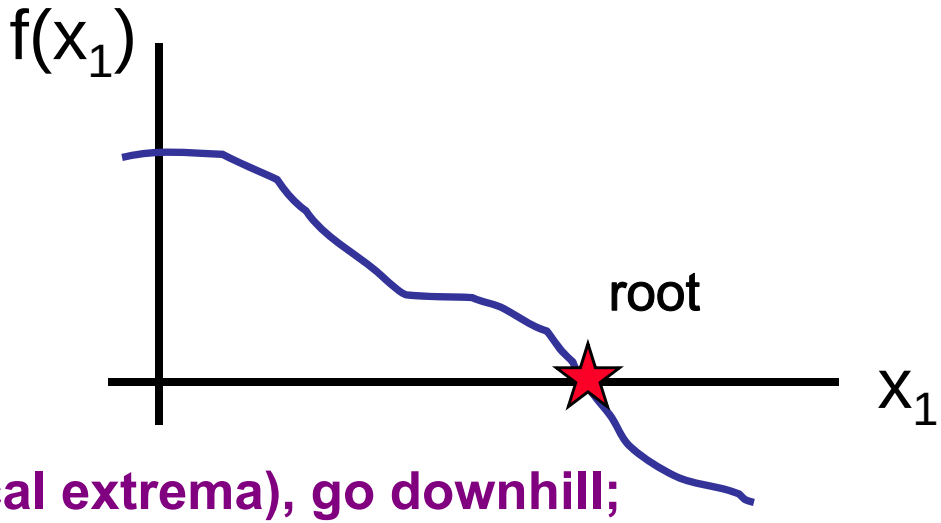
$$\frac{\partial f}{\partial x} \approx \frac{f(x+\Delta x) - f(x)}{\Delta x}$$

  - **automatic differentiation via**
    - **source transformation**
    - **operator overloading**
- **Examples**
  - **Steepest descent (in practice, only with modifications)**
  - **Conjugate gradient (unconstrained)**
  - **Newton, quasi-Newton, and variants**

Gradient-based Optimization: Go Downhill

Modify Newton's root-finding method for solving f(x) = 0.

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

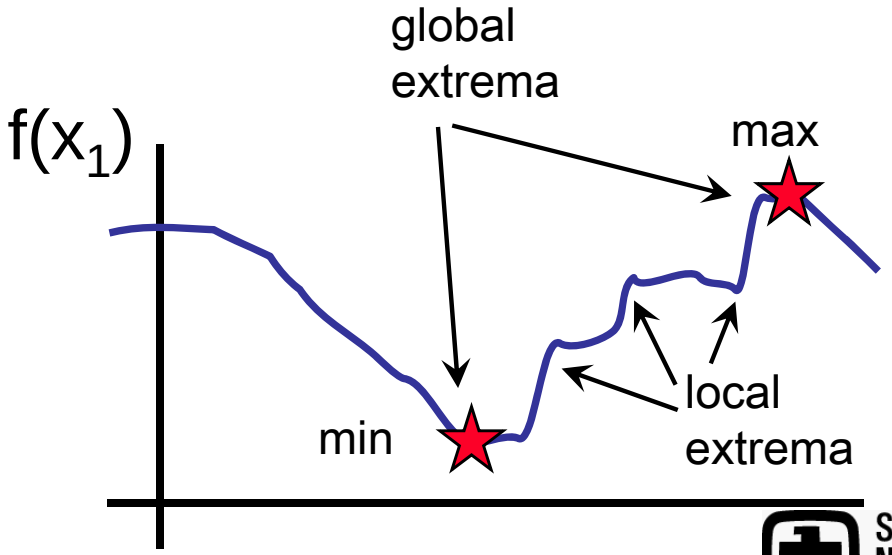


For optimization: find zeros of f'(x) = 0 (local extrema), go downhill; loosely

$$x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$$

These derivatives extend to gradients and Hessians in the multivariate case:

$$\nabla_x f(x), \quad \nabla_x^2 f(x)$$



Many Algorithmic Variations

Many proven packages offer derivative-based methods with

- Strategies for managing convergence:
  - Line search (minimize along a line in the Newton direction to ensure sufficient decrease)
  - Trust region (manage a quadratic model and expand/contract a trust region based on performance)
- Finite difference Newton, inexact Newton
- Second derivatives differentiate minima from maxima inflection points; Hessian approximations often used in practice (quasi-Newton)
  - BFGS, limited memory BFGS, rank one
- Nonlinear constraints
  - Reduced gradient
  - Sequential linear or quadratic programming (SLP/SQP)
  - Augmented Lagrangian or exact penalty methods
  - Interior point / barrier, filter methods

Nonlinear Least Squares (NLLS)

- **Calibration problems are often formulated to minimize the two norm of the error between the model and data: *minimize***

$$f(x) = \frac{1}{2} r(x)^T r(x) = \frac{1}{2} \left[s(x) - d \right]^T \left[s(x) - d \right] = \frac{1}{2} \sum_{i=1}^{n} \left(s_i(x) - d_i \right)^2$$

- **Example: `osborne1` analytic test problem, with *i = 1,…,33*:**

$$r_i(x) = \underbrace{\left(x_1 + x_2 e^{t_i x_4} + x_3 e^{t_i x_5} \right)}_{\text{model/simulation}} - \underbrace{d_i}_{\text{data}}; \quad t_i = -10(i-1)$$

- **Any optimizer can be applied to the sum of squared residuals *f(x)***
- **Variance-weighted and Bayesian calibration also popular**
- **A specialized class of derivative-based optimization algorithms exploit least sqaures structure for efficient solution without second derivative information**

Don't Hide Least-squares Structure from a Local Optimizer!

- **When minimizing *f(x)* with gradient-based methods, can take advantage of the form of its derivatives:**

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

$$\nabla f(x) = J(x)^T r(x); \quad J_{ij} = \frac{\partial r_i}{\partial x_j}$$

$$\nabla^2 f(x) = J^T J + \sum_{i=1}^{n} r_i(x) \nabla^2 r_i(x)$$

Algorithms vary in how they approximate this Hessian, but need the residuals r(x) and Jacobian separately instead of aggregate f(x)

Hessian Approximations

$$\nabla^2 f(x) = J^T J + \sum_{i=1}^{n} r_i(x) \nabla^2 r_i(x)$$

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

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

(typically used with a trust region)

