FY07 ASC Vertical Integration Milestone

Overview, Lessons Learned, and Next Steps

Roscoe A. Bartlett
Department of Optimization & Uncertainty Estimation

Sandia National Laboratories

Trilinos Users Group Meeting, November 8th, 2007

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy under contract DE-AC04-94AL85000.

Sandia National Laboratories

FY07 ASC Vertical Integration Milestone

- **Goal:** Vertically integrate newly developