Dakota: Benefits and Challenges of Lab-developed Open Source Scientific Software

Brian M. Adams
Optimization and Uncertainty Quantification Dept.
http://dakota.sandia.gov

Scientific Software Days
February 25—26, 2016
Austin, TX
Talk Goals

- Give perspective on national lab context for Dakota development
- Share Dakota software and project goals
- Understand drivers for (open-source) software development
- Raise challenges and get community feedback

This talk is not unique to Dakota, nor the lab environment, but I hope to seed discussion.
SNL Mission: Advanced Science and Engineering for National Security

- Nuclear Weapons
- Defense Systems and Assessments
- Energy and Climate
- International, Homeland, and Nuclear Security

- Collegial environment with 12,000 (5,000 R&D; 6,000 advanced degreed) staff in Albuquerque, NM and Livermore, CA

- Dakota Mission: To serve Sandia’s mission through state-of-the-art research and robust, usable software for optimization and uncertainty quantification.
SNL’s Research Framework

Strong research foundations play a differentiating role in our mission delivery

Computing & Information Sciences

Center for Computing Research
Optimization and UQ
Dakota

Materials Sciences

Radiation Effects & High Energy Density Science

Engineering Sciences

Nanodevices & Microsystems

Bioscience

Geoscience
Dakota: Algorithms for Design Exploration and Simulation Credibility

- Suite of iterative mathematical and statistical methods that interface to computational models
- Makes sophisticated parametric exploration of black-box simulations practical for a computational design-analyze-test cycle:
  - Sensitivity Analysis
  - Uncertainty Quantification
  - Design Optimization
  - Model Calibration

**Goal:** provide scientists and engineers (analysts, designers, decision makers) richer perspective on model predictions
Diverse Simulations Across Scales

- Emergencies: weather, logistics, economics, human behavior
- Electrical circuits: networks, PDEs, differential algebraic equations (DAEs), E&M
- Shock loading of polymer foam: molecular dynamics
- Micro-electro-mechanical systems (MEMS): quasi-static nonlinear elasticity, process modeling
- Joint mechanics: system-level FEA for component assessment
- Systems of systems analysis: multi-scale, multi-phenomenon
Relations with Other Scientific Software

Dakota is comprised of

- Dakota and other Sandia-developed optimization, design of experiments, UQ, and surrogate model packages (only some actively developed)
- Partially DOE funded third-party libraries, e.g., FSUDace, PSUADE, QUESO
- Historical (legacy) third-party libraries (technical debt, usability challenge)
- Trilinos for numerics foundations

And interfaces with

- Simulation Codes (scalability is a challenge here!):
  - Sandia-developed: both loose and tight integration
  - Other open source
  - Commercial

- Visualization and post-processing tools: both for simulation output and Dakota results

- Simulation analysis environments / GUIs
Engineering Needs Drive Dakota R&D

Develop/deploy advanced approaches to help solve practical problems:

- Characterize parameter uncertainty $\rightarrow$ Bayesian calibration
- Hybrid analysis $\rightarrow$ mix methods, surrogates, and models
- Mixed uncertainty characterizations $\rightarrow$ epistemic and mixed UQ approaches
- Costly simulations $\rightarrow$ surrogate-based optimization and UQ
- Build in safety or robustness $\rightarrow$ mixed deterministic/probabilistic methods

$$\begin{align*}
\text{min} & \quad f(d) + W s_{u}(d) \\
\text{s.t.} & \quad g_l \leq g(d) \leq g_u \\
& \quad h(d) = h_t \\
& \quad d_l \leq d \leq d_u \\
& \quad a_l \leq A_i s_{u}(d) \leq a_u \\
& \quad A e s_{u}(d) = a_t
\end{align*}$$

![Diagram of optimization problem](image)
**SNL Environment: Benefits/Challenges**

- Rich, though challenging, problems across science/engineering domains
  
  “I want to do UQ with 200 parameters, but can only run two simulations.”

- Healthy culture of intra- and inter-institution collaboration

- Strong Dakota name recognition and track record; hundreds of SNL users, more DOE-wide; many support requests

- Must regularly deliver and support application-ready, usable software

- Rewarded by customers/users for both time-tested and leading-edge algorithms in software as well as close consulting partnerships

- CIS research foundation and CCR expect and reward research, software, and publications, though we aren’t in the commercial software business
Life of Dakota

1994
- Unify engineering optimization
  $: LDRD
- Surrogate-based methods
  Proprietary

2001
- Risk-informed decision making
- $: NW/ASC V&V UQ, OUU, multi-fidelity methods
- Open source v3.0 (GPL)

2006
- Significant UQ investments
  $: Energy, Climate
- Public mailing lists
  2009: v5.0 (LGPL)

2011
- Scalable algorithms
- Production support
  $: Office of Science, DARPA
- Agile development, v5.2

Mike Eldred
Founder

Invested developer, solving a practical problem
Why Open Source?

