SAND2012-7388P

DAKOTA 101

DAKOTA Overview

http://dakota.sandia.gov/

DAKOTA 101
Course Goals

DAKOTA 101: Introduction to DAKOTA capabilities, application use cases, understanding its relevance for you. Understand major classes of algorithms, how to choose from among them, and create DAKOTA studies to execute them.

One day, 8:00—11:45, 1:30—5:00

- **8:00 Overview of DAKOTA**
- **9:30 Parameter Studies and Sensitivity Analysis**
- **10:45 Uncertainty Quantification**
- **1:30 Optimization**
- **2:45 Calibration / Parameter Estimation from Data**
- **4:00 Wrap-up: best practices and teasers for advanced interfacing and algorithms**

12:00—1:15: DAKOTA Brown Bag with experienced DAKOTA users.

Sandia National Laboratories

Learning Goals: DAKOTA Overview

- Learn design and analysis processes DAKOTA supports; assess relevance in your domain
- Survey key algorithm classes available in DAKOTA; what differentiates it
- Requirements and process for getting started
- Tour the mechanics of running DAKOTA
- Know where to get help

DAKOTA Supports

- Simulation-based engineering design: optimize virtual (computational) prototypes
- Risk analysis and quantification of margins and uncertainty (QMU): assess the effect of parametric uncertainty on the probability of achieving desired system performance
- Verification and validation: automate mesh convergence or solver tolerance studies, generate ensembles of possible simulations or statistics to compare to experimental data

DAKOTA in a Nutshell

DAKOTA includes a wide array of algorithm capabilities to support engineering transformation through advanced modeling and simulation.

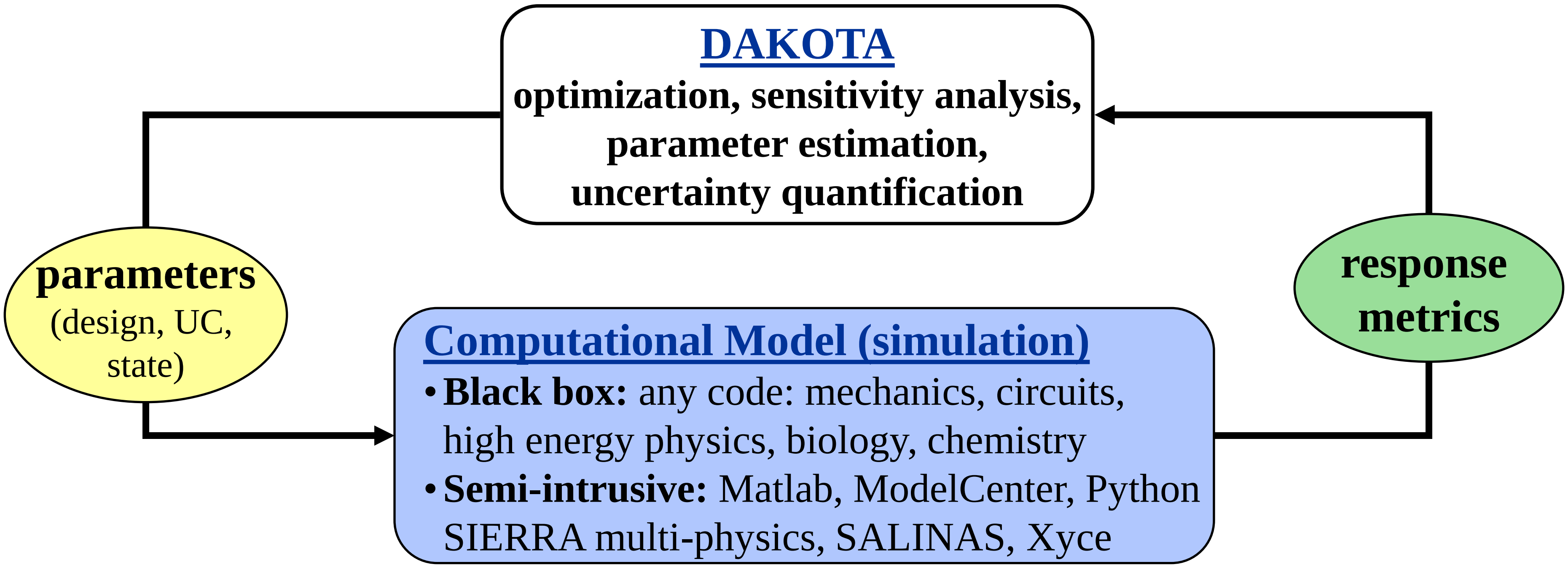
Adds value to simulation-based analysis by answering fundamental science and engineering questions:

