SAND 2009-0463P

DAKOTA 101

Optimization

http://www.cs.sandia.gov/dakota

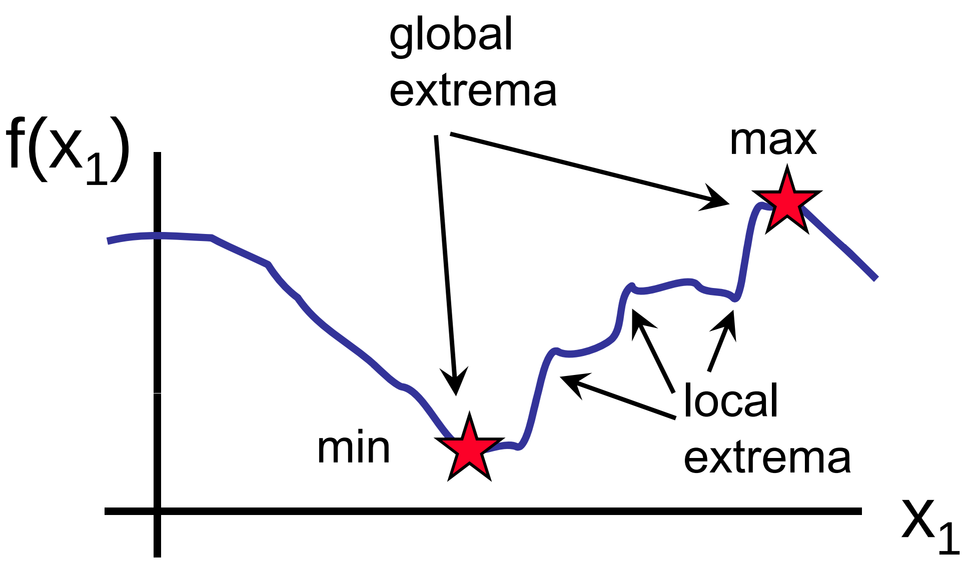
Learning Goals:

- **Understand basic goals of optimization**
- **Run DAKOTA examples of optimization methods**
- **Understand basic classes of problems and considerations for selecting and optimization method**

*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.

Optimization

- **GOAL: Vary parameters to extremize objectives, while satisfying constraints to find (or tune) the best design, estimate best parameters, analyze worst-case surety, e.g., determine:**
  - **delivery network maximizing profit / minimizing environ. impact**
  - **case geometry that minimizes drag and weight, yet is sufficiently strong and safe**
  - **AF&F with maximum design margin**
  - **material atomic configuration of minimum energy**
  - **class needs?**

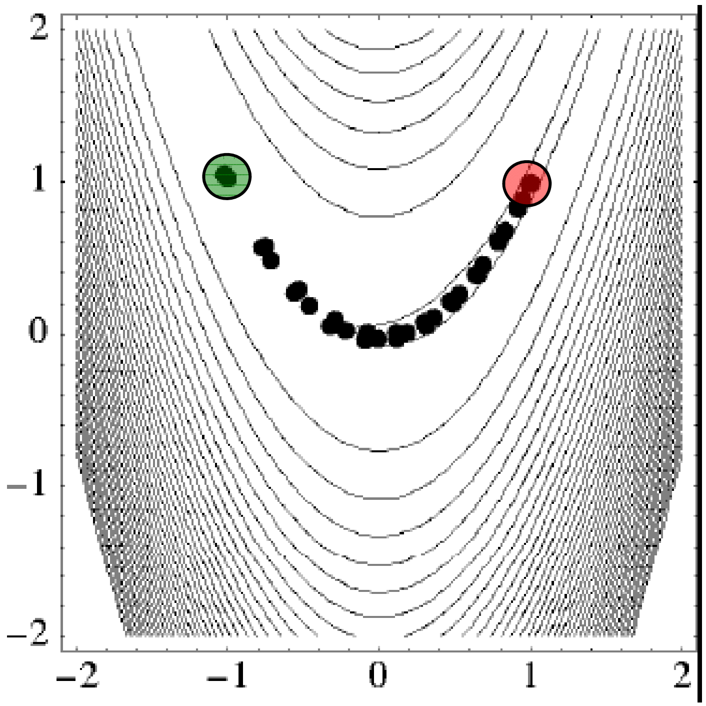


Some applications: local improvement suffices; others: must find global minimum at any cost

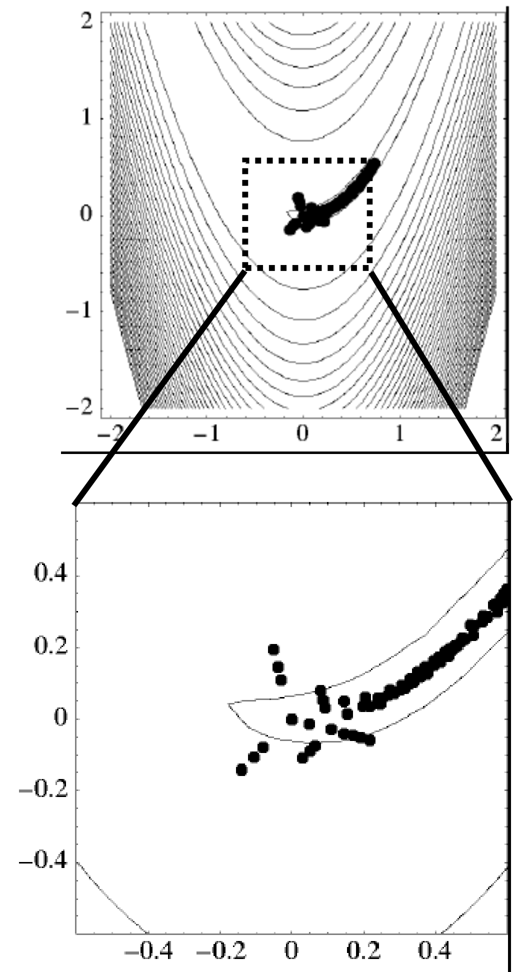
Sandia National Laboratories

Gradient-based Methods vs. Derivative-free Methods

Gradient-based search algorithm follows the general descent direction "around the bend"