- Lab default is (typically) government use, then commercial license
- Open source (2001) for easier collaboration on algorithm development, primarily with faculty and students (before, during, after internships)
  - Vanderbilt: reliability methods for UQ
  - MIT: surrogate-based and multi-fidelity optimization
  - Stanford: UQ and active subspace methods; PSAAP applications
  - UT Austin: Bayesian inference

- Also attract integrators across sectors, e.g.,
  - NREL/NASA integration into OpenMDAO
  - Lockheed Martin integration with ModelCenter
  - Use with OpenFOAM; integration with CAESES commercial CFD

- Better scale with user base: create an engaged user community
- Over 20,000 package downloads since 2010 across all sectors (impact?)
- Whether we are genuinely, all-in open source hasn’t been tested...
Dakota Project as a Competency

- Dakota is more than software: an enthusiastic team (of fractional persons) with balanced strengths in algorithm research, software design and development, and application deployment and support
  - Mathematicians, statisticians, computer scientists, computational engineers
  - Expertise in sensitivity analysis, optimization, calibration, UQ, surrogate modelling
  - Software engineering with C++, Python, Java
  - User support to deep consulting

- But core team entirely at SNL!

- How do we grow the team / contributors as the user community grows?
- Or more critically, how do we realize benefits of open source to help scale?
Funding Picture

- Dakota strives to maintain a balanced funding portfolio; across
  - Research to production spectrum
  - Sponsor type and sizes: both core stewardship and smaller exploratory
  - Application domains
- Dakota often central to proposals, e.g., CASL, DARPA
- Example balanced portfolio:
  LDRD, ASCR, SciDAC, DARPA, CASL, NW/ASC Software, NEAMS, Industry

- Discussion points (*how do audience members manage?):*
  - Individually funded PIs may bring their capability to Dakota
  - How to steward Dakota capability base and manage technical debt; may not be valued by some sponsors
  - How to pool / manage small funded requests, whether development or training/support?
  - What drives much needed usability efforts?
Dakota Community

- Extensive website: documentation, training materials, downloads
- Active public mailing list, *though not browsable*; moving to online forums
- Publicly readable Subversion repository

- High usage in and outside labs
- Solicited for both research and commercial engagements, mostly small scale
- Receive a few patches and bug reports monthly (many languish; perhaps due to misalignment)
- Team cannot respond to all user (or developer) requests nor reach all analysis domains
- Some users help each other, including a few superstars

http://dakota.sandia.gov
Toward a Self-Sustaining Community

- We would like to build a more engaged community that
  - Helps itself (basic usage, advanced support)
  - Improves portability and interfaces by deploying to new platforms and application codes
  - Contributes to software development

- What should our team put priority on to attract and build trust with a user/developer community?
  - Incentivize use case contributions?
  - Explicitly prioritize engagements with certain super-users?
  - Better web resources (can be challenging in the lab environment)?
    - Clear public interfaces for bugs, patches, discussion?
  - User / developer group meetings?
  - External partnerships for deployment and user support?
Technical Growth to Promote Engagement

Potential development priorities to increase contribution

- **Improved modularity** so users can extend, contribute, components, e.g.,
  - Surrogate model module with Python bindings
  - More usable simulation interfacing that encourages best practices

- **Community repository** of contributed code, examples, scripts

- **Clear development practices**, e.g., principles, code standards, easier build/test on new platforms

- **Remain on cutting edge of algorithms** to encourage it as a research vehicle. Representative current directions:
  - Bayesian calibration and model discrepancy
  - Multi-fidelity UQ and inference
  - Portability to extreme scale computers, growth into hybrid parallel
  - SA and UQ scalability with active subspaces; generalize to random fields
  - Expanding mixed-integer optimization
To Seed Discussion

- What approaches and resources have you found most helpful in creating a vibrant user community?
- What investments or behaviors have yielded the most effective developer contributions?
- How do you set, communicate, and manage expectations and priorities?

- Regrets I can’t stay long today...
  briadam@sandia.gov
  http://dakota.sandia.gov

Thanks for your attention!