- **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***

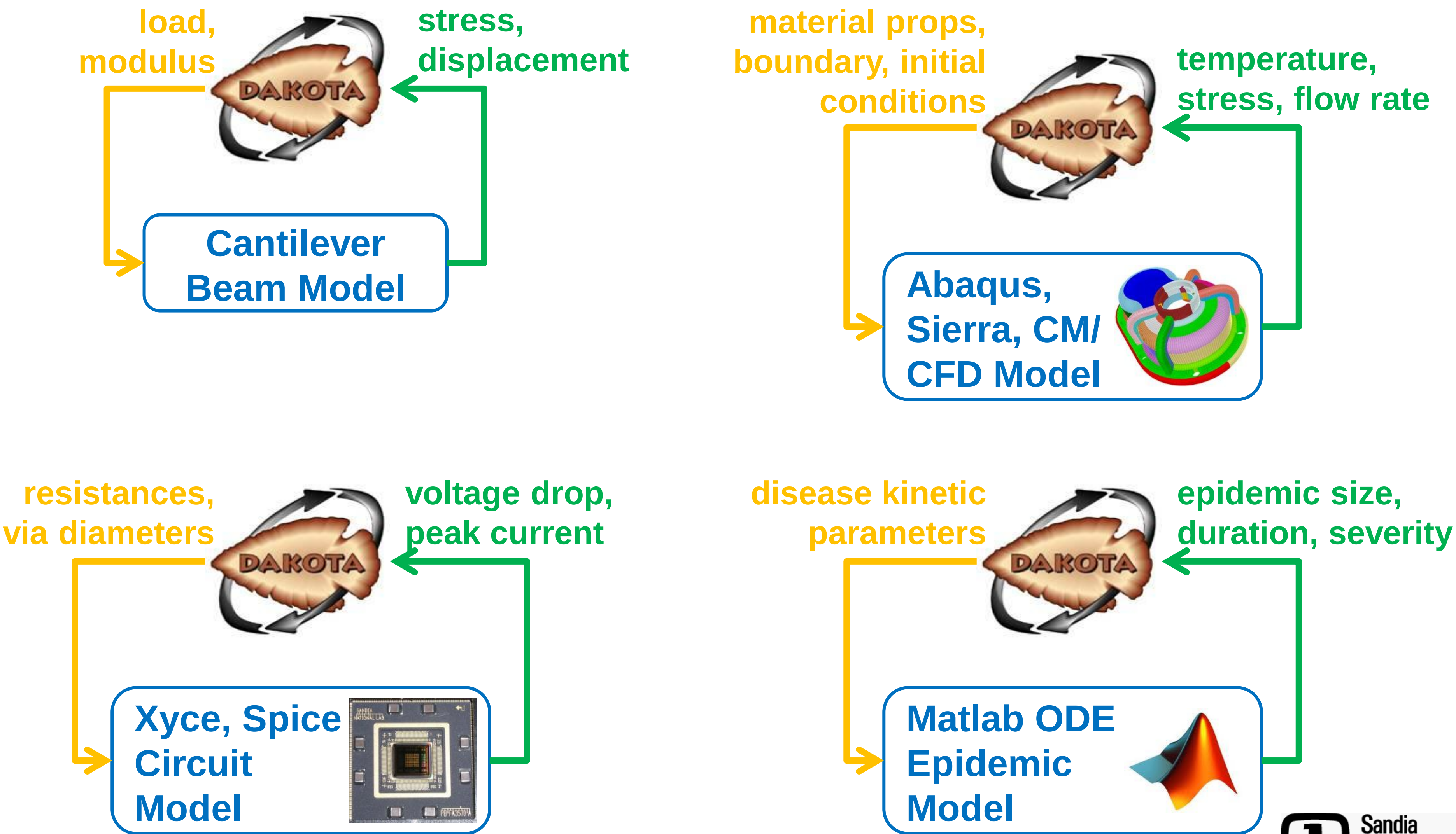
Automated Iterative Analysis of Computational Models