One example of derivative-free search is stencil-based with expansion/contraction



Mileage may vary based on optimization approach

- **Use a gradient-based method to minimize Rosenbrock's function**
  - **Exercise: Create input file from template (Optimization Local GradientBased OPTPP) and run**
  - **Exercise: Modify input to use numerical gradients and rerun**
- **Use a derivative-free method to optimize Rosenbrock's function**
  - **Exercise: Create input file from scratch (use the APPS method) and run**

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

$-2 \leq x1 \leq 2$

$-2 \leq x2 \leq 2$

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

Classes of optimization problems and methods

- Constrained
  – Minimize an objective given constraints
  – Exercise: Use template Optimization Local Constrained GradientBased
- Multi-start local
  – Provide multiple starting points to a local optimizer to find multiple local minima
  – Exercise: Use template Optimization Local MultiStart
- Global
  – Find the global extreme value
  – Exercise: Use template Optimization Global Evolutionary Algorithm
- Multi-objective
  – Optimize across multiple competing objectives
  – Exercise: Use template Optimization Local MultiObjective (modify to use optpp_q_newton method)
- Surrogate-based/multifidelity
  – Reduce the computational cost (i.e., number of function evaluations) of optimization
  – Exercise: Use template Optimization Global Surrogate
- Hybrid
  – Use multiple optimization methods to solve a single problem
  – Exercise: Use template Optimization Hybrid Textbook Example

Considerations when Choosing an Optimization Method

Key considerations:

- Local and global sensitivity study data; trend and smoothness
- Simulation expense
- Constraint types present
- Goal: local optimization (improvement) or global optimization (best possible)

Unconstrained or bound-constrained problems:

- Smooth and cheap: nearly any method; 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)*

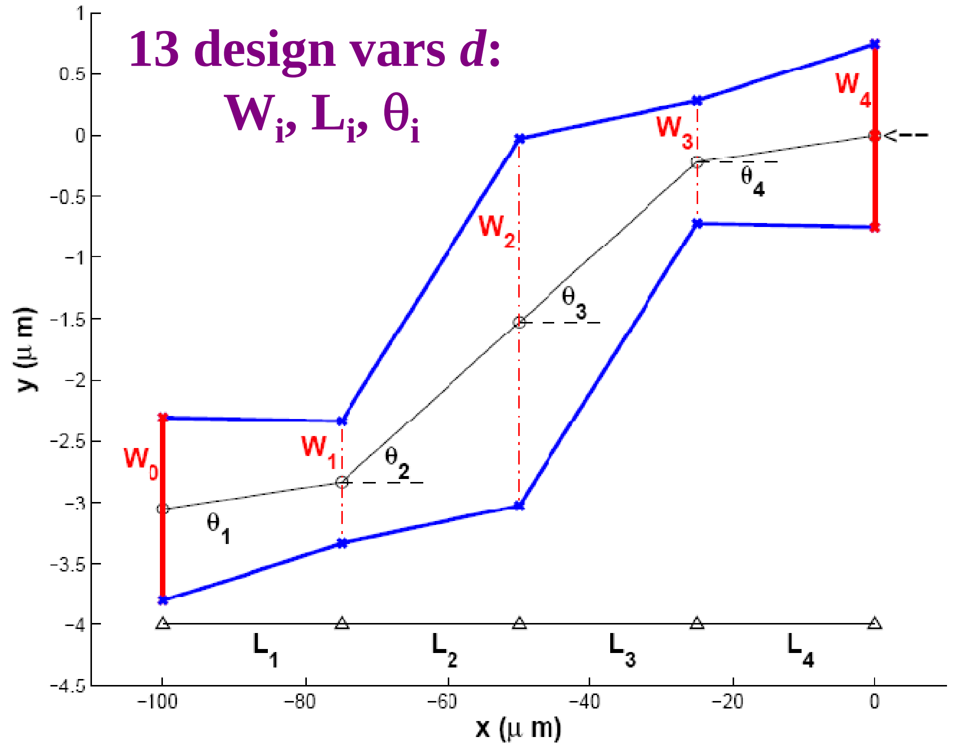
Nonlinearly-constrained problems:

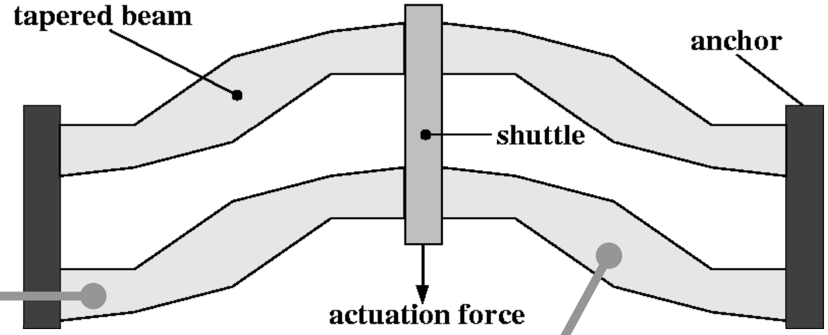
- Smooth and cheap: gradient-based methods
- Smooth and expensive: gradient-based methods
- Nonsmooth and cheap: non-gradient methods w/ penalty functions, SBO
- Nonsmooth and expensive: SBO

