MS17

A Case Study on the Vertical Integration of Trilinos Solver Algorithms with a Production Application Code

Organizer: Roscoe A. Bartlett

_Sandia National Laboratories_

10:00-10:25 Overview of the Vertical Integration of Trilinos Solver Algorithms in a Production Application Code

_Roscoe A. Bartlett_, Sandia National Laboratories

10:30-10:55 Analytic Sensitivities in Large-scale Production Applications via Automatic Differentiation with Sacado

_Eric Phipps_, Sandia National Laboratories

11:00-11:25 To PDE Components and Beyond

_Andy Salinger_, Sandia National Laboratories

11:30-11:55 Analysis Tools for Large-scale Simulation with Application to the Stationary Magnetohydrodynamics Equations

_Roger Pawlowski_, Eric Phipps, Heidi K. Thornquist, and Roscoe A. Bartlett, Sandia National Laboratories
Overview of the Vertical Integration of Trilinos Solver Algorithms in a Production Application Code

Roscoe A. Bartlett
Department of Optimization & Uncertainty Estimation

http://www.cs.sandia.gov/~rabartl

Sandia National Laboratories

March 13th, 2008
Overview of Trilinos Vertical Integration Project (Milestone)

• **Goal:** Vertically integrate Trilinos solver algorithms in Trilinos to build new predictive embedded analysis capabilities
  • **Impact:** Vertically integrated 10+ Trilinos algorithm packages
• **Goal:** Demonstrate on relevant production applications
  • **Impact:** Solved steady-state parameter estimation problems and transient sensitivities on semiconductor devices in Charon
  • **Impact:** Solved Eigen problems on MHD problem in Charon
• **Added Goal:** Explore refined models of collaboration between production application developers and algorithm researchers.
  • **Impact:** Closer collaboration between application and algorithm developers yielding better algo and app R&D

Outline

- Overview of Trilinos and Charon
- Overview of vertical solver algorithm integration
- Moving beyond the forward solve
  - Challenges/barriers to embedded analysis methods
  - Enabling methods
- Examples of vertically integrated algorithms with Trilinos and Charon
- Steady-state parameter estimation optimization with MOOCHO/Charon
Outline

- Overview of Trilinos and Charon

- Overview of vertical solver algorithm integration

- Moving beyond the forward solve
  - Challenges/barriers to embedded analysis methods
  - Enabling methods

- Examples of vertically integrated algorithms with Trilinos and Charon

- Steady-state parameter estimation optimization with MOOCHO/Charon
Overview of Trilinos

- Provides a suite of numerical solvers to support predictive simulation for Sandia’s customers
  
  => Scope has expanded to include discretizations methods, …!
- Provides a decoupled and scalable development environment to allow for algorithmic research and production capabilities => “Packages”
- Provides support for growing SQA requirements
- Mostly C++ with some C, Fortran, Python …
- Advanced object-oriented and generic C++ …

**Current Status**

- Current release: Trilinos 8.0.x (September 2007)
- Next release Trilinos 9.0 (September 2008)