Automate typical "parameter variation" studies with various advanced methods and a generic interface to your simulation



- **Can support experimental testing: examine many accident conditions with computer models, then physically test a few worst-case conditions.**

DAKOTA Analysis: Iterating over Parameters of Computational Models



Simulation Challenges DAKOTA Addresses

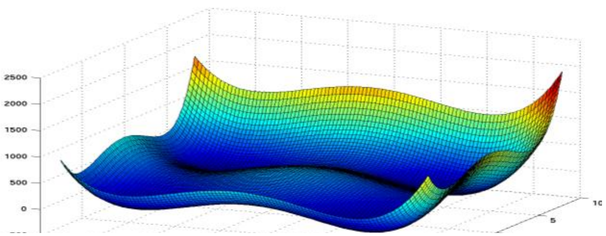
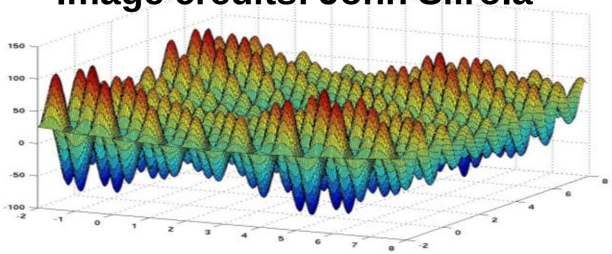
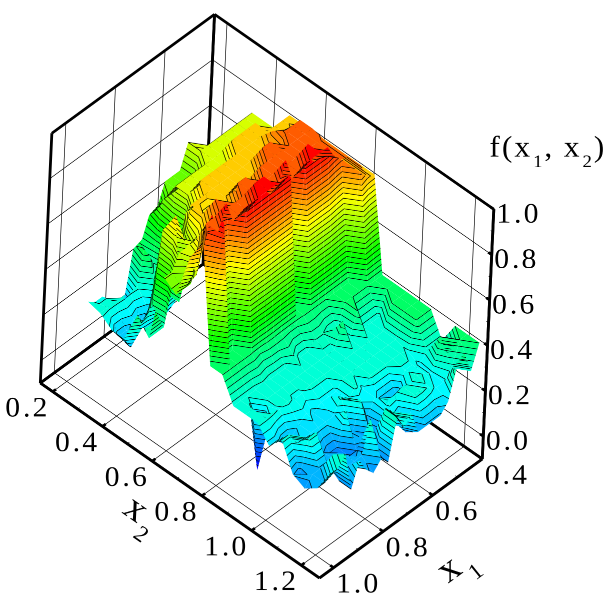


Image credits: John Siirola





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.

Key DAKOTA Capabilities

Largely Addressed in Advanced Topics

- **Generic interface to simulations: parameters in, responses out**
- **Time-tested and advanced research algorithms to address challenging science and engineering simulations**
- **Strategies to combine methods for advanced studies or improve efficiency with surrogates**
- **Mixed deterministic / probabilistic analysis**
- **Scalable parallel computations from desktop to clusters**
- **Primarily command-line and text-based interface; JAGUAR DAKOTA GUI and DART Workbench ease creating studies**
- **Object-oriented code; modern software quality practices**

