DAKOTA: Virtual Prototyping with Large-Scale Engineering Simulations

Michael S. Eldred
Optimization and Uncertainty Estimation Dept.
Sandia National Laboratories
Albuquerque, NM

http://endo.sandia.gov/DAKOTA
Outline

Intro/Overview
• Motivation
• OO software design

Algorithms R&D
• Surrogate-based optimization
• Mixed integer nonlinear programming with PICO
• Simultaneous analysis and design
• UQ with sampling/reliability/SFE methods
• Optimization under uncertainty (SBOUU, RBDO, SFE/SAND)

Framework Services
• Simulation interfacing
• Multilevel parallel computing

Case Studies
• Large-scale design of an electronics package
• OUU for ICF capsules
Answer fundamental engineering questions:
- What is the best design?
- How safe is it?
- How much confidence in my answer?

Motivations and Challenges:
- Reuse tools and interfaces
- Leverage optimization, UQ, et al.
- Nonsmooth/discontinuous/multimodal, mixed variables, unreliable gradients, simulation failures → state-of-the-art methods
- ASCI-scale applications and architectures → scalable parallelism
- Open-source framework for algorithm prototyping and dissemination
Overview of DAKOTA Framework

Strategy: control of multiple iterators and models

- Coordination:
  - Nested
  - Layered
  - Cascaded
  - Concurrent
  - Adaptive/Interactive

- Parallellism:
  - Asynchronous local
  - Message passing
  - Hybrid
  - 4 nested levels with Master-slave/dynamic Peer/static

Model:
- Parameters
- Design
  - continuous
discrete
- Uncertain
  - normal/logn
  - uniform/logu
  - weibull
  - histogram
- State
  - continuous
discrete

Application
- system call
- fork
direct
distr. resource

Approximation
- polynomial
- neural network
- splines
- kriging
- Taylor series
- hierarchical

Functions
- objectives
- constraints
- least sq. terms
- generic

Gradients
- numerical
- analytic
- mixed

Hessians
- analytic

Parameter
Model

Optimizer

Iterator

LeastSq

Vector

List

Dot
CONMIN
NPSOL
OPT++
SGOPT
FSPQ++

NonDeterm

AMV+
FORM

LHS/MC

DDACE

DoE

LeastSq

NonDeterm

AMV+
FORM

LHS/MC

DDACE

DoE

 Iterator

 Model

 Iterator

 Model

 LeastSq

 Vector

 List

 Dot
CONMIN
NPSOL
OPT++
SGOPT
FSPQ++

 Strategy

 Uncertainty

 Optimization

 Hybrid

 SurrBased

 Pareto/MStart

 Branch&Bound/PICO

 ModelFormExtrap

 DoE

 CCD/BB

 NonDeterm

 AMV+
 FORM

 LHS/MC

 DDACE

 Design
 continuous
discrete

 Uncertain
 normal/logn
 uniform/logu
 weibull
 histogram

 State
 continuous
discrete

 Application
 system call
 fork
direct
distr. resource

 Approximation
 polynomial
 neural network
 splines
 kriging
 Taylor series
 hierarchical

 Functions
 objectives
 constraints
 least sq. terms
 generic

 Gradients
 numerical
 analytic
 mixed

 Hessians
 analytic
Overview: Projects & Staff

**DAKOTA project (optimization with engineering simulations):**
- PI - Mike Eldred, 9211, mseldre@sandia.gov, (505) 844-6479
- Team - Tony Giunta, Mario Alleva, Steve Wojtkiewicz, Bart van Bloemen Waanders, Roscoe Bartlett, Bill Hart

**SGOPT project (stochastic global optimization):**
- PI - Bill Hart, 9211, wehart@cs.sandia.gov, (505) 844-2217

**PICO project (mixed integer programming, scheduling and logistics):**
- PI - Cindy Phillips, 9211, caphill@cs.sandia.gov, (505) 845-7296
- Team - Jonathan Eckstein (Rutgers), Bill Hart, Vitus Leung, Bob Carr

**rSQP++ project (large-scale SAND):**
- PI - Roscoe Bartlett, 9211, rabartl@sandia.gov
- Team - Bart van Bloemen Waanders, Omar Ghattas (CMU), Larry Biegler (CMU)

**OPT++/DDACE/APPS projects (NLP, sampling, & pattern search libraries):**
- PI - Steve Thomas, 8950, swhoma@ca.sandia.gov, (925) 294-2954
- Team - Paul Boggs, Patty Hough, Tammy Kolda, Leslea Lehoucq, Kevin Long, Pam Williams

**DAKOTA/UQ project (analytic reliability, sampling, and SFE UQ library):**
- PI - Steve Wojtkiewicz, 9124, sfwojtk@sandia.gov, (505) 284-5482
- Team - Mike Eldred, John Red-Horse, Laura Swiler, Angel Urbina
Overview: External Partnerships

**Investments in new collaborations are paying dividends back to Sandia**

- **LLNL**: A-Division is using DAKOTA with their *production physics codes* and has co-developed new database support capabilities. B and W divs. showing similar potential.