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

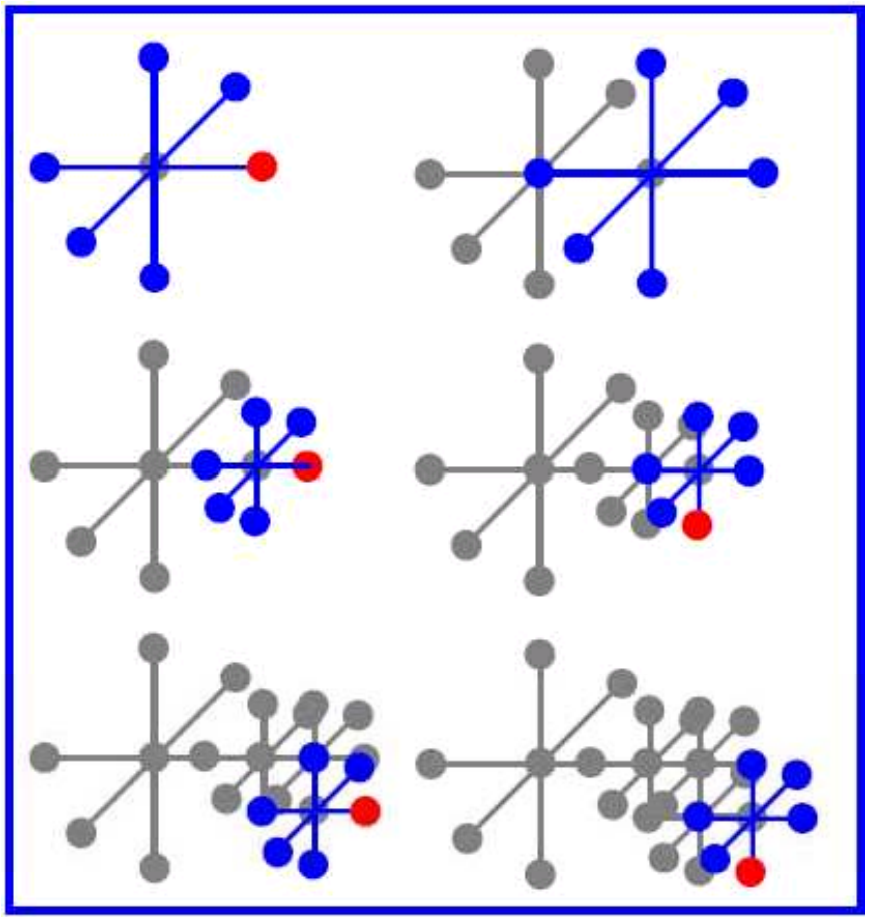
with $S = 0$ or $S =$ Quasi-Newton approximation to $\sum_{i=1}^{n} f_i(x) \nabla^2 f_i(x)$

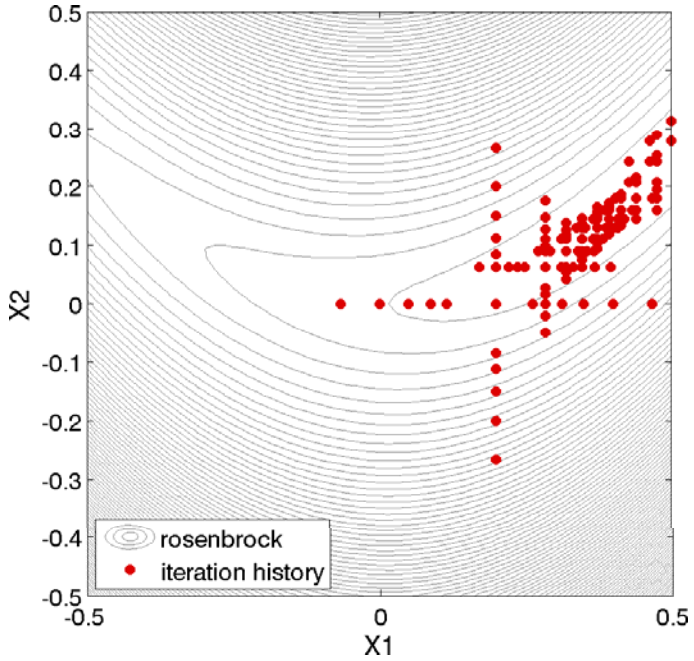
Derivate-free Local Algorithms

- Derivative-free local methods are good at finding local minima in the presence of noise or general absence or derivative information
- Essentially sampling-based, but with provable local convergence
- Examples
  - Pattern/direct search and simplex methods, e.g., Hooke-Jeeves, Nelder-Mead, multi-directional search
  - Implicit filtering (steepest descent, but with an adaptive simplex gradient)
  - Trust region surrogate-based (later)

Coordinate-basis Pattern Search

- Evaluate model at a stencil of points; recenter at point with best function value
- If no improvement, contract stencil (to achieve local convergence)
- Stencils typically are simplexes or align with coordinate directions

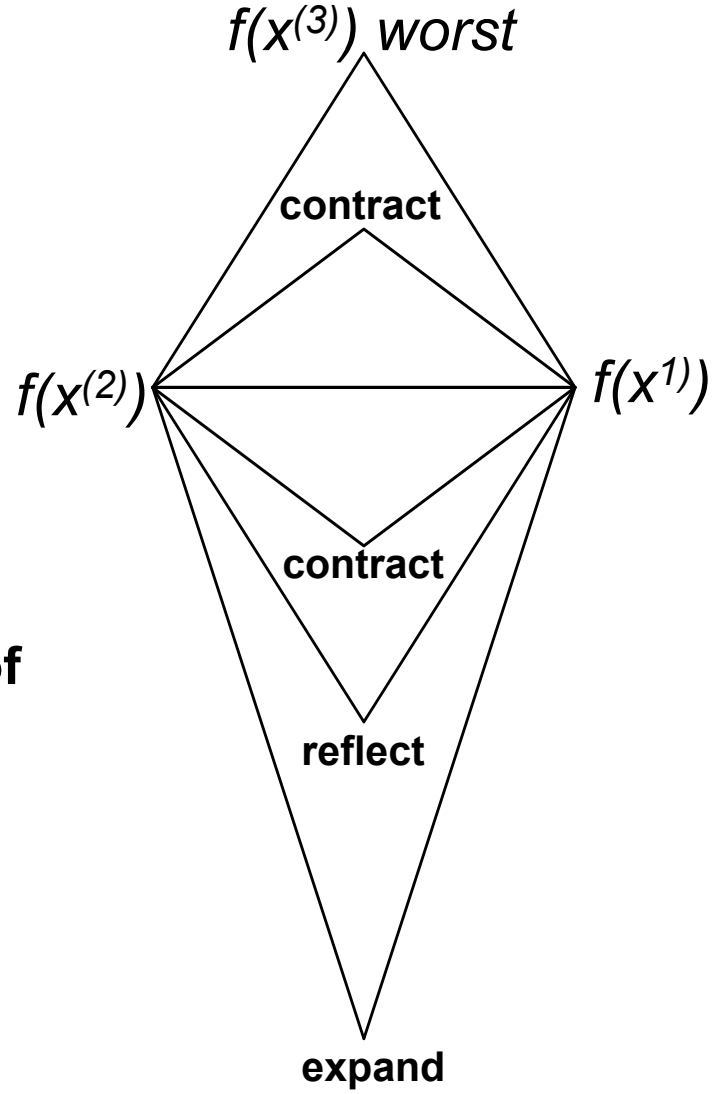




Sandia National Laboratories

Nelder-Mead

- **Simplex stencil of d+1 points, with centroid of d best**
- **Adapt the simplex iteratively to go downhill:**
  - **Reflect through centroid and if better value, replace worst**
  - **If improved, attempt to expand further in that direction**
  - **If no improvement, attempt contraction of the simplex to find a better point**
  - **If fails, shrink simplex**