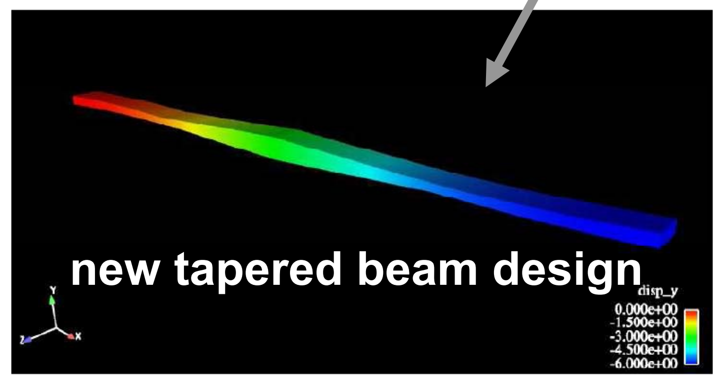
See guidance in User's Manual, Chapter 19

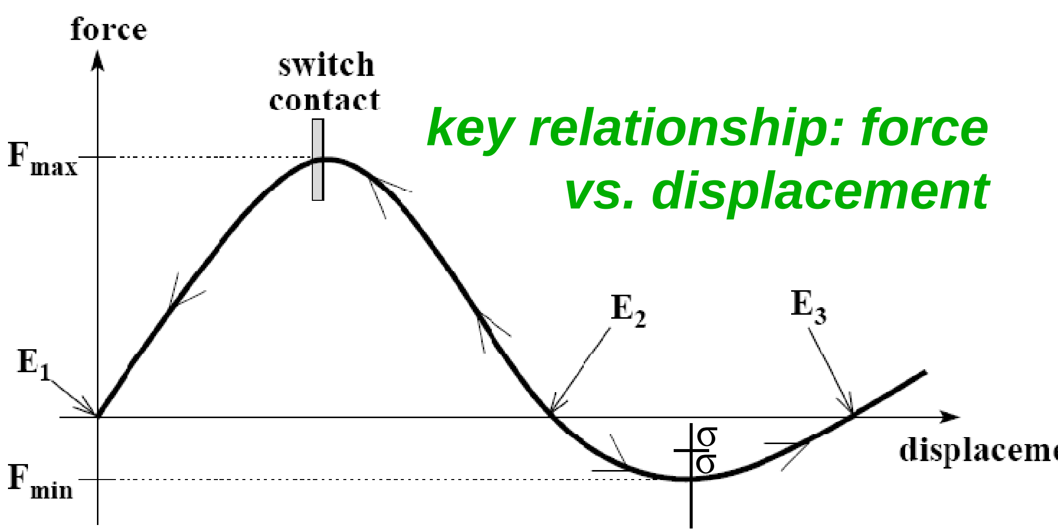
Backup Slides

MEMS Switch Design: Geometry Optimization







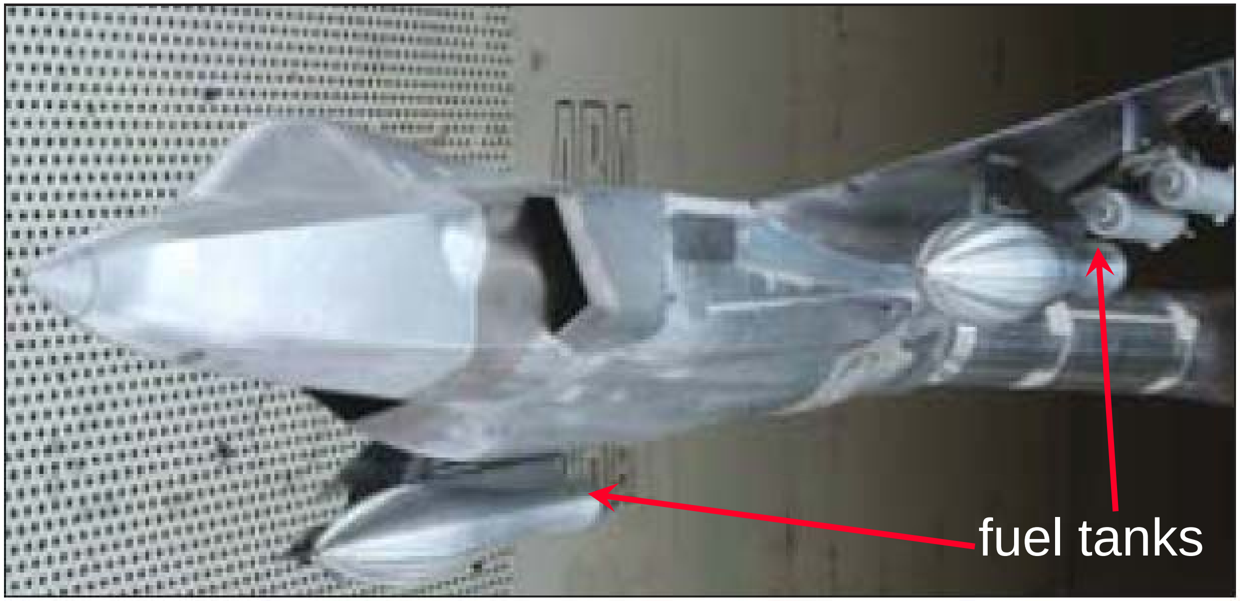


key relationship: force vs. displacement

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

Optimization Problem Formulation

Minimize:	$f(x_1, \ldots, x_N)$	***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 computed by or extracted from a simulation code)
		(Analytic metrics below are typically specified directly in a DAKOTA input deck)
	$A_I x \leq b_I$	***Linear inequality constraints***
	$A_E x = b_E$	***Linear equality constraints***
	$x_{LB} \leq x \leq x_{UB}$	***Bound constraints***

In practice, we can have multiple f-values in the objective function (aka "multiobjective optimization"), and multiple constraints of each type.

