Open-Source Software for Interfacing and Support of Large-scale Embedded Nonlinear Optimization

Roscoe A. Bartlett


Department of Optimization & Uncertainty Estimation

Sandia National Laboratories

INFORMS Annual Meeting, October 14, 2008
Computational Sciences at Sandia: PDEs and More...

- Chemically reacting flows
- Climate modeling
- Combustion
- Compressible flows
- Computational biology
- Circuit modeling
- Inhomogeneous fluids
- Materials modeling
- MEMS modeling
- Seismic imaging
- Shock and multiphysics
- Structural dynamics
- Heat transfer
- Network modeling

Large-Scale Parallel Distributed Memory and Implicit Iterative Solvers are Major Themes!
Overview of Trilinos

• Provides a suite of numerical solvers to support predictive simulation for Sandia’s customers
  => Scope has expended to include discretizations methods, …
• Provides a decoupled and scalable development environment to allow for algorithmic research and production capabilities => “Packages”
• Mostly C++ with some C, Fortran, Python …
• Advanced object-oriented and generic C++ …
• Freely available under and open-source LGPL license …

Current Status
• Current Release Trilinos 9.0 (September 2008)
• Next Release Trilinos 10.0? (March 2009?)

Trilinos website
  http://trilinos.sandia.gov
Introducing Abstract Numerical Algorithms

What is an abstract numerical algorithm (ANA)?
An ANA is a numerical algorithm that can be expressed abstractly solely in terms of vectors, vector spaces, linear operators, and other abstractions built on top of these without general direct data access or any general assumptions about data locality.

**Example: Linear Conjugate Gradients**
Given:
\[ A \in \mathcal{X} \rightarrow \mathcal{X} : \text{s.p.d. linear operator} \]
\[ b \in \mathcal{X} : \text{right hand side vector} \]

Find vector \( x \in \mathcal{X} \) that solves \( Ax = b \)

**Linear Conjugate Gradient Algorithm**

Compute \( r^{(0)} = b - Ax^{(0)} \) for the initial guess \( x^{(0)} \).

\[
\begin{align*}
\text{for } i = 1, 2, \ldots & \\
\rho_{i-1} &= \langle r^{(i-1)}, r^{(i-1)} \rangle \\
\beta_{i-1} &= \rho_{i-1}/\rho_{i-2} \quad (\beta_0 = 0) \\
p^{(i)} &= r^{(i-1)} + \beta_{i-1}p^{(i-1)} \\
q^{(i)} &= Ap^{(i)} \\
\gamma_i &= \langle p^{(i)}, q^{(i)} \rangle \\
\alpha_i &= \rho_{i-1}/\gamma_i \\
x^{(i)} &= x^{(i-1)} + \alpha_ip^{(i)} \\
r^{(i)} &= r^{(i-1)} - \alpha_iq^{(i)}
\end{align*}
\]

check convergence; continue if necessary

**Key Points**
- ANAs can be very mathematically sophisticated!
- ANAs can be extremely reusable!
- Flexibility needed to achieve high performance!

**Types of operations**
- Linear operator applications
- Vector-vector operations
- Scalar operations
- Scalar product \( <x,y> \) defined by vector space

**Types of objects**
- Linear Operators
  - \( A \)
- Vectors
  - \( r, x, p, q \)
- Scalars
  - \( \rho, \beta, \gamma, \alpha \)
- Vector spaces?
  - \( \mathcal{X} \)