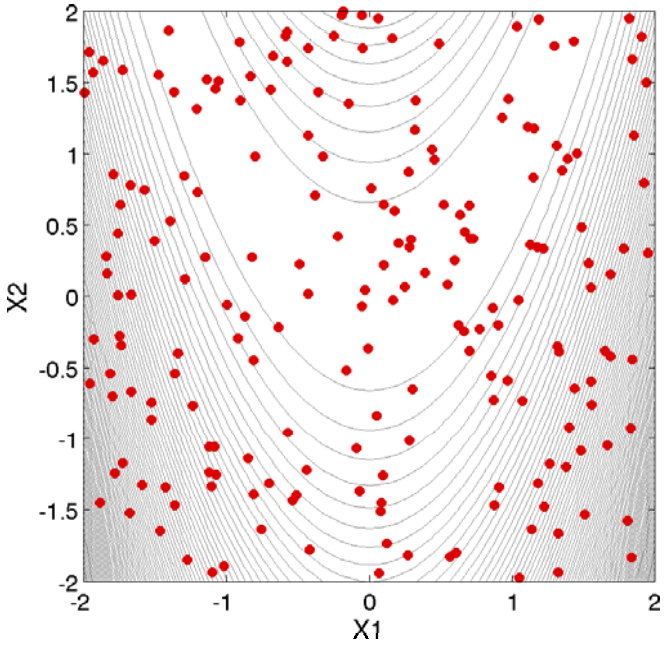


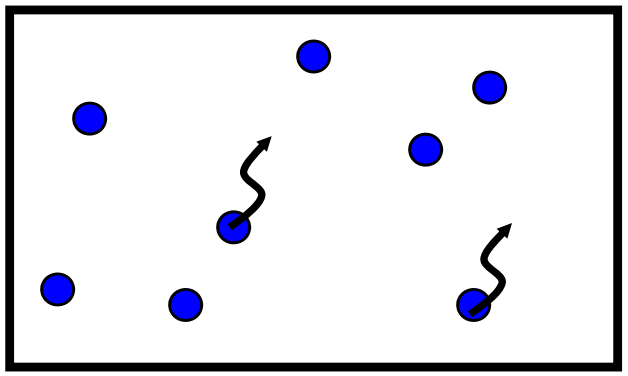
Global (derivative-free) Algorithms

- **Global solvers attempt broad exploration of the design space with a strategy for selective exploitation of promising regions**
- **Can be extremely costly, but will deliver good results when you have a large computational budget**
- **A few approaches:**
  - **Random (Monte Carlo) sampling**
  - **Multi-start local search**
  - **Box decomposition, e.g., DIRECT and other Lipschitzian approaches**
  - **Population search, e.g., genetic/evolutionary algorithms**
  - **Global surrogate-based algorithms**
  - ***For more see Cindy's talk on Friday, including meta-heuristics, e.g., simulated annealing, tabu search, ant colony optimization***

Random and Multi-start

- **Random sampling (Monte Carlo or more optimal space-filling designs) offers broad exploration; then just take the best point**
- **Can combine with local optimization by refining each of the set of promising optima using a local optimizer:**