Sandia National Laboratories

Basic Constraint Lingo

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: the 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 applications*

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)*

Surrogate-based optimization

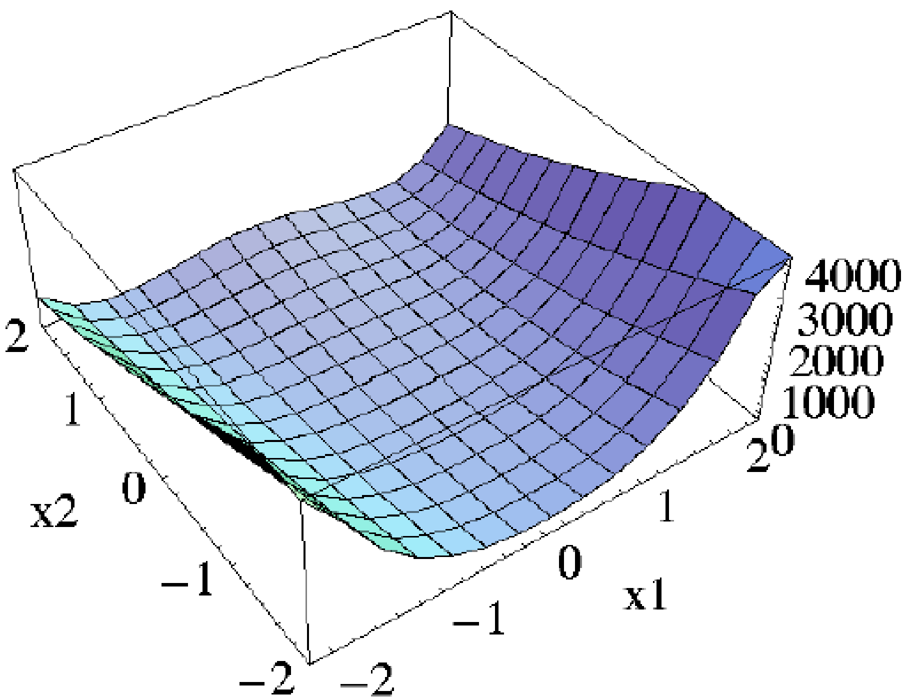
Efficient global optimization

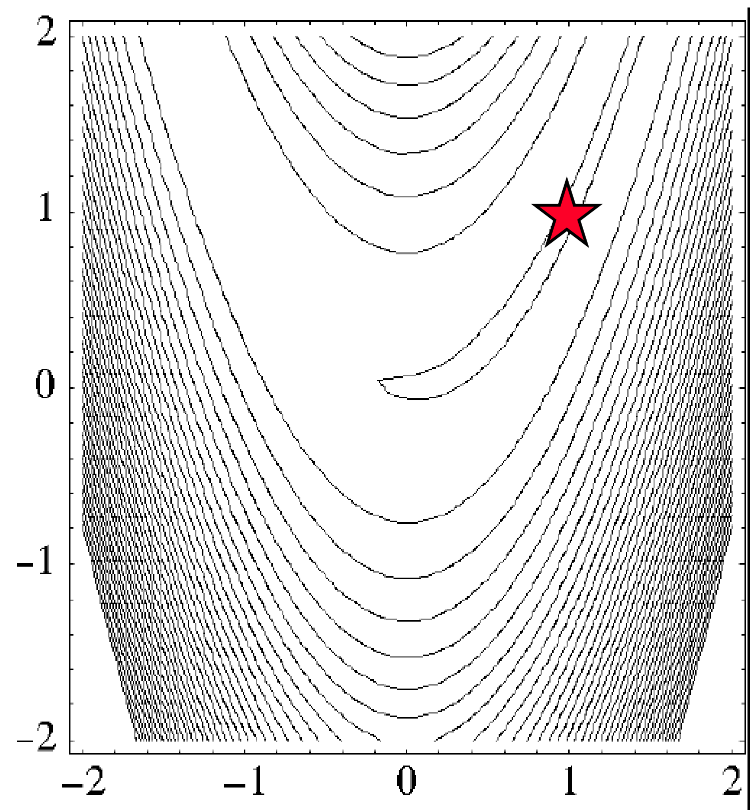
Optimization Learning Goals

Use optimization methods to find parameters yielding the best performing or minimum cost design. Or maximize agreement between simulation and experimental results (calibration).

- **Survey optimization terminology, problem formulations, and sample problems**
- **Understand considerations for selecting an optimization method**
- **Run DAKOTA examples of optimization methods**
  - **Gradient-based methods**
  - **Non-gradient pattern search and genetic algorithms**
  - **Constrained optimization**
- **Using least-squares solvers for model calibration (parameter estimation)**

Recall: Rosenbrock Function





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

s.t. $-2 \leq x1 \leq 2$

$-2 \leq x2 \leq 2$

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

Rosenbrock: Evolutionary Algorithm

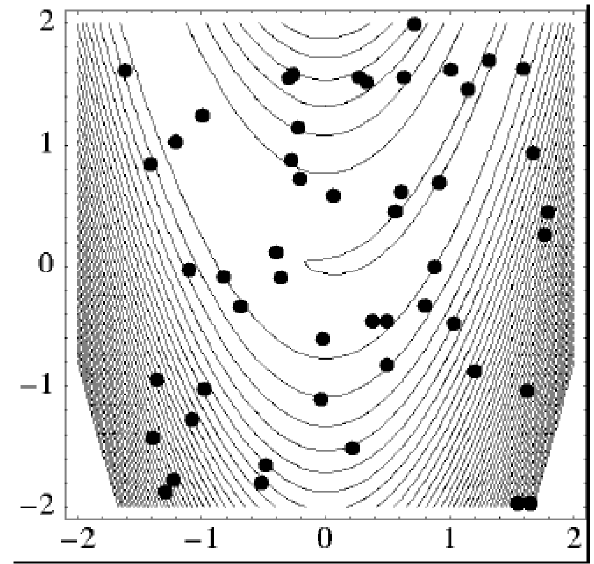
- **See examples/tutorial/ dakota_rosenbrock_ea_opt.in** **(Genetic algorithm (GA): non-gradient method, more global search than pattern search)**