**Trilinos website**

http://trilinos.sandia.gov
# Trilinos (8.0 & 9.0+) Package Summary

<table>
<thead>
<tr>
<th>Objective</th>
<th>Package(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Discretizations</strong></td>
<td><strong>Meshing &amp; Spatial Discretizations</strong></td>
</tr>
<tr>
<td><strong>Methods</strong></td>
<td><strong>Automatic Differentiation and UQ Prop.</strong></td>
</tr>
<tr>
<td><strong>Core</strong></td>
<td><strong>Mortar Methods</strong></td>
</tr>
<tr>
<td><strong>Solvers</strong></td>
<td><strong>Linear algebra objects</strong></td>
</tr>
<tr>
<td><strong>Load Balancing</strong></td>
<td><strong>Abstract interfaces</strong></td>
</tr>
<tr>
<td><strong>“Skins”</strong></td>
<td><strong>Epetra, Jpetra, Tpetra</strong></td>
</tr>
<tr>
<td><strong>C++ utilities, (some) I/O</strong></td>
<td><strong>Thyra, Stratimikos, RTOp</strong></td>
</tr>
<tr>
<td><strong>Solvers</strong></td>
<td><strong>Iterative (Krylov) linear solvers</strong></td>
</tr>
<tr>
<td><strong>Direct sparse linear solvers</strong></td>
<td><strong>Amesos</strong></td>
</tr>
<tr>
<td><strong>Direct dense linear solvers</strong></td>
<td><strong>Epetra, Teuchos, Pliris</strong></td>
</tr>
<tr>
<td><strong>Iterative eigenvalue solvers</strong></td>
<td><strong>Anasazi</strong></td>
</tr>
<tr>
<td><strong>ILU-type preconditioners</strong></td>
<td><strong>AztecOO, IFPACK</strong></td>
</tr>
<tr>
<td><strong>Multilevel preconditioners</strong></td>
<td><strong>ML, CLAPS</strong></td>
</tr>
<tr>
<td><strong>Block preconditioners</strong></td>
<td><strong>Meros</strong></td>
</tr>
<tr>
<td><strong>Nonlinear system solvers</strong></td>
<td><strong>NOX, LOCA</strong></td>
</tr>
<tr>
<td><strong>Time Integration &amp; Sensitivities</strong></td>
<td><strong>Rythmos</strong></td>
</tr>
<tr>
<td><strong>Analysis</strong></td>
<td><strong>Optimization (SAND)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>MOOCHO, Aristos</strong></td>
</tr>
</tbody>
</table>

**Green:** Packages used in Vertical Integration Milestone  
**Gray:** New packages that will be included in Trilinos 9.0 (September 2008) or later
Trilinos Strategic Goals

- **Scalable Computations**: As problem size and processor counts increase, the cost of the computation will remain nearly fixed.
- **Hardened Computations**: Never fail unless problem essentially intractable, in which case we diagnose and inform the user why the problem fails and provide a reliable measure of error.
- **Full Vertical Coverage**: Provide leading edge enabling technologies through the entire technical application software stack: from problem construction, solution, analysis and optimization.

- **Grand Universal Interoperability**: All Trilinos packages will be interoperable, so that any combination of solver packages that makes sense algorithmically will be possible within Trilinos.
- **Universal Accessibility**: All Trilinos capabilities will be available to users of major computing environments: C++, Fortran, Python and the Web, and from the desktop to the latest scalable systems.
- **Universal Solver RAS**: Trilinos will be:
  - **Reliable**: Leading edge hardened, scalable solutions for each of these applications
  - **Available**: Integrated into every major application at Sandia
  - **Serviceable**: Easy to maintain and upgrade within the application environment.

**Courtesy of Mike Heroux, Trilinos Project Leader**
• Internal SNL Code for QASPR project
• Large-scale parallel (MPI)
• Unstructured grid finite elements
• Automatic Differentiation
• Adaptive Mesh Refinement
• Generalized operators – fast addition of new operators/equations
• Physics
  – Semiconductor Device
  – Multi-phase Aerosol
  – Reacting flows/gas-phase Combustion
  – MHD/Plasma
• Algorithms testing ground
Outline

- Overview of Trilinos and Charon

- Overview of vertical solver algorithm integration

- Moving beyond the forward solve
  - Challenges/barriers to embedded analysis methods
  - Enabling methods

- Examples of vertically integrated algorithms with Trilinos and Charon

- Steady-state parameter estimation optimization with MOOCHO/Charon
Trilinos Strategic Goals

• **Scalable Computations**: As problem size and processor counts increase, the cost of the computation will remain nearly fixed.

• **Hardened Computations**: Never fail unless problem essentially intractable, in which case we diagnose and inform the user why the problem fails and provide a reliable measure of error.

• **Full Vertical Coverage**: Provide leading edge enabling technologies through the entire technical application software stack: from problem construction, solution, analysis and optimization.

• **Grand Universal Interoperability**: All Trilinos packages will be interoperable, so that any combination of solver packages that makes sense algorithmically will be possible within Trilinos.

• **Universal Accessibility**: All Trilinos capabilities will be available to users of major computing environments: C++, Fortran, Python and the Web, and from the desktop to the latest scalable systems.

• **Universal Solver RAS**: Trilinos will be:
  - **Reliable**: Leading edge hardened, scalable solutions for each of these applications
  - **Available**: Integrated into every major application at Sandia
  - **Serviceable**: Easy to maintain and upgrade within the application environment.