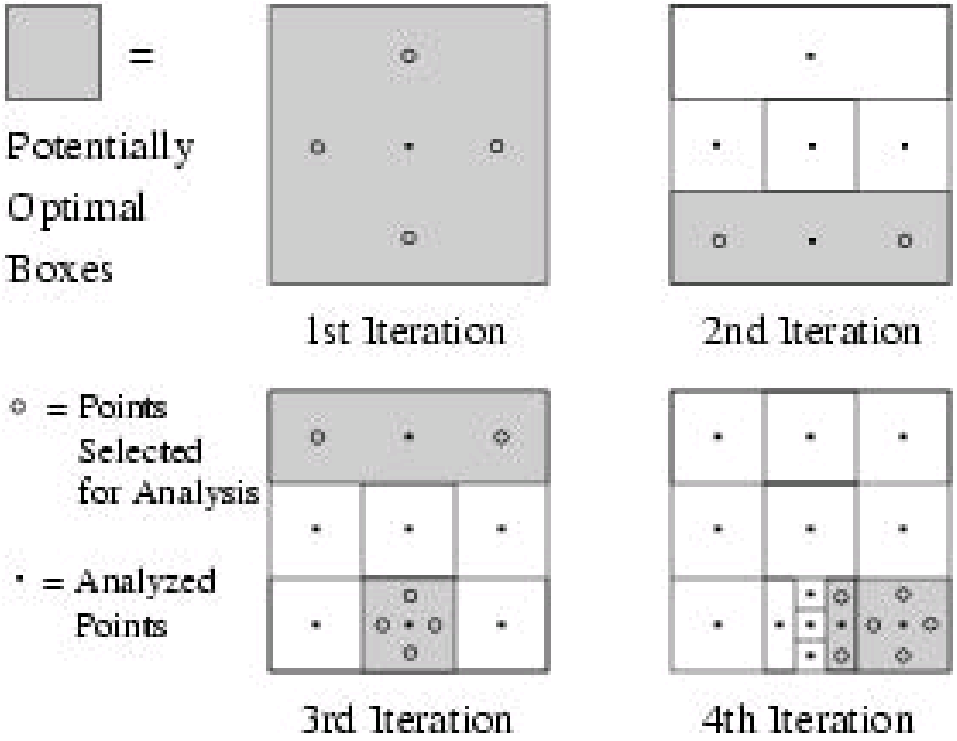




Sandia National Laboratories

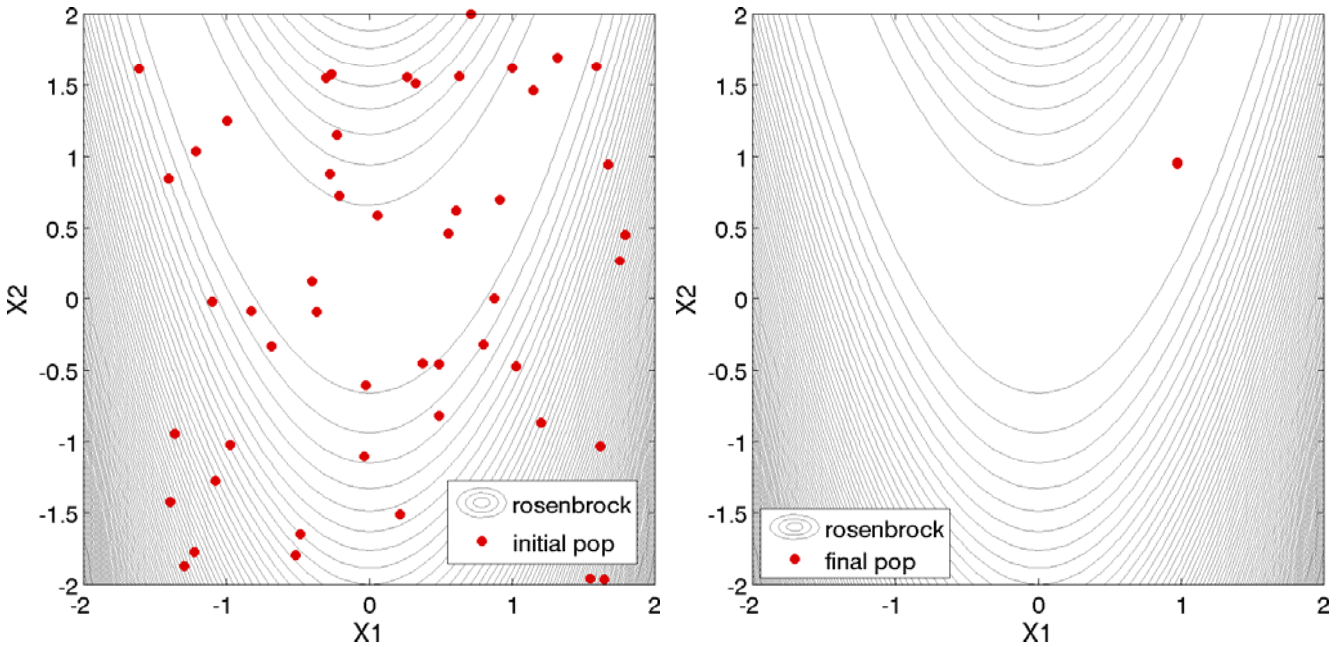
Division of RECTangles (DiRECT)

- Subdivide the search domain into non-overlapping ‘boxes’
- Boxes are ranked with estimates of their best value
- Boxes are selected for subdivision based on their rank and box size
- Successive refinement ensures that a near-optimal point will be found in finite time



Evolutionary/Genetic Algorithms

- Based on Darwin's theory of survival of the fittest
- Random initial population of design points
- Design parameters values are a unique "genetic string," analogous to DNA
- Sequence of generations, where most "fit" survive and reproduce
- Simulates natural selection, breeding, and mutation
- Ultimately identifies a design point (or family of points) satisfying optimization problem



Sandia National Laboratories

Advanced Considerations

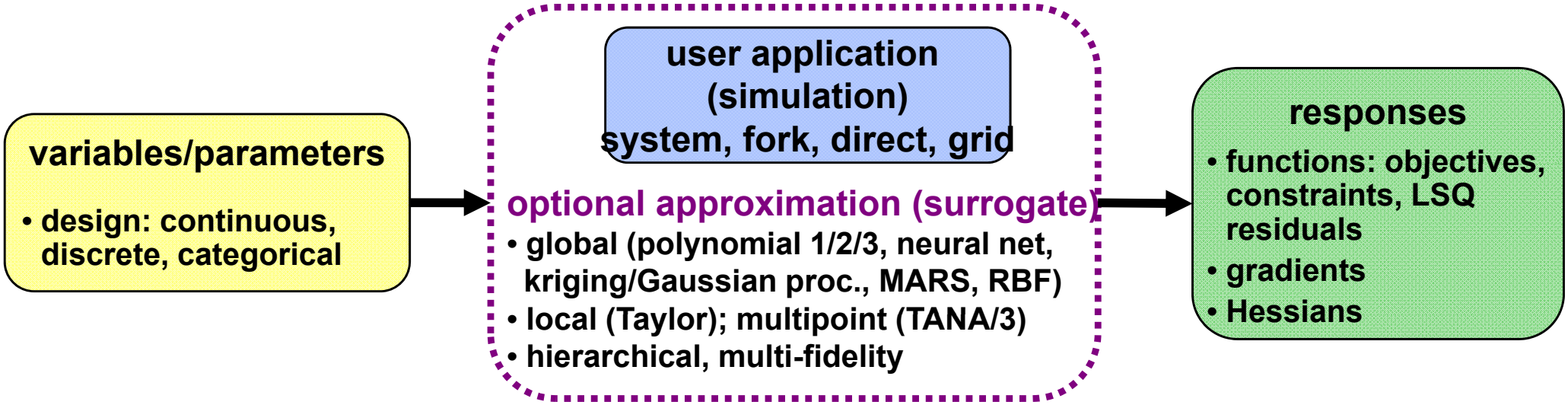
- **Mixed-integer nonlinear programming (MINLP): not discussed here, but see Cindy's talk Friday**
  - **Use branch and bound over integer variables**
  - **For each discrete scenario, apply any of the nonlinear optimization techniques discussed here**
- **Multi-objective (trade-off) optimization**
  - **Few solvers treat directly (however some GAs and other heuristics can map out the necessary Pareto frontier)**
  - **Explicit weighting of objectives: limits intuition, but allows use of any solver:**

$$f(x) = \sum_k \omega_k f_k(x)$$

- **Surrogate-based optimization (local and global)**
- **Uncertainty: optimization under uncertainty and robustness (uncertainty of optima)**

Surrogate-based Minimization (Calibration and Optimization)

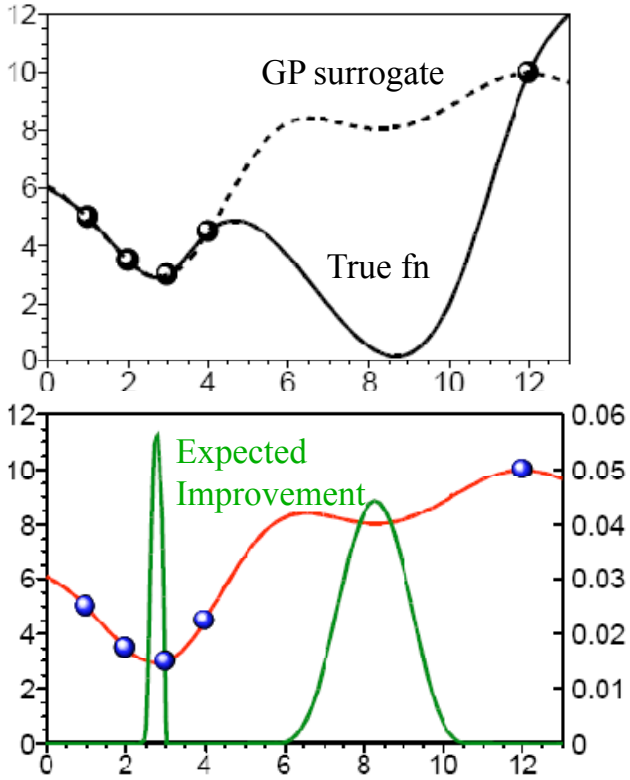
- Surrogate-based techniques replace or augment costly model evaluations with a less expensive stand-in; a key approach to make hard optimization problems tractable
- Response surface or meta-models are most common: use design of experiments to sample variable space and then build an approximation