- **Started with 50 random points in the parameter space; fitness, selection, reproduction**

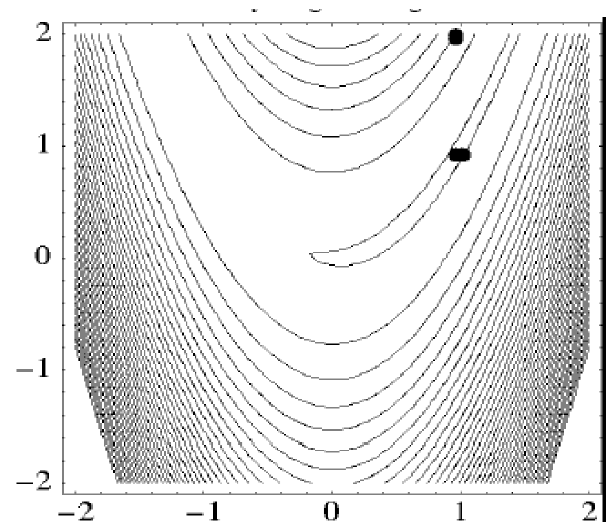
- **GA search algorithm run to generate 10,000 f-values. 46 of the 50 samples have settled close to the true optimum**

- **GAs and other global optimizers are great for problems with many local minima in which a gradient-based optimizer might get trapped.**

Initial population (50 random samples)

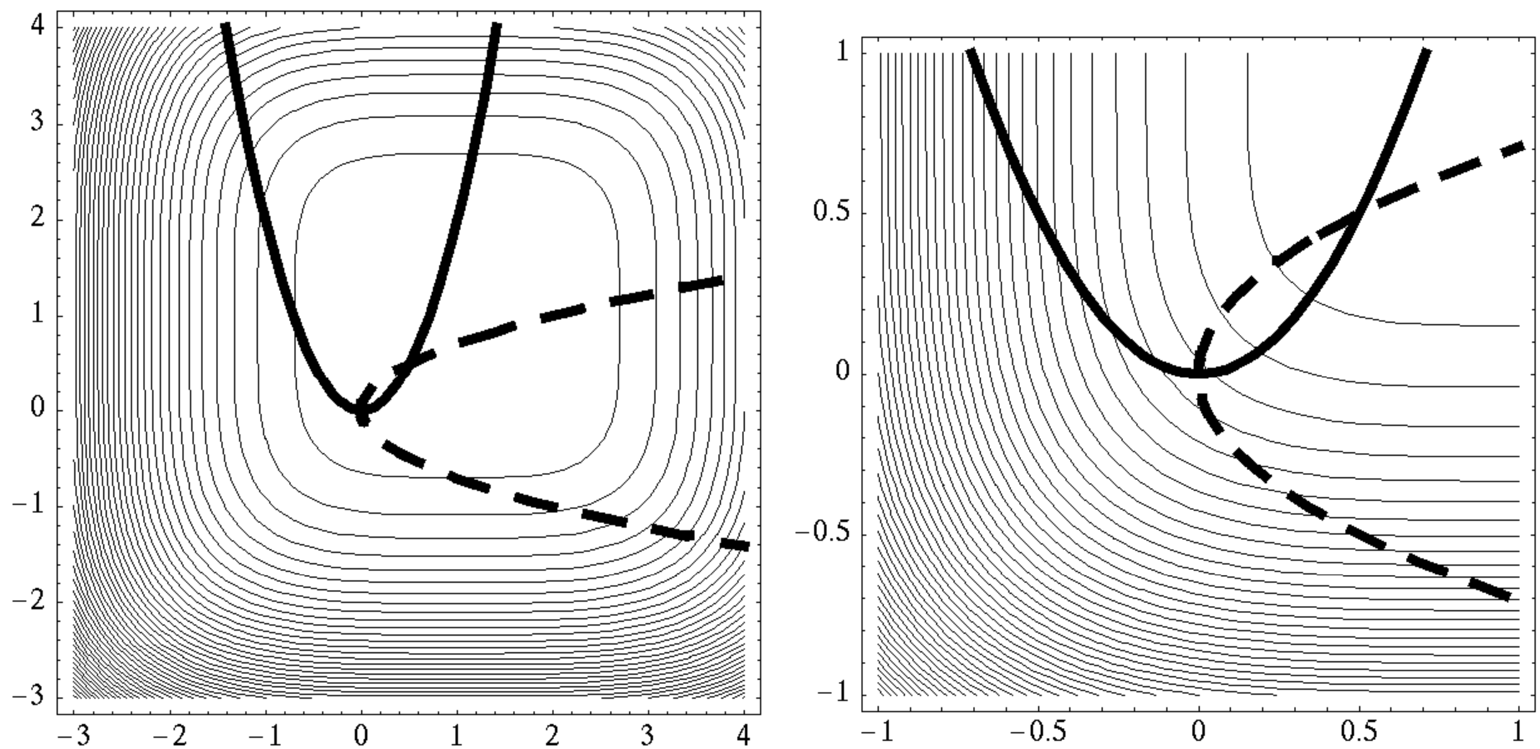


Final population (46 of 50 near minimum)