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

DAKOTA in (Simplified) Simulation-based Design Process

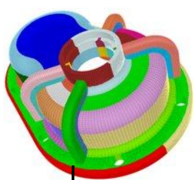
requirements, goals

design concept, physical model

computational model

analysis, insight, alternatives

build, test, evaluate



DAKOTA embedded in design refinement process to

- Tune model parameters to get desired behavior or match data
- Do simulation/design space exploration; explore alternative possibilities
- Validate designs/alternatives with their uncertainty

Parameterized by mesh/geometry, materials, equations

Small Group Discussion: DAKOTA Relevance

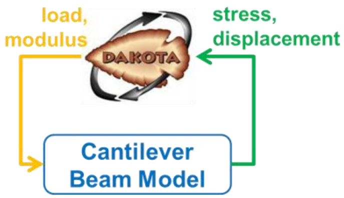
- **Discuss your initial impressions of DAKOTA's relevance for your problems**
- **With what kinds of applications, simulations, or computational models would you use it**
- **On what kinds of computer architecture would you want to use it**
  (desktop workstation, Windows laptop, high-performance compute cluster)

Steps to Using DAKOTA

1. **Define analysis goals; understand how DAKOTA helps, learn about and select from possible methods**

2. **Access DAKOTA and understand help resources**

3. **Workflow: create an automated workflow so DAKOTA can communicate with your simulation**
   - **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**



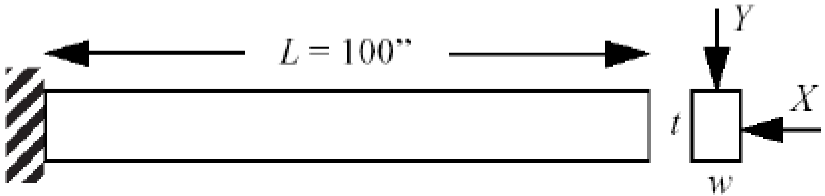
built-in cantilever beam analysis

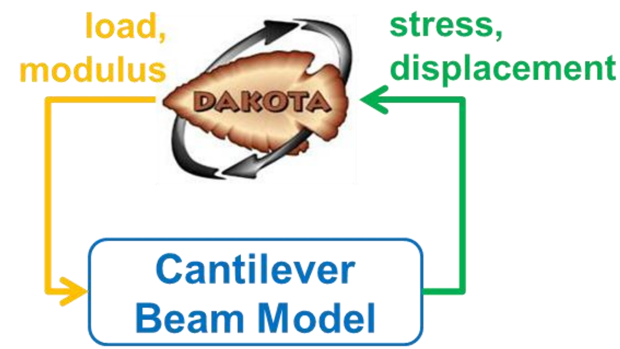
4. **DAKOTA input file: Jaguar GUI or text editor to configure DAKOTA to exercise the workflow to meet your goals**
   - **Tailor variables, methods, responses to analysis goals**
   - **Syntax documented in Reference Manual**

5. **Run DAKOTA: command-line; text input / output**

Create DAKOTA input file for cantilever beam analysis…

Cantilever Beam Analysis Problem





- **Beam computational model:**

  weight (*area = w*t*)