- Multi-fidelity approaches useful when you have a physics-based surrogate, empirical approximation, or low-fidelity model option

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"



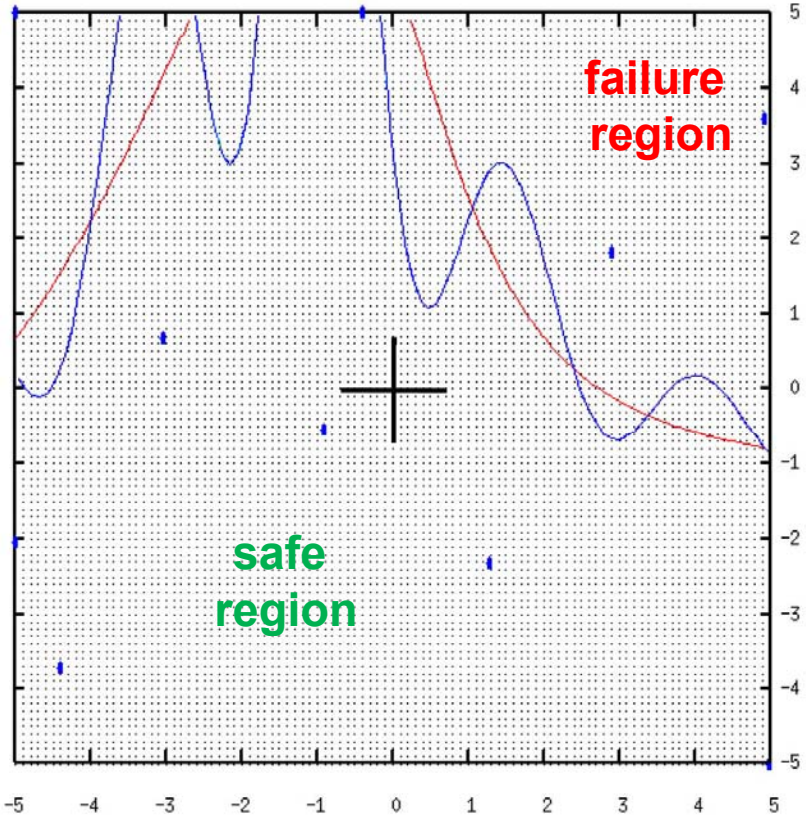
From Jones, Schonlau, Welch, 1998

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

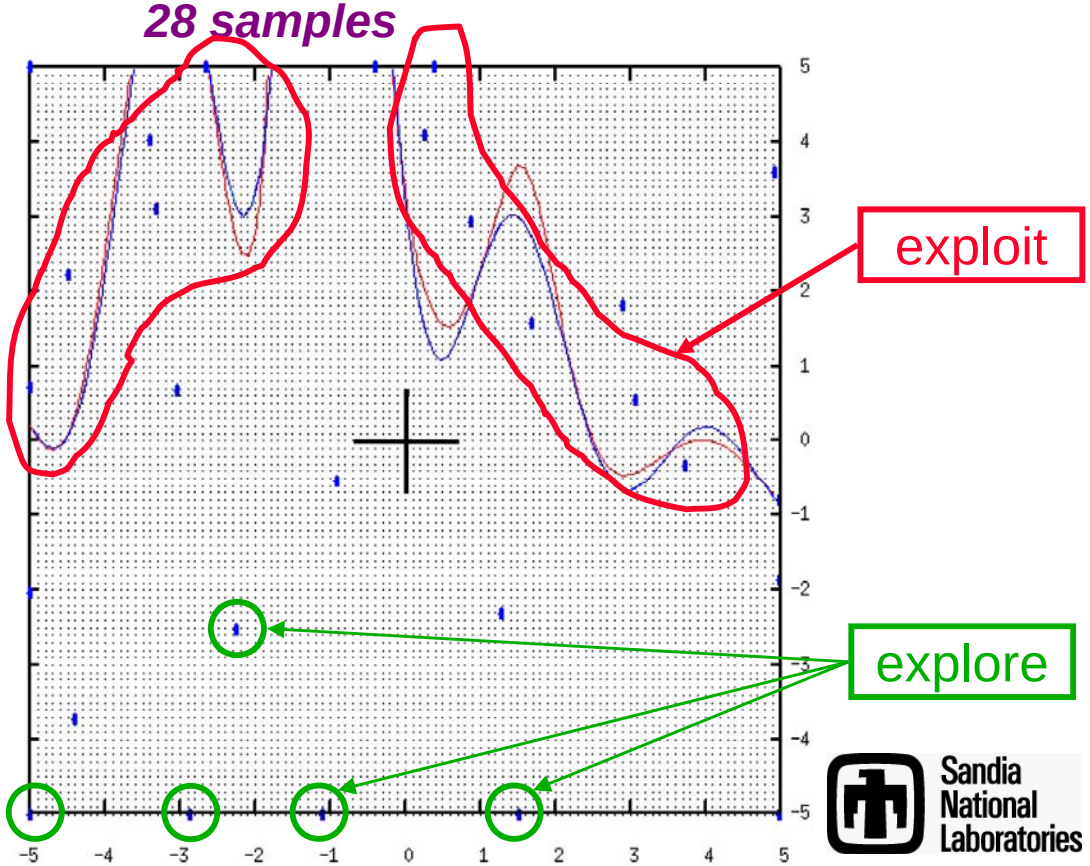
- Apply an EGO-like method to the equality-constrained optimization problem
- In EGRA, an expected feasibility function balances exploration with local search near the failure boundary to refine the GP
- Cost competitive with best MPP search methods, yet better probability of failure estimates; addresses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