**Courtesy of Mike Heroux, Trilinos Project Leader**
Linear Problems: Given linear operator (matrix) \( A \in \mathbb{R}^{n \times n} \)
- Linear equations: Solve \( Ax = b \) for \( x \in \mathbb{R}^n \)
- Eigen problems: Solve \( Av = \lambda v \) for (all) \( v \in \mathbb{R}^n \) and \( \lambda \in \mathbb{R} \)

Nonlinear Problems: Given nonlinear operator \( f(x, p) \in \mathbb{R}^{n+m} \rightarrow \mathbb{R}^n \)
- Nonlinear equations: Solve \( f(x) = 0 \) for \( x \in \mathbb{R}^n \)
- Stability analysis: For \( f(x, p) = 0 \) find space \( p \in P \) such that \( \frac{\partial f}{\partial x} \) is singular

Transient Nonlinear Problems:
- DAEs/ODEs: Solve \( f(\dot{x}(t), x(t), t) = 0, t \in [0, T], x(0) = x_0, \dot{x}(0) = x_0' \)
  for \( x(t) \in \mathbb{R}^n, t \in [0, T] \)

Optimization Problems:
- Unconstrained: Find \( p \in \mathbb{R}^m \) that minimizes \( g(p) \)
- Constrained: Find \( x \in \mathbb{R}^n \) and \( p \in \mathbb{R}^m \) that:
  minimizes \( g(x, p) \)
  such that \( f(x, p) = 0 \)
Vertical Integration and Interoperability is Important

Example: Numerous interactions exist between layers of abstract numerical algorithms (ANAs) in a transient optimization problem.

What is needed to solve problem?
• Standard interfaces to break $O(N^2)$ 1-to-1 couplings

Thyra is being developed to address interoperability of ANAs by defining interfaces for:
- Linear operators/vectors
- Preconditioners / Linear solvers
- Nonlinear models
- Nonlinear solvers
- Transient solvers

Key Points
• Higher level algorithms, like optimization, require a lot of interoperability
• Interoperability and vertical integration must be “easy” or these configurations will not be achieved in practice
Outline

- Overview of Trilinos and Charon

- Overview of vertical solver algorithm integration

- Moving beyond the forward solve
  - Challenges/barriers to embedded analysis methods
  - Enabling methods

- Examples of vertically integrated algorithms with Trilinos and Charon

- Steady-state parameter estimation optimization with MOOCHO/Charon
Embedded Analysis Algorithms and “The Cutting Edge”

- **Forward Simulator** (linear solvers, preconditioners, …)
- **Sensitivities and Error-Estimation**
- **Optimization (Opt)**
- **UQ**
- **Opt and UQ**

**Complexity/Fidelity of Simulation (% of “cutting edge”)**

- **The “Cutting Edge” for the Forward Simulation Application**
  - Drives capability computing (e.g., Gordan Bell, etc.)
  - Drives (i.e., “Pulls”) R&D for linear solvers, preconditioners, …

- **Advanced Analysis Methods**
  - Lag behind the “cutting edge” of the forward simulation
  - R&D reduces the lag!
  - Less direct impact on the forward simulation results => Leads to “Push” instead of “Pull”
  - Requires a different approach w.r.t. working with APP developers and customers!
We are now addressing these barriers in a fundamental way to provide the foundation for sustained embedded algorithms R&D.
**Key Points**