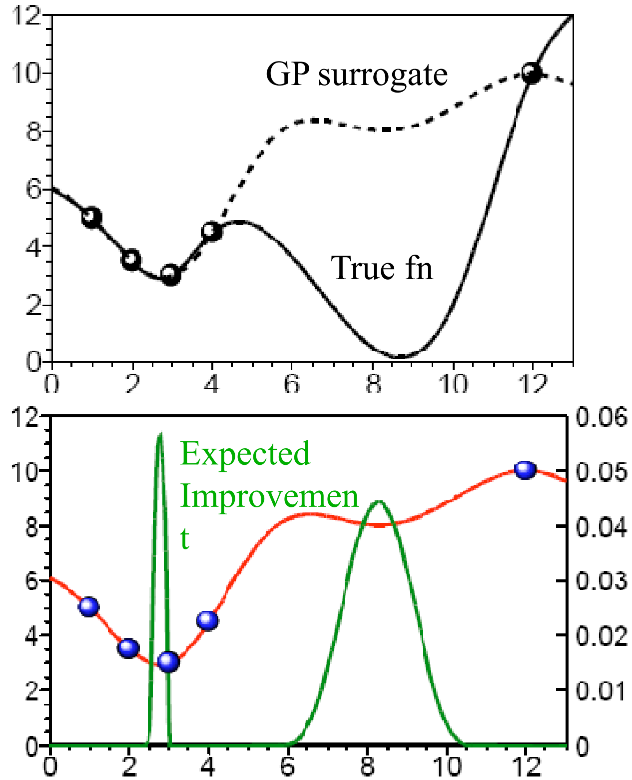
Gradient-based Constrained Optimization

- GOAL: Minimize, subject to nonlinearly-constrained feasible region (see textbook example, page 27 of User's Manual; 2.2).
- See examples/tutorial/dakota_textbook.in (see 2.4.4); notice constraints in input deck responses
- Modify to use fork interface with parameters_file and results_file, file_tag, file_save
- Inspect a results.out.x file to see the derivatives and constraints being returned to DAKOTA



Efficient Global Optimization

- **Technique due to Jones, Schonlau, Welch**
- **Balance global exploration with local optimality via expected improvement**

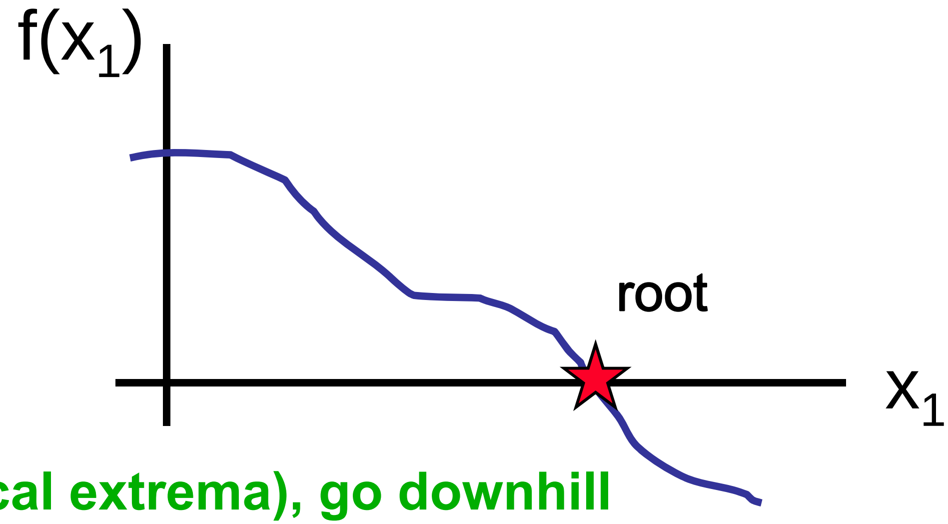


From Jones, Schonlau, Welch, 1998

Gradient-based Optimization

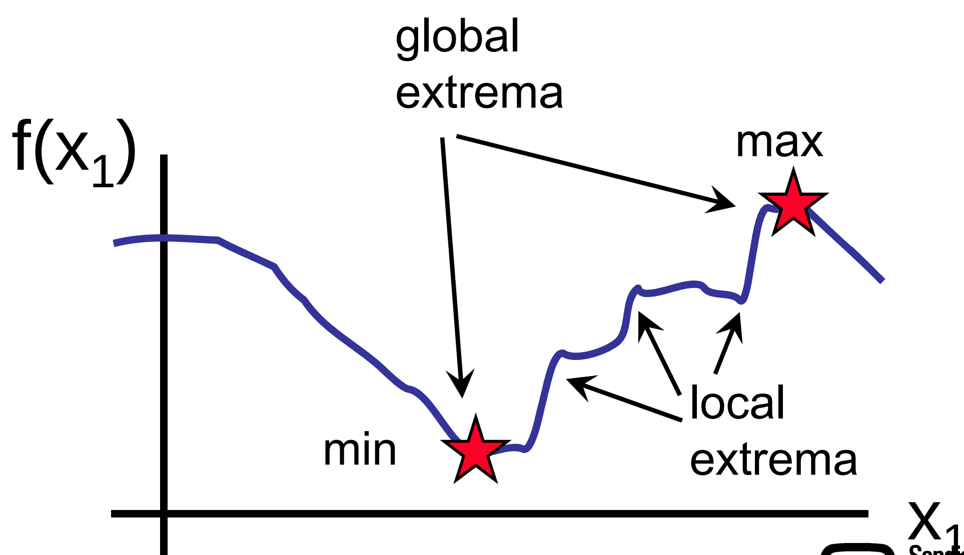
Modify Newton's root-finding method for solving f(x) = 0.