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

10 samples

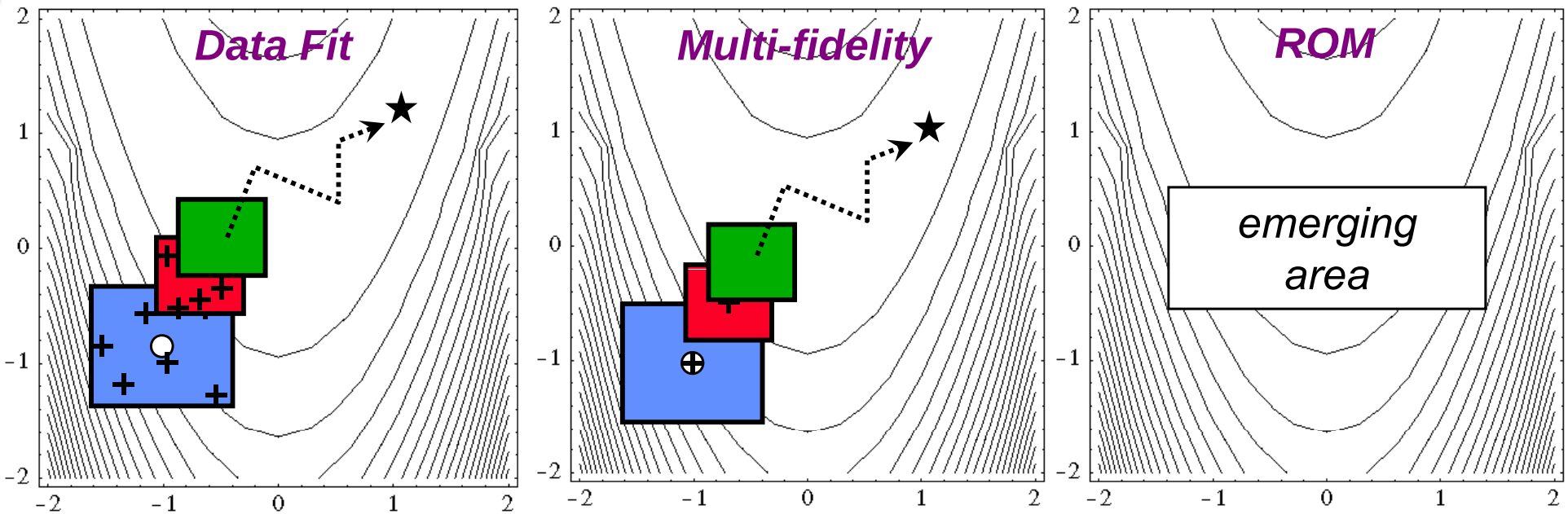


28 samples



Sandia National Laboratories

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)

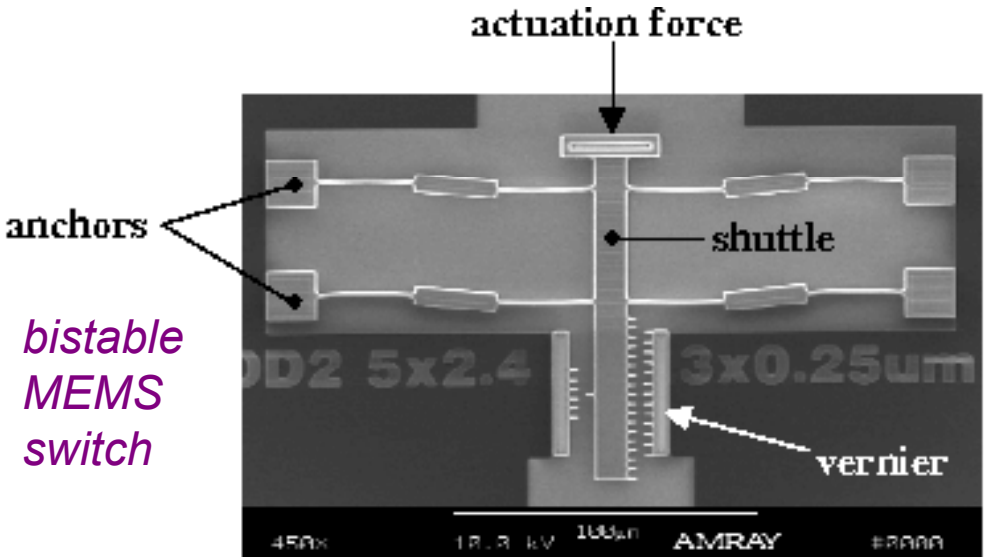
ROMs in SBO

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

Sandia National Laboratories

Shape Optimization of Compliant MEMS

- **Micro-electromechanical system (MEMS):** typically made from silicon, polymers, or metals; used as micro-scale sensors, actuators, switches, and machines
- **MEMS designs are subject to substantial variability** and lack historical knowledge base. Materials and micromachining, photo lithography, etching processes all yield uncertainty.
- Resulting part yields can be low or have poor cycle durability
- **Goal: shape optimize finite element model of bistable switch to…**
  - **Achieve prescribed reliability** in actuation force
  - Minimize sensitivity to uncertainties **(robustness)**



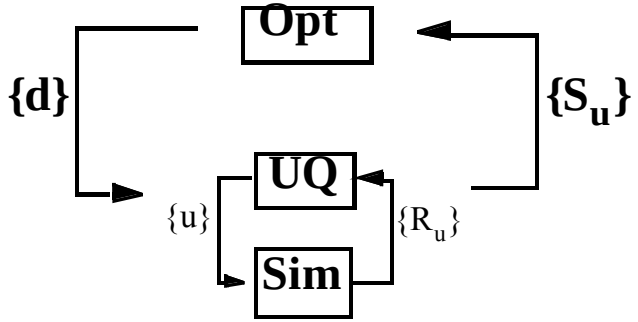
uncertainties to be considered (edge bias and residual stress)

variable	mean	std. dev.	distribution
Δw	-0.2 μm	0.08	normal
S_r	-11 Mpa	4.13	normal

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:



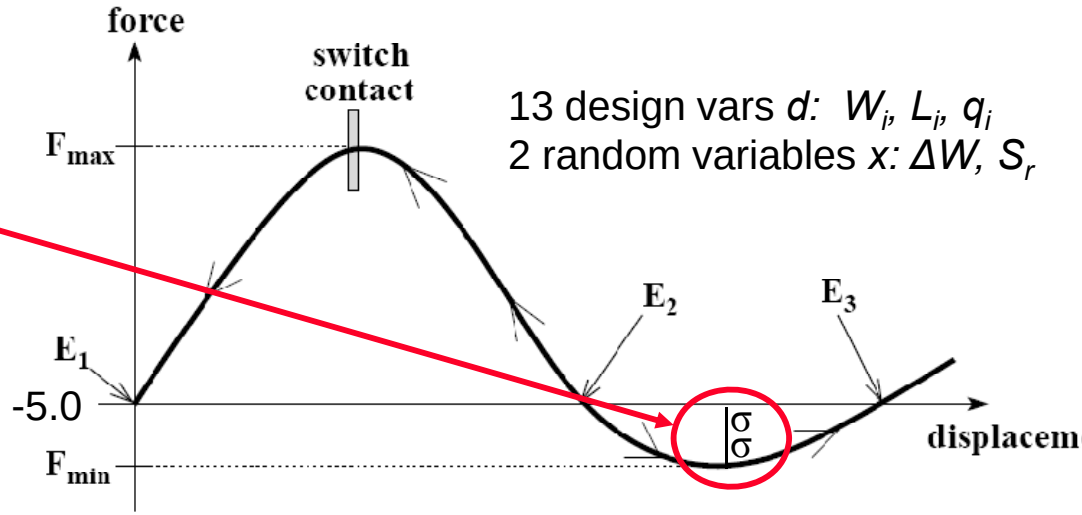
(nested paradigm)