- Provide single interface from nonlinear ANAs to applications
- Provides for shared, uniform access to linear solver capabilities
- Once an application implements support for one ANA, support for other ANAs can be added incrementally
<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nonlinear equations:</td>
<td>Solve $f(x) = 0$ for $x \in \mathbb{R}^n$</td>
</tr>
<tr>
<td>Stability analysis:</td>
<td>For $f(x, p) = 0$ find space $p \in P$ such that $\frac{\partial f}{\partial x}$ is singular</td>
</tr>
<tr>
<td>Explicit ODEs:</td>
<td>Solve $\dot{x} = f(x, t) = 0$, $t \in [0, T]$, $x(0) = x_0$, for $x(t) \in \mathbb{R}^n$, $t \in [0, T]$</td>
</tr>
<tr>
<td>DAEs/Implicit ODEs:</td>
<td>Solve $f(\dot{x}(t), x(t), t) = 0$, $t \in [0, T]$, $x(0) = x_0$, $\dot{x}(0) = x'_0$, for $x(t) \in \mathbb{R}^n$, $t \in [0, T]$</td>
</tr>
<tr>
<td>Explicit ODE Forward Sensitivities:</td>
<td>Find $\frac{\partial x}{\partial p}(t)$ such that: $\dot{x} = f(x, p, t) = 0$, $t \in [0, T]$, $x(0) = x_0$, for $x(t) \in \mathbb{R}^n$, $t \in [0, T]$</td>
</tr>
<tr>
<td>DAE/Implicit ODE Forward Sensitivities:</td>
<td>Find $\frac{\partial x}{\partial p}(t)$ such that: $f(\dot{x}(t), x(t), p, t) = 0$, $t \in [0, T]$, $x(0) = x_0$, $\dot{x}(0) = x'_0$, for $x(t) \in \mathbb{R}^n$, $t \in [0, T]$</td>
</tr>
<tr>
<td>Unconstrained Optimization:</td>
<td>Find $p \in \mathbb{R}^m$ that minimizes $g(p)$</td>
</tr>
<tr>
<td>Constrained Optimization:</td>
<td>Find $x \in \mathbb{R}^n$ and $p \in \mathbb{R}^m$ that:</td>
</tr>
<tr>
<td></td>
<td>minimizes $g(x, p)$</td>
</tr>
<tr>
<td></td>
<td>such that $f(x, p) = 0$</td>
</tr>
<tr>
<td>ODE Constrained Optimization:</td>
<td>Find $x(t) \in \mathbb{R}^n$ in $t \in [0, T]$ and $p \in \mathbb{R}^m$ that:</td>
</tr>
<tr>
<td></td>
<td>minimizes $\int_0^T g(x(t), p)$</td>
</tr>
<tr>
<td></td>
<td>such that $\dot{x} = f(x(t), p, t) = 0$, on $t \in [0, T]$</td>
</tr>
<tr>
<td></td>
<td>where $x(0) = x_0$</td>
</tr>
</tbody>
</table>
• The Idea:
  – Keep the development versions of APP and Trilinos code updated and tested daily
  – Automated daily integrations tests

  => Results in better production capabilities and better research

• Charon + Trilinos Dev
  – Development versions of Charon and Trilinos are kept up-to-date every day!
  – New embedded optimization and sensitivity capabilities are run and tested every day!

• Aria/SIERRA + Trilinos Dev
  – We have automated configuration and daily integration testing of Aria/SIERRA VOTD against Trilinos Dev working!
  – Now, we are addressing Aria/SIERRA software infrastructure issues and will start adding new embedded Trilinos analysis algorithms!

Outline

- Overview of Trilinos and Charon

- Overview of vertical solver algorithm integration

- Moving beyond the forward solve
  - Challenges/barriers to embedded analysis methods
  - Enabling methods

- Examples of vertically integrated algorithms with Trilinos and Charon

- Steady-state parameter estimation optimization with MOOCHO/Charon
Vertical Integrations of Trilinos Capabilities: Example 1

See Andy Salinger’s Talk at 11:00 AM

Trilinos Capabilities

- Analysis Tools *(embedded)*
  - Nonlinear Solver
  - Time Integration
  - Continuation
  - Sensitivity Analysis
  - Stability Analysis
  - Constrained Solves
  - Optimization

- Linear Algebra
  - Data Structures
  - Iterative Solvers
  - Direct Solvers
  - Eigen Solver
  - Preconditioners
  - Matrix Partitioning

- Derivative Tools
  - Derivatives
  - Sensitivities

Parameter 1 (0) absolute sensitivities
Integrated vs. finite diff (1e-2)

Transient sensitivity analysis of a 2n2222 BJT in Charon with AD+Rythmos: 14x faster than FD

See Eric Phipp’s Talk at 10:30 AM
Vertical Integrations of Trilinos Capabilities: Example 2

Trilinos Capabilities

- **Analysis Tools (embedded)**
  - Nonlinear Solver
  - Time Integration
  - Continuation
  - Sensitivity Analysis
  - Stability Analysis
  - Constrained Solves
  - Optimization

- **Linear Algebra**
  - Data Structures
  - Iterative Solvers
  - Direct Solvers
  - Eigen Solver
  - Preconditioners
  - Matrix Partitioning