- **LANL/SwRI**: DAKOTA installed in ESA and CCS divisions; substantial benefit to Sandia (*several man years of effort*) with the delivery of a SwRI-developed GUI for our use.

- **LM-ATC**: funds-in through Shared Vision for DAKOTA training and development.

- **Johns Hopkins University**: faculty and students trained in DAKOTA and are using, testing, and *extending DAKOTA’s UQ techniques*.

- **Goodyear CRADA**: DAKOTA now performing studies in Akron & European Tech Center.

- **Commercialization partner**: Phoenix Integration is developing into a strategic partner for providing commercial support to our industrial user base. The integration of DAKOTA within Phoenix’s ModelCenter product is *funded and sponsored externally* by NASA Langley, LM-ATC, and the Navy IHAT program (RLV and hypersonic missile apps.).

- **Also**: CRMPC, ESI FIPER, Universities (CMU, Rice, U. Florida, VA Tech, Rutgers, Johns Hopkins, William & Mary)
Drivers (technical & political)

DOE DP labs environment: reduced reliance on testing
- shift towards increased reliance on high-end computer modeling
- uncertainties must be quantified for M&S-based certification

Complex simulations
- large/expensive simulations (up to 10M+ DOF)
  - exotic physics
  - component/sub-system/system performance
  - parameter estimation/material characterization
  - homeland security: inversion
  - UQ/OUU: V&V, certification, robust/reliable designs
- nonsmooth/discontinuous/multimodal
- mixed continuous/discrete variables
- gradients unavailable or unreliable
- simulation/hardware failures

Tactics
- address need for common tools within DOE DP labs
- leverage development wherever possible (OO design)
- solution “strategies” using algorithmic building blocks (SBO, OUU, MINLP)
Outline

Intro/Overview
• Motivation
• OO software design

Algorithms R&D
• Surrogate-based optimization
• Mixed integer nonlinear programming with PICO
• Simultaneous analysis and design
• UQ with sampling/reliability/SFE methods
• Optimization under uncertainty (SBOUU, RBDO, SFE/SAND)

Framework Services
• Simulation interfacing
• Multilevel parallel computing

Case Studies
• Large-scale design of an electronics package
• OUU for ICF capsules
Trust Region Surrogate-Based Optimization (SBO)

**Purpose:**

- Reduce the number of expensive, high-fidelity simulations by using a succession of approximate (surrogate) models
- Address nonsmoothness in the variations of response quantities by using continuously differentiable functions which smooth noisy surfaces with a best analytic fit

**Sequence of trust regions**

**Recent progress:**

- Provable convergence: 1st order consistency with beta correction
- Hierarchical case supported
  - Truth evaluation only at center
  - Beta correction is critical
- New SurfPack++ library initiated
  - Extensible library of global, local, multipoint surrogates
Mixed Integer Nonlinear Programming

**Purpose:**

- many engineering applications have variables which are discrete in nature, e.g. material selections, feature counts, stock sizes, gauge thicknesses
- if the number of discrete variables is more than a few, then simple enumeration of all combinations quickly becomes intractable

The Parallel Integer Combinatorial Optimization (PICO) package has been implemented as a multilevel parallel branch and bound strategy

$$\min f = \sum_{i=1}^{6} (x_i - 1.4)^4$$

subject to

$$g_1 = x_1 - \frac{x_2}{2} \leq 0$$

$$g_2 = x_2 - 0.5 \leq 0$$

$$-10.0 \leq x_1, x_2, x_3, x_4 \leq 10.0$$

$$x_5, x_6 \in \{0, 1, 2, 3, 4\}$$

5 to 7 NLP’s instead of 25

*Limitation:* restricted to noncategorical variables

![Diagram](image)
Simultaneous Analysis and Design

- Domains: few design vars, simulation intrusion impractical, SS/transient problems
- Simple and reliable, coarse- and fine-grained parallel

- Domains: many design variables, simulation intrusion practical (framework leveraging), steady-state nonlinear problems
- Fine-grained parallel, more special purpose but efficiency gains are very promising