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

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

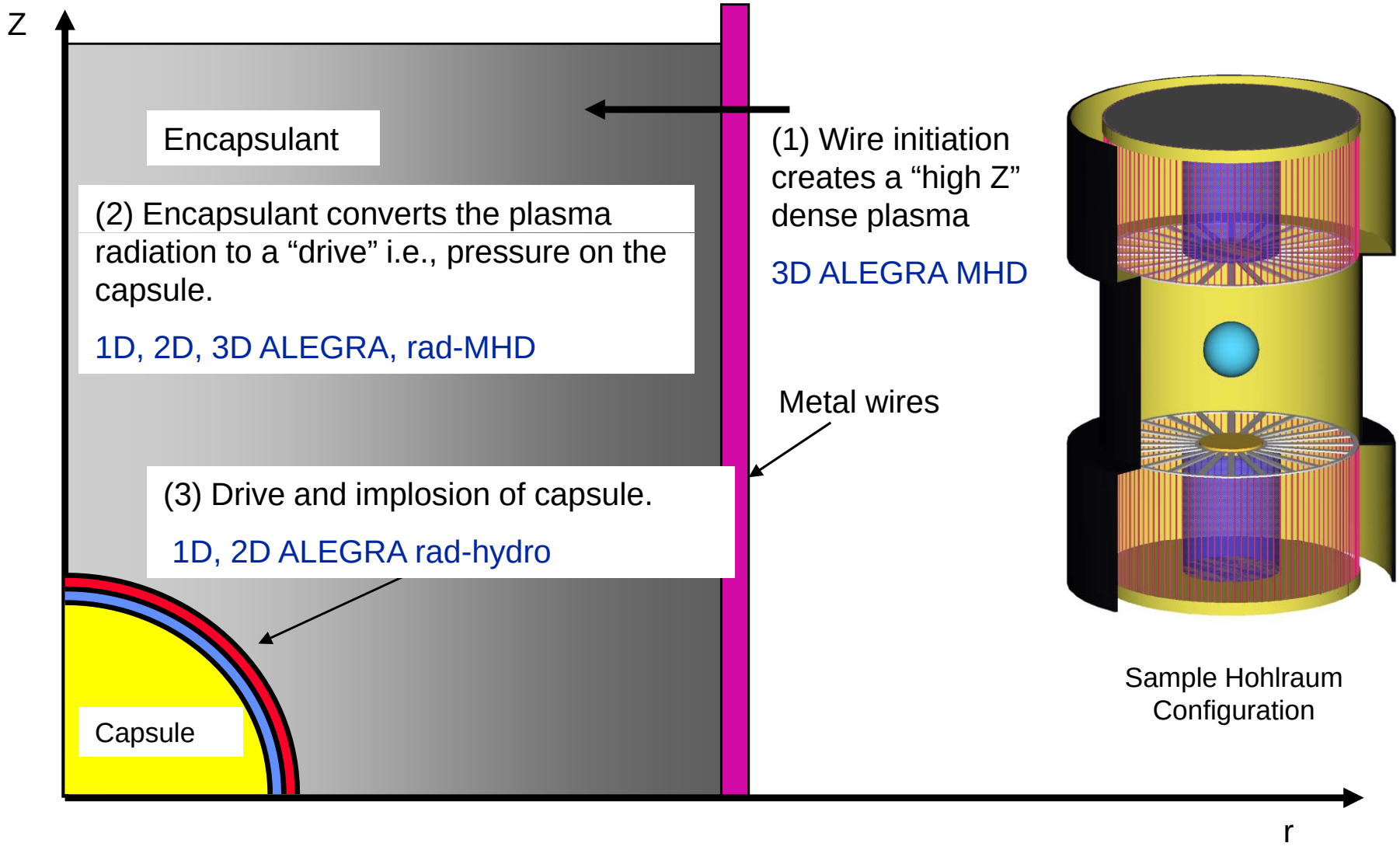
simultaneously reliable and robust designs