- **Derivative Tools**
  - Derivatives
  - Sensitivities

Destabilizing eigen-vector for heated fluid in magnetic field

See Roger Pawlowski’s Talk at 11:30 AM
Vertical Integrations of Trilinos Capabilities: Example 3

Trilinos Capabilities

- Analysis Tools (embedded)
  - Nonlinear Solver
  - Time Integration
  - Continuation
  - Sensitivity Analysis
  - Stability Analysis
  - Constrained Solves
  - Optimization

- Linear Algebra
  - Data Structures
  - Iterative Solvers
  - Direct Solvers
  - Eigen Solver
  - Preconditioners
  - Matrix Partitioning

- Derivative Tools
  - Derivatives
  - Sensitivities

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \| g(x, p) - g^* \|_2^2 + \frac{1}{2} \beta \| p \|_2^2 \\
\text{subject to} & \quad f(x, p) = 0
\end{align*}
\]

Steady-State Parameter Estimation Problem using 2n2222 BJT in Charon MOOCHO + AD

I am talking about this next
Outline

- Overview of Trilinos and Charon
- Overview of vertical solver algorithm integration
- Moving beyond the forward solve
  - Challenges/barriers to embedded analysis methods
  - Enabling methods
- Examples of vertically integrated algorithms with Trilinos and Charon
- Steady-state parameter estimation optimization with MOOCHO/Charon
QASPR
Qualification of electronic devices in hostile environments

Stockpile BJT

PDE semiconductor device simulation

Defect reactions
Si interstitial (I) (+2,+1,0,–1,–2)
Vacancy (V) (+2,+1,0,–1,–2)
VV (+1,0,–1,–2)
BI (+,0,–)
CI (+,0,–)
VP (0,–)
VB (+,0)
VO (0,–)
BIB (0,–)
BIO (+,0)

Annihilation

Graph showing base current vs. time with annotations:
No irradiation: $I_B = 0.05 \mu A$
Experiment
Defect annealing

Graph with data points and trend lines.
Minimize Current model vs. target mismatch
Subject to: Steady-state semiconductor defect physic FE model

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2}\|g(x, p) - g^*\|_2^2 + \frac{1}{2}\beta\|p\|_2^2 \\
\text{subject to} & \quad f(x, p) = 0
\end{align*}
\]

- Solved current matching optimization problems to calibrate model parameters against target currents
- MOOCHO (Bartlett) optimization solver converges simulation model and optimality at same time
  - Faster and more robust than black-box optimization methods
  - More accurate solutions

- **Successes**
  - Very accurate inversion of currents and model parameters for contrived "inverse" problems

- **Challenges**
  - Extremely difficult nonlinear solver convergences on model convergence
    => Opportunities for algorithm research
  - Inability to match experimental data
    => May indicate incomplete FE model
Summary of Trilinos Vertical Integration Project (Milestone)

• **Goal:** Vertically integrate Trilinos solver algorithms in Trilinos to build new predictive embedded analysis capabilities
  • **Impact:** Vertically integrated 10+ Trilinos algorithm packages
• **Goal:** Demonstrate on relevant production applications
  • **Impact:** Solved steady-state parameter estimation problems and transient sensitivities on semiconductor devices in Charon
  • **Impact:** Solved Eigen problems on MHD problem in Charon
• **Added Goal:** Explore refined models of collaboration between production application developers and algorithm researchers.
  • **Impact:** Closer collaboration between application and algorithm developers yielding better algo and app R&D

MS17

A Case Study on the Vertical Integration of Trilinos Solver Algorithms with a Production Application Code

Organizer: Roscoe A. Bartlett

Sandia National Laboratories

10:00-10:25 Overview of the Vertical Integration of Trilinos Solver Algorithms in a Production Application Code

Roscoe A. Bartlett, Sandia National Laboratories

10:30-10:55 Analytic Sensitivities in Large-scale Production Applications via Automatic Differentiation with Sacado

Eric Phipps, Sandia National Laboratories

11:00-11:25 To PDE Components and Beyond

Andy Salinger, Sandia National Laboratories

11:30-11:55 Analysis Tools for Large-scale Simulation with Application to the Stationary Magnetohydrodynamics Equations

Roger Pawlowski, Eric Phipps, Heidi K. Thornquist, and Roscoe A. Bartlett, Sandia National Laboratories