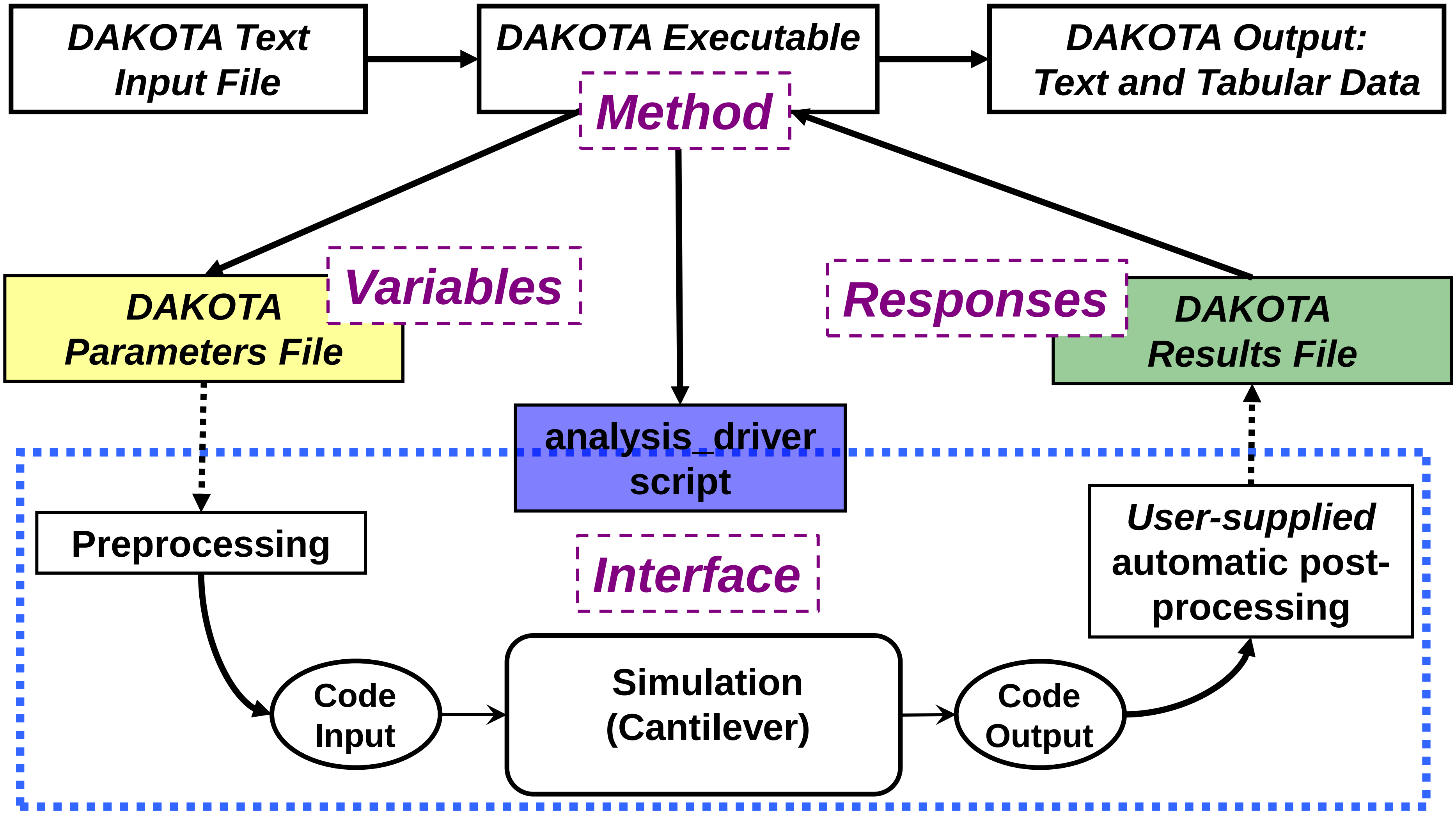
$$stress = \frac{600}{wt^2}Y + \frac{600}{w^2 t}X \le R$$

$$displacement = \frac{4L^3}{Ewt}\sqrt{\left(\frac{Y}{t^2}\right)^2 + \left(\frac{X}{w^2}\right)^2} \le D_0$$

Given values of *w, t, R, E, X, Y,* DAKOTA's `mod_cantilever` driver computes *area, stress-R, displacement-D_0*

- **Definitions:**
  - **t: thickness** 1.0 ≤ t ≤ 4.0
  - **w: width** 1.0 ≤ t ≤ 4.0
  - **R: yield stress** R ~ Normal(40000, 2000)
  - **E: Young's modulus** E ~ Normal(2.9e7, 1.45e6)
  - **X: horizontal load** X ~ Normal(500, 100)
  - **Y: vertical load** Y ~ Normal(1000, 100)
- **Constants: L = 100in and D_0 = 2.2535 in (max displacement)**

DAKOTA Execution & Info Flow



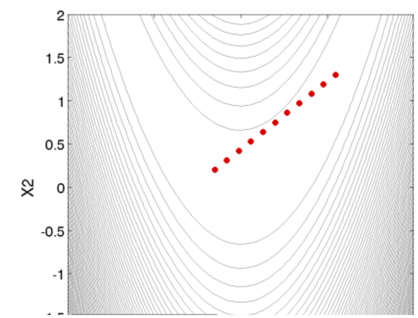
What are examples of these for the cantilever beam problem?

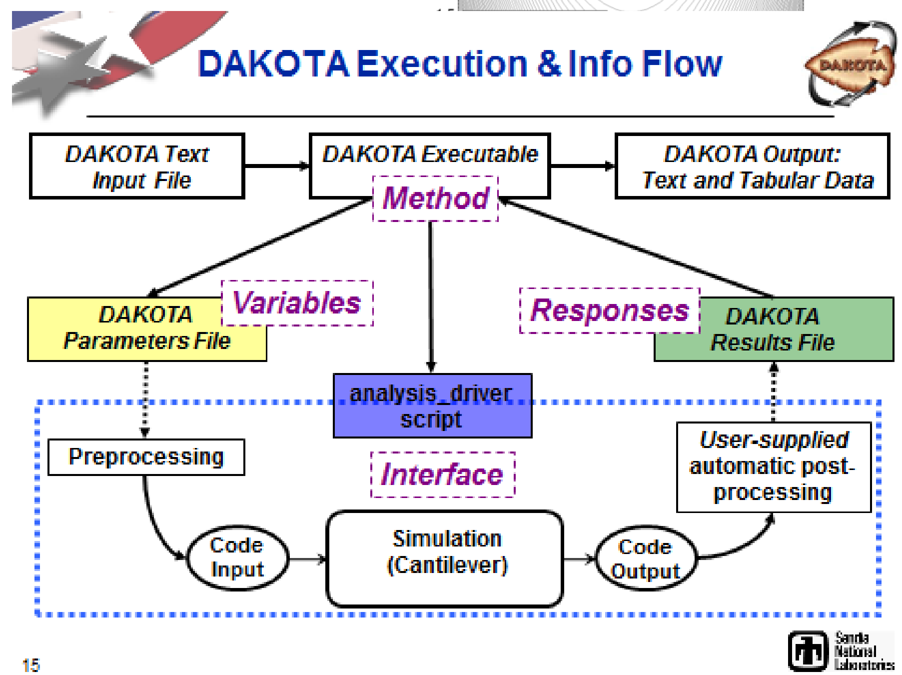
Configure DAKOTA Input File for a Vector Parameter Study

```
strategy
  single_method
  graphics, tabular_graphics_data

method
  vector_parameter_study
  num_steps = 10
  final_point  4.0 4.0 40000. 29.E+6 500. 1000.

variables
  continuous_design = 2
    initial_point  1.0          1.0
    descriptors    'beam_width' 'beam_thickness'
  continuous_state = 4
    initial_state  40000. 29.E+6 500. 1000.
    descriptors    'R'    'E'    'X'   'Y'

interface
  direct
  analysis_driver = 'mod_cantilever'

responses
  num_objective_functions = 3
  descriptors = 'area' 'stress' 'displacement'
  no_gradients
  no_hessians
```

(cantilever_vector.in)

Define Flow / Algorithm





Define Problem / Mapping

Sandia National Laboratories

DAKOTA Input File: 4 Required, 2 Optional Sections

```
strategy
  single_method
  graphics, tabular_graphics_data

method
  vector_parameter_study
  num_steps = 10
  final_point  4.0 4.0 40000. 29.E+6 500. 1000.

variables
  continuous_design = 2
    initial_point  1.0          1.0
    descriptors    'beam_width' 'beam_thickness'
  continuous_state = 4
    initial_state  40000. 29.E+6 500. 1000.
    descriptors    'R'    'E'    'X'   'Y'

interface
  direct
  analysis_driver = 'mod_cantilever'

responses
  num_objective_functions = 3
  descriptors = 'area' 'stress' 'displacement'
  no_gradients
  no_hessians
```