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



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

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



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

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

Gradient-based method with analytic gradient

```
method
 convergence_tolerance = 1e-10
 optpp q newton
variables
 continuous_design = 2
  initial_point = -1.2 1
  lower_bounds = -2 -2
  upper_bounds = 2 2
  descriptors = 'x1' 'x2'
interface
 analysis_drivers = 'rosenbrock'
  system
responses
 num_objective_functions = 1
 analytic_gradients
 no_hessians
```

```
*******************************************************
  *
                OPT++ TERMINATION CRITERION
      SUCCESS - optpp_q_newton converged to a solution
Algorithm converged - Difference in successive fcn
  values less than tolerance
*******************************************************
  *
<<<<< Function evaluation summary: 45 total (45 new, 0
  duplicate)
<<<<< Best parameters          =
                      1.0000000115e+00 x1
            1.0000000229e+00 x2
<<<<< Best objective function  =
                      1.3209463367e-16
<<<<< Best data captured at function evaluation 44

<<<<< Iterator optpp_q_newton completed.
<<<<< Single Method Strategy completed.
DAKOTA execution time in seconds:
  Total CPU        =      0.05 [parent =   0.052992,
  child =  -0.002992]
  Total wall clock =   0.559708
```

Gradient-based method with numerical gradient

```
method
 convergence_tolerance = 1e-10
 optpp q newton
variables
 continuous_design = 2
  initial_point = -1.2 1
  lower_bounds = -2 -2
  upper_bounds = 2 2
  descriptors = 'x1' 'x2'
interface
 analysis_drivers = ‘rosenbrock'
  system
responses
num_objective_functions = 1
 numerical_gradients
  dakota
  central
 no_hessians
```

```
*******************************************************
  *
               OPT++ TERMINATION CRITERION
    FAILURE - optpp_q_newtonterminated
Algorithm terminated - No longer able to compute step
  with sufficient decrease
*******************************************************
  *
<<<<< Function evaluation summary: 111 total (111 new,
  0 duplicate)
<<<<< Best parameters          =
                      9.9980325612e-01 x1
                      9.9960724889e-01 x2
<<<<< Best objective function  =
                      3.8756866561e-08
<<<<< Best data captured at function evaluation 103

<<<<< Iterator optpp_q_newton completed.
<<<<< Single Method Strategy completed.
DAKOTA execution time in seconds:
  Total CPU        =       0.15 [parent =   0.139979,
  child =   0.010021]
  Total wall clock =    1.84613
```

Derivative-free method

```
variables
 continuous_design = 2
  initial point = -1.2 1
  lower_bounds = -2 -2
  upper_bounds = 2 2
  descriptors = 'x1' 'x2'
interface
 analysis_drivers = 'rosenbrock'
  system
responses
 num_objective_functions = 1
 no_gradients
 no_hessians
method
 asynch_pattern_search
```

```
<<<<< Function evaluation summary: 27 total (27
  new, 0 duplicate)
<<<<< Best parameters          =
                      8.0000000000e-01 x1
                      6.2500000000e-01 x2
<<<<< Best objective function  =
                      6.2500000000e-02
<<<<< Best data captured at function evaluation
  21

<<<<< Iterator asynch_pattern_search completed.
<<<<< Single Method Strategy completed.
DAKOTA execution time in seconds:
  Total CPU        =       0.04 [parent =
  0.036994, child =   0.003006]
  Total wall clock =   0.396993
```

Constrained Optimization

```
strategy,
  single_method
    graphics,tabular_graphics_data
method,
  conmin_mfd,
    max_iterations = 50,
    convergence_tolerance = 1e-4
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,
  direct
    analysis_driver = 'text_book'
responses,
  num_objective_functions = 1
  num_nonlinear_inequality_constraints = 2
  numerical_gradients
    method_source dakota
    interval_type central
    fd_gradient_step_size = 1.e-4
  no_hessians
```

```
<<<<< Function evaluation summary: 59 total (48 new,
  11 duplicate)
<<<<< Best parameters          =
                      5.0000000000e-01 x1
                      4.3603021882e-01 x2
<<<<< Best objective function  =
                      1.6366338119e-01
<<<<< Best constraint values   =
                      3.1984890592e-02
                     -5.9877648280e-02
<<<<< Best data captured at function evaluation 54

<<<<< Iterator conmin_mfd completed.
<<<<< Single Method Strategy completed.
DAKOTA execution time in seconds:
  Total CPU        =       0.03 [parent =
  0.022996, child =   0.007004]
  Total wall clock =  0.0923381
```

Multi-start Local Optimization

```
strategy,
  multi_start graphics
  method_pointer = 'NLP'
  random_starts = 3 seed = 123
  starting_points = -.8  -.8
                    -.8   .8
                     .8  -.8
                     .8   .8
                      0.   0.
method,
  id_method = 'NLP'
  dot_bfgs
variables,
  continuous_design = 2
    lower_bounds    -1.0     -1.0
    upper_bounds     1.0    1.0
    descriptors      'x1'     'x2'
interface,
  system #asynchronous
    analysis driver = 'quasi sine fcn'
responses,
  num_objective_functions = 1
  analytic_gradients
  no_hessians
```

```
<<<<< Function evaluation summary: 34 total (34 new, 0 duplicate)
<<<<< Best parameters          =
                      3.5729239873e-02 x1
                      1.7709282199e-01 x2
<<<<< Best objective function  =
                      7.3776093493e-02
<<<<< Best data captured at function evaluation 185

<<<<< Iterator dot_bfgs completed.

<<<<< Concurrent iteration completed.

<<<<< Results summary:
   set_id            x1             x2            x1*            x2*
   obj_fn
        1          -0.8           -0.8  -0.8543728666  -0.8543728666
   0.5584096919
        2          -0.8            0.8  -0.9998398719    0.177092822
   0.291406596
        3           0.8           -0.8    0.177092822  -0.9998398719
   0.291406596
        4           0.8            0.8   0.1770928217   0.1770928217
   0.0602471946
        5             0              0  0.03572926375  0.03572926375
   0.08730499239
        6  -0.5356872085    -0.18209903  -0.5509956192  -0.1074790789
   0.3261816536
        7   0.1419702143  -0.7143527167   0.1770928312  -0.7024117488
   0.3030233398
        8   0.5240928925   0.7939232343  0.03572923987    0.177092822
   0.07377609349

DAKOTA execution time in seconds:
  Total CPU        =       0.24 [parent =   0.244963, child =  -0.004963]
  Total wall clock =    3.22742
```