Both are needed in an optimization framework
Uncertainty Quantification

- Many problems must be solved in the presence of uncertainty
  - inherent variability: aleatory/irreducible
  - lack of knowledge: epistemic/reducible
- New DAKOTA/UQ “iterator branch” leverages parallelism, simulation interfacing, surrogates, optimizers, etc.
  - Sampling: LHS, MC (McMC, QMC, Bootstrap, AIS planned)
  - Analytic reliability: MV/AMV/AMV+, FORM/SORM
  - SFE: Polynomial chaos expansions
- Driving connections to V&V programs at other labs and SwRI
- Collocation of opt and UQ in DAKOTA enables OUU
Optimization under Uncertainty (OUU)

• Optimization under uncertainty (OUU) methods encompass both:
  – Design for robustness (moment statistics: mean, variance)
  – Design for reliability (tail statistics: probability of failure)

• (Simulation-based) OUU methods best classified by UQ approach:
  – **Sampling-based** (noise-tolerant optimization)
    • SBOUU: trust region surrogate-based (Sandia) and stochastic approximation (Igusa); nongradient-based (Trosset)
  – **Analytic reliability-based** (exploit AMV/AMV+/FORM/SORM structure)
    • RBDO: reuse of reliability index ($\beta$) gradient, “cross-iterated” opt./UQ (Wu), combined opt./MPP search (Wang)
  – **Stochastic finite element-based** (large-scale, multiphysics)
    • SFE/SAND: polynomial chaos expansions (Ghanem) amenable to SAND

Also, evidence theory-based OUU (Renaud), …
Optimization under Uncertainty with Surrogates

Nested model: internal iterators/models execute a complete iterative study as part of every evaluation.

Layered model: internal iterators/models used for periodic update and verification of data fit (global/local/multipoint) or hierarchical (variable fidelity) surrogates.

Nestings/Layerings can recurse

Trust-region approaches maintain quality of results
Goal: provable convergence (for a selected confidence level)
Outline

Intro/Overview
• Motivation
• OO software design

Algorithms R&D
• Surrogate-based optimization
• Mixed integer nonlinear programming with PICO
• Simultaneous analysis and design
• UQ with sampling/reliability/SFE methods
• Optimization under uncertainty (SBOUU, RBDO, SFE/SAND)

Framework Services
• Simulation interfacing
• Multilevel parallel computing

Case Studies
• Large-scale design of an electronics package
• OUU for ICF capsules
Simulation Interfacing

- **Non-intrusive/black box**: system calls, forks, ... (sockets/XML)
  - greater than 20 codes interfaced to date
  - custom interfaces: development time as little as 15 mins., as much as several days
  - JAVA GUI (adapted from NESSUS) under dev. to automate process

- **Semi-intrusive**: direct linkage (simulation subroutine), POSIX threads

- **Fully intrusive**: SAND in SIERRA/NEVADA

Driver: shell script, C/C++/F77, Perl, Python, ...

Pre-.proc. • APREPRO
Simulation Code
Post-proc. • grep/cut
• BLOT
• ALGEBRA
• GROPE

Parameters File

Results File
Opportunities for exploiting parallelism

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)

**Math analysis & experiments**
- identify schemes which maximize parallel efficiency and are robust w.r.t. variability
- build schemes into automatic configuration utilities
Parallel DAKOTA options

A breadth of parallel models are flexibly supported, from the ASCI MP machines to networks of workstations to desktop multiprocessors

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 (ASCI Red/Blue/White, Cplant): Internal MPI partitions (nested parallelism)

```
yod -sz 601 dakota -i dakota.in:
```

```
MPI_COMM_WORLD
601 processors
```

```
COMM0 □ Master
COMM1 Slave analysis 100 procs.
COMM2 Slave analysis 100 procs.
COMM3 Slave analysis 100 procs.
COMM4 Slave analysis 100 procs.
COMM5 Slave analysis 100 procs.
COMM6 Slave analysis 100 procs.
```
Outline

Intro/Overview
• Motivation
• OO software design

Algorithms R&D
• Surrogate-based optimization
• Mixed integer nonlinear programming with PICO
• Simultaneous analysis and design
• UQ with sampling/reliability/SFE methods
• Optimization under uncertainty (SBOUU, RBDO, SFE/SAND)

Framework Services
• Simulation interfacing
• Multilevel parallel computing

Case Studies
• Large-scale design of an electronics package
• OUU for ICF capsules
ASCI Level 1 FY00 Milestone: EC design for hostile environments

**Purpose:**
- Electrical component (EC) incorporates new components supporting safety
- Refurbishment must meet strict weight & SM targets
- Design 500K DOF SALINAS model for transient impulse loads

**Impact:**
- 2-level parallelism on ASCI Red: utilized up to 2560 procs. (256 x 10) for ~5 days
- Comparable calculation on 1 processor: > 12 years
- Increased the minimum SM from -0.48 to +0.06 within weight bounds
- Generated design trade-off curve
ASCI Level 1 FY01 Milestone: UQ for microslip in joints (normal env.)