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



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

Robust Hohlraum Design for Inertial Confinement Fusion



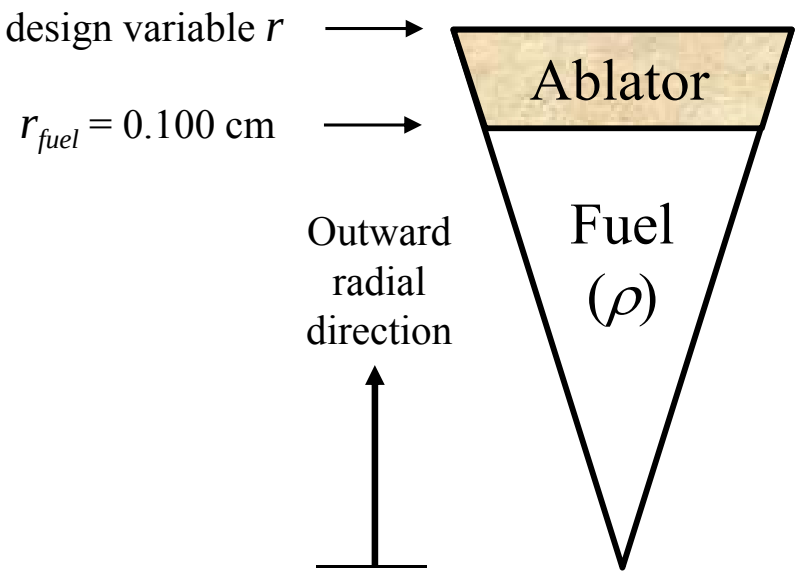
Sample Hohlraum Configuration

Uncertainties in plasma, drive, and capsule characteristics

Sandia National Laboratories

ICF Capsule Robust Design

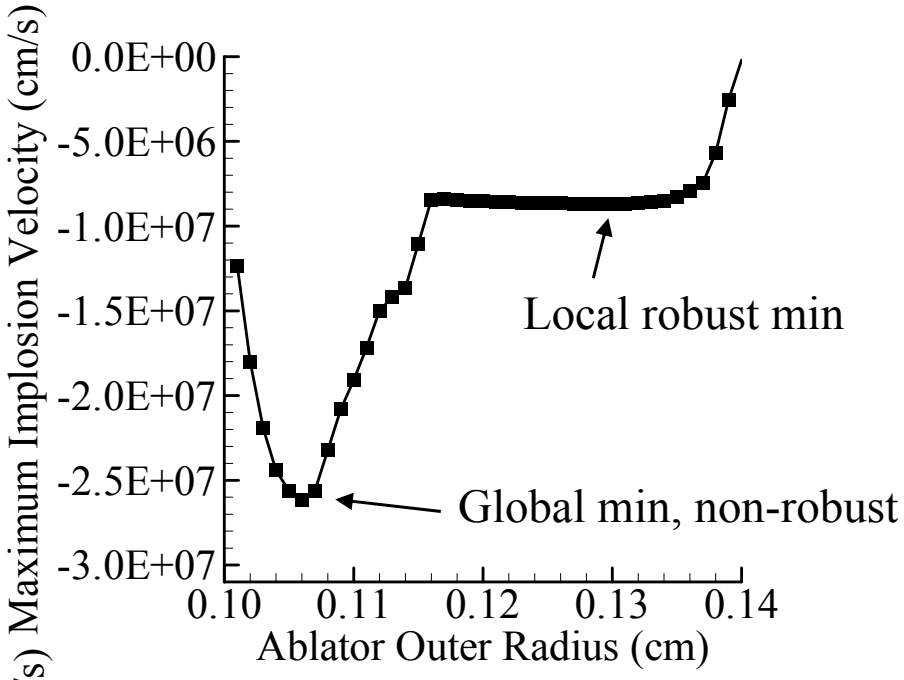
Design goal: maximize the implosion velocity w.r.t. ablator radius r and fuel density ρ, but remain robust w.r.t. manufacturing variability

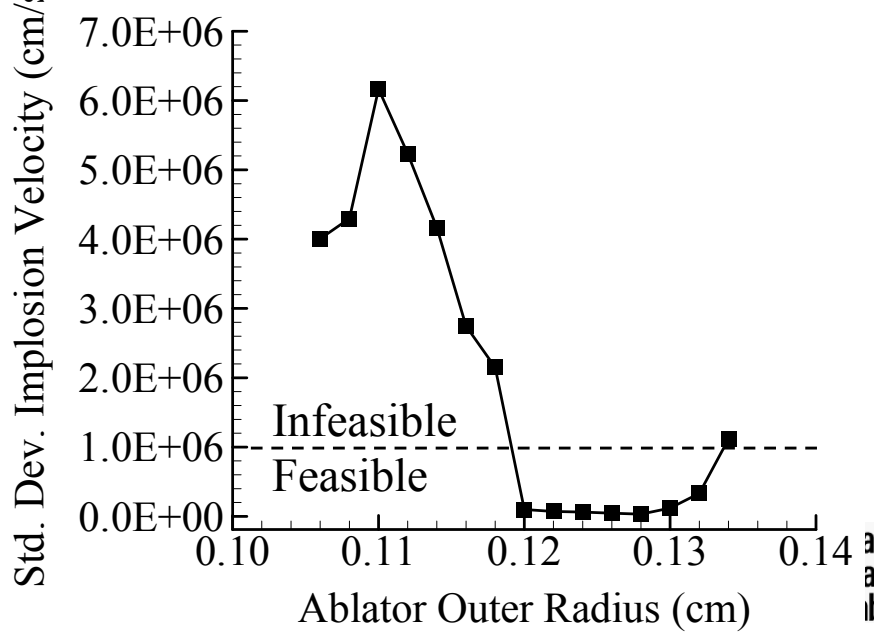


Minimize $V(r, \rho)$

Subject to $\sigma_V(r, \rho) \leq$ target value

uniform: +/- 2.5% range in r, ρ





andia
ational
boratories

Software Guides and Packages
Jumping Off Points Only

- *Optimization Software Guide*, Moré and Wright (1993)
- NEOS Optimization Software Guide
  http://www.neos-guide.org/NEOS/index.php/Optimization_Software_Guide
- Decision Tree for Optimization Software
  http://plato.asu.edu/guide.html

- Matlab, R, SAS, etc., optimization tools
- COIN-OR (DFO, IPOPT, etc.; http://coin-or.org)
- DAKOTA (http://dakota.sandia.gov)
- Acro/Coliny, CONMIN, DOT, HOPSPACK, JEGA, KNITRO, NGSA-II, NL2SOL, NLPQL, NPSOL, OPT++, SNOPT, SQP
- Algebraic modeling languages (AMPL, GAMS, AIMMS)

Discussion
Now or over Break

- What kinds of simulation-based optimization problems are you interested in solving?
  - Kind of variables, objectives, and constraints
  - Computational cost
  - Optimization goals
- What are the key challenges you face?
- What approaches tend to work well or have wide use (regardless of performance) in your field?

DAKOTA in a Nutshell

Design and Analysis toolKit for Optimization and Terascale Applications includes a wide array of algorithm capabilities to support engineering transformation through advanced modeling and simulation.

Adds value to simulation-based analysis by answering fundamental science and engineering questions:

- **What are the crucial factors/parameters and how do they affect key metrics? *(sensitivity)***
- **How safe, reliable, robust, or variable is my system? *(quantification of margins and uncertainty: QMU, UQ)***
- **What is the best performing design or control? *(optimization)***
- **What models and parameters best match experimental data? *(calibration)***

- ***All rely on iterative analysis with a computational model for the phenomenon of interest***

Sandia National Laboratories

Key DAKOTA Capabilities

- Generic interface to simulations
- Time-tested and advanced algorithms to address nonsmooth, discontinuous, multimodal, expensive, mixed variable, failure-prone
- Strategies to combine methods for advanced studies or improve efficiency with surrogates (meta-models)
- Mixed deterministic / probabilistic analysis
- Supports scalable parallel computations on clusters
- Object-oriented code; modern software quality practices
- Limited Windows interface (run via command prompt); however new graphical user interface. DART integration in progress.

- Additional details: http://dakota.sandia.gov/
  - Extensive documentation, including a tutorial
  - Support mailing lists
  - Software downloads: stable releases and nightly builds (freely available worldwide via GNU LGPL)

Sandia National Laboratories

DAKOTA Optimization Methods

Gradient-based methods

(DAKOTA will compute finite difference gradients and FD/quasi-Hessians if necessary)

- *DOT (various constrained)*
- CONMIN (FRCG, MFD)
- *NPSOL (SQP)*
- *NLPQL (SQP)*
- OPT++ (CG, Newton)

Calibration (least-squares)

- NL2SOL (GN + QH)
- *NLSSOL (SQP)*
- OPT++ (Gauss-Newton)

Derivative-free methods

- COLINY (PS, APPS, Solis-Wets, COBYLA2, EAs, DIRECT)
- JEGA (single/multi-obj GAs)
- EGO (efficient global opt via Gaussian Process models)
- DIRECT (Gablonsky)
- OPT++ (parallel direct search)

- *TMF (templated meta-heuristics framework)*

Sandia National Laboratories

Discussion
Now or over Break

- What kinds of simulation-based optimization problems are you interested in solving?
  - Kind of variables, objectives, and constraints
  - Computational cost
  - Optimization goals
- What are the key challenges you face?
- What approaches tend to work well or have wide use (regardless of performance) in your field?

Thank you for the invitation!

briadam@sandia.gov

http://dakota.sandia.gov/