Rapid Optimization Library: A Preview

Denis Ridzal and Drew Kouri, Org. 1441

Sandia National Laboratories

Special thanks to Joseph Young.

SAND2013-9828P

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.
Research sponsored by the NNSA Advanced Scientific Computing Program under CSAR.

Purpose

— Optimization of differentiable simulated processes:
  - partial differential equations (PDEs)
  - differential algebraic equations (DAEs)
  - network equations (gas networks, electrical networks)

— Inverse problems, optimal design and control problems.

— The parameter/design/control spaces can be very large, often related to the size of the computational mesh (PDEs) or the size of the device network or graph (DAEs).

⇒ *Matrix-free, gradient-based, embedded methods.*

History

- **ROL** started as a special-purpose effort in 2009: Delivering unconstrained optimization algorithms for PDE-constrained optimization to a few customers.
- Parallel development: **PEOpt** by Joe Young.
  - An experimental platform for a variety of optimization algorithms and utilities: unconstrained, equality-constrained, inequality-constrained optimization; checkpointing and restarting; consistency checks for derivatives, symmetry, etc.
  - Open source, not in Trilinos, available upon request.
- **ROL** is a Trilinos package for gradient-based matrix-free optimization with focus on uniform yet flexible interfaces, robustness and close-to-optimal efficiency.

Coming soon

- **ROL** is currently in preCopyrightTrilinos.
- Will supersede most of Aristos, MOOCHO and Optipack.
- Will include all of **PEOpt** functionality.
- **Additionally focused on:** inexact objective functions, inexact gradients, adaptive models, optimization under uncertainty.

Infrastructure

- User specifies linear algebra through a vector class:
  `plus`, `scale`, `dot`, `norm`, `clone`, `axpy`, `zero`, `basis`.

- `Algorithm`: Combines `Step` and `StatusTest`.

- Globalization performed at the `Step` level:
  `LineSearchStep`, `TrustRegionStep`.

- Efficient computations, restarts and checkpointing enabled through
  `AlgorithmState` and `StepState`.

- Minimal functional interface. For instance, `Objective` class:
  `value`, `gradient`, `hessVec`, `invHessVec`, `dirDeriv`.

Basic methods

- **Unconstrained optimization:**
  Gradient descent, quasi-Newton (secant) methods, nonlinear CG, Gauss-Newton, Newton, with line-search and trust-region globalizations.

- **Equality constraints:**
  Sequential quadratic programming (SQP), with line-search and trust-region globalizations.

- **Inequality constraints:**
  For box constraints, use projected gradient and projected Newton methods. For general inequalities, use interior-point algorithms.

Advanced methods

- All trust-region based algorithms will take advantage of inexact objective functions and inexact gradients.
- The inexactness framework enabless (computationally feasible) large-scale *optimization under uncertainty*: Stochastic programming meets matrix-free gradient-based optimization.
- Later in 2014, **subpackage for optimization under uncertainty**.