Global Optimization

```
strategy,
  single_method
    graphics,tabular_graphics_data
method,
  coliny_ea
    max_iterations = 100
    max_function_evaluations = 2000
    seed = 11011011
    population_size = 50
    fitness_type merit_function
    mutation_type offset_normal
    mutation_rate 1.0
    crossover_type two_point
    crossover_rate 0.0
    replacement_type chc = 10
model,
  single
variables,
  continuous_design = 2
    lower_bounds    -2.0     -2.0
    upper_bounds     2.0      2.0
    descriptors      'x1'     "x2"
interface,
  system
    analysis_driver = 'rosenbrock'
responses,
  num_objective_functions = 1
  no_gradients
  no_hessians
```

```
<<<<< Function evaluation summary: 2010 total
  (2010 new, 0 duplicate)
<<<<< Best parameters          =
                      9.7663371881e-01 x1
                      9.5371909000e-01 x2
<<<<< Best objective function  =
                      5.4687292509e-04
<<<<< Best data captured at function
  evaluation 1972

<<<<< Iterator coliny_ea completed.
<<<<< Single Method Strategy completed.
DAKOTA execution time in seconds:
  Total CPU        =       4.85 [parent =
  4.85626, child =  -0.006262]
  Total wall clock =     167.46
```

Multi-Objective Optimization

```
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
```

```
********************************************************
           OPT++ TERMINATION CRITERION
        SUCCESS - optpp_q_newton converged to a solution
Algorithm converged - Difference in successive fcn values less than
   tolerance
********************************************************
<<<<< Function evaluation summary: 16 total (16 new, 0 duplicate)
       obj_fn_1: 9 val (9 n, 0 d), 8 grad (8 n, 0 d), 0 Hess (0 n, 0
   d)
       obj_fn_2: 9 val (9 n, 0 d), 8 grad (8 n, 0 d), 0 Hess (0 n, 0
   d)
       obj_fn_3: 9 val (9 n, 0 d), 8 grad (8 n, 0 d), 0 Hess (0 n, 0
   d)
<<<<< Best parameters          =
                      5.9388388497e-01 x1
                      7.4158025710e-01 x2
<<<<< Best objective functions =
                      3.1661673866e-02
                     -1.8092059720e-02
                      2.5299933524e-01
<<<<< Best data captured at function evaluation 15

<<<<< Iterator optpp_q_newton completed.
<<<<< Single Method Strategy completed.
DAKOTA execution time in seconds:
  Total CPU        =       0.02 [parent =   0.020997, child =  -
   0.000997]
  Total wall clock =   0.171464
```

Surrogate-Based Optimization

```
strategy,
    single_method
    tabular_graphics_data
    method_pointer = 'SBGO'
method,
    id_method = 'SBGO'
    surrogate_based_global
    model_pointer = 'SURROGATE'
    approx_method_pointer = 'MOGA'
    max_iterations = 5
    replace_points
    output verbose
method,
    id_method = 'MOGA'
    moga
model,
    id_model = 'SURROGATE'
    surrogate global
        responses_pointer = 'SURROGATE_RESP'
        dace_method_pointer = 'SAMPLING'
        correction additive zeroth_order
     gaussian_process
variables,
    continuous_design = 3
        initial_point      0     0     0
        upper_bounds       4     4     4
        lower_bounds      -4    -4    -4
        descriptors      'x1'  'x2'  'x3'
responses,
    id_responses = 'SURROGATE_RESP'
    num_objective_functions = 2
    no_gradients
    no_hessians
method,
    id_method = 'SAMPLING'
    model_pointer = 'TRUTH'
    nond_sampling
      samples = 100
      seed = 531
      sample_type lhs
      all variables
model,
    id_model = 'TRUTH'
    single
      interface_pointer = 'TRUE_FN'
      responses_pointer = 'TRUE_RESP'
interface,
    id_interface = 'TRUE_FN'
    system
       analysis_driver = 'mogatest1'
responses,
    id_responses = 'TRUE_RESP'
    num_objective_functions = 2
    no_gradients
    no_hessians
```

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 = 'PS2'
  model_pointer = 'M1'
  max_function_evaluations = 10
  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
```

```
*****************************************************
  *
           OPT++ TERMINATION CRITERION
         SUCCESS - optpp_newton converged to a
  solution
Algorithm converged - Difference in successive fcn
  values less than tolerance
*****************************************************
  *
<<<<< Function evaluation summary (I1): 39 total (39
  new, 0 duplicate)
       obj_fn: 14 val (14 n, 0 d), 14 grad (14 n, 0
  d), 13 Hess (13 n, 0 d)
<<<<< Best parameters          =
                      1.0000853639e+00 x1
                      1.0000774619e+00 x2
<<<<< Best objective function  =
                      8.9104436231e-17
<<<<< Best data captured at function evaluation 1249

<<<<< Iterator optpp_newton completed.

<<<<< Sequential Hybrid Optimizer Strategy completed.
DAKOTA execution time in seconds:
  Total CPU        =       0.31 [parent =   0.307953,
  child =   0.002047]
  Total wall clock =    0.73759
```