Forward propagation of parametric joint uncertainty

2-level parallel on ASCI Red (10 proc. SALINAS x 100 instances) used to assess variability in peak acceleration
Robust Hohlraum Design for Inertial Confinement Fusion

Wire initiation creates a “high Z” dense plasma
3D ALEGRA MHD

Encapsulant
Encapsulant converts the plasma radiation to a “drive” i.e., pressure on the capsule.
1D, 2D, 3D ALEGRA, rad-MHD

Drive and implosion of capsule.
1D, 2D ALEGRA rad-hydro

Sample Hohlraum Configuration

Uncertainties in: plasma, drive, and capsule characteristics
ICF Capsule Design

**Design goal:** maximize implosion velocity, but remain robust w.r.t. manufacturing variability

Minimize $V(r)$

Subject to $\sigma_v(r) \leq 1.6 \times 10^6$ cm/s

$0.105 \text{ cm} \leq r \leq 0.135 \text{ cm}$

uniform: $u = [-0.005, 0.005]$

---

TR-SBOUU finds solution vicinity in a single cycle, effectively stepping over the non-smoothness in $V(r)$ and $\sigma_v(r)$.
Conclusions/Current Directions

• GNU GPL open source release
  – Major UNIX (Sun, SGI, DEC, IBM), LINUX, & Windows (Cygwin). Mac OSX coming.
  – ~1000 registered download users since 12/01
  – streamlined interactions with universities, industry, & software vendors
  – return on investment is starting (distributed testing, contributed enhancements, contributed ports)

• OUU research (SBOUU, RBDO, SFE/SAND)

• SIERRA: intrusive coupling with DAKOTA

• SAND/SFE integration in DAKOTA/SIERRA
  (“S. DAKOTA” - SIERRA, “N. DAKOTA” - NEVADA)
Web Pages

DAKOTA:
A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis

Mike Eldred
Tony Giroto
Bart von Bloemen Waanders
Steve Wojtowicz
Bill Hart
Maha Alleva
Roman Bodennett
Sandia National Laboratories
P.O. Box 5800, Mail Stop 6487
Albuquerque, NM 87185-0847

Overview

The DAKOTA (Design Analysis Kit for Optimization and Terascale Applications) toolkit provides a flexible, extensible interface between analysis codes and optimization methods. DAKOTA contains algorithms for optimization with gradient and asymptotic-based methods, uncertainty estimation with sampling, reliability analysis, and stochastic finite element methods; parameter estimation with nonlinear least squares methods; and sensitivity analysis with design of experiments and parameter study capabilities. These capabilities may be used on their own or as components within automated systems. They are implemented as a public library to form a flexible and extensible problem-solving environment as well as a platform for research and rapid prototyping of advanced solution methods.

http://endo.sandia.gov/DAKOTA

DAKOTA Version 3.0 Public Release: 3/31/02

Release Notes

Downloading pre-built binaries provides the simplest route to accessing DAKOTA. For more experienced users, building DAKOTA from source allows customization with additional packages and porting to additional platforms or operating systems. It is recommended that people who have previously licensed DOT or NPSOL use the source distribution of DAKOTA and then follow the instructions in the INSTALL file to add DOT and/or NPSOL to their DAKOTA build. For SGI users, it is recommended to use a pre-compiled version of DAKOTA installed on one of the shared compute servers by the DAKOTA development team, rather than downloading DAKOTA from this site (since the former approach will give access to all supported packages).

To download, right-click on the filename, and choose “Save Link As”.

Binary distributions

- DAKOTA_3.0_devel3.2.tar.gz (Sun Solaris 2.8, 9 MB)
- DAKOTA_3.0_devel3.1.tar.gz (SGI IRIX 6.5, 16 MB)
- DAKOTA_3.0_devel2.1.tar.gz (DEC OSF/1 4.1, 10 MB)
- DAKOTA_3.0_devel3.3.tar.gz (IBM AIX 4.3, 9 MB)

These are binary distributions. You will need to uncompress and extract the tar file (using “gunzip” and “tar xz”), and then install the executables in the desired location (e.g., /usr/local/bin). See the INSTALL file in the distribution for more information.

Source distribution

- DAKOTA_3.0_devel.tar.gz (9 MB)

This is a source distribution. You will need to unpack and extract the tar file (using “gunzip” and “tar xz”). Then, to build the DAKOTA executable, the basic steps are to execute

```
configure
make
```

See the INSTALL file in the distribution for more information.