(cantilever_vector.in)

strategy (opt): coordinate hybrid methods and high level settings

method (req): specify iterative algorithm and settings

Define Flow / Algorithm

variables (req): parameters input to simulation; different types, domains for different studies

interface (req): map from variables to responses; control parallelism

responses (req): model output(s) to be studied

Define Problem / Mapping

model (opt): single, surrogate, nested

Dakota Execution and Output

- **DAKOTA is most commonly run from a UNIX or Windows command prompt, also in Jaguar**

- **Capture output: input variable and response information for each function evaluation; method-specific info**

```
>> dakota –i my_run.in –o my_run.out
```

- **`strategy tabular_graphics_data:` generates tabular listing of inputs and outputs, called `dakota_tabular.dat`. Useful for Excel, Matlab, or other package import.**

- ***Other command-line options***

```
>> dakota –help
```

JAGUAR DAKOTA GUI

- Graphical user interface for creating, editing, and running DAKOTA input files (alternative to text editor)
- Java; based on Eclipse IDE/Workbench
- Windows, Mac, Linux support
- Beta integration in DART Workbench

Key Features:

- Templates for common studies (most common starting point)
- Synchronized auto-completing text and hierarchical graphical editors
- Error checking and integrated help
- Sensitivity analysis, parameterization, and DAKOTA study wizards
- Basic visualization
- At SNL: job submission to clusters

Sandia National Laboratories

Getting Started and Getting Help

- **Access a Sandia installation:** `module avail dakota`
  AMECH (CA), CEE (ESHPC/SCICO, NM), Computer clusters (both)
  or download

- **Supported on Linux/Unix, Mac OS X, Windows (no MinGW or Cygwin install required)**

- **Tour DAKOTA web pages: http://dakota.sandia.gov**
  - **Extensive documentation (user, reference, developer)**
  - **Support mailing lists / archives**
  - **Software downloads: official releases and nightly stable (freely available worldwide via GNU GPL)**

- **User's Manual, Chapter 2: Tutorial with example input files**

- **Support:**
  - **dakota-users@software.sandia.gov**
    (DAKOTA team and internal/external user community)
  - **dakota-help@sandia.gov**
    (for SNL-specific or issues involving proprietary information)

Advanced Capabilities Topics

General features
- Restart
- Evaluation cache
- Utilities in dakota_restart_util
- Tabular graphics data
- Failure capturing: abort, retry, recover, ignore
- Constraint specification: linear, nonlinear; equality, inequality
- Input/output scaling
- Matlab interface

Approximation methods
- Global data fit surrogate methods (polynomials, MARS, Kriging, etc.)
- Local surrogate methods (Taylor series, multipoint)
- Hierarchical: high/low fidelity models
- Corrections

Strategies/Advanced approaches
- Nested models: OUU
- Multi-objective (Pareto) optimization
- Multistart; multi-level hybrid
- Surrogate-based optimization (variety of constraint handling approaches): trust region; EGO/EGRA
- Reliability-based design optimization
- Advanced UQ topics: polynomial chaos, second-order probability, Dempster-Shafer, surrogate-based UQ
- AMPL: for analytic problems / algebraic mappings

Parallel capabilities: message passing, asynchronous local, hybrid
- Asynchronous evaluations
- Dakota parallel, application serial
- Dakota serial, application parallel
- Multi-level parallel: concurrent iteration, concurrent function evaluations, concurrent analyses,
- multiprocessor simulations

Learning Goals: DAKOTA Overview

Were the DAKOTA Overview goals met?

- **Learn design and analysis processes DAKOTA supports; assess relevance in your domain**
- **Survey key algorithm classes available in DAKOTA; what differentiates it**
- **Requirements and process for getting started; do you see challenges?**
- **Tour the mechanics of running DAKOTA**
- **Know where to get help**

What questions remain?

DAKOTA 101: Coming Up

Method Tour

- **Sensitivity Analysis**
- **Uncertainty Quantification**
- **Optimization**
- **Calibration / Parameter Estimation**

Advanced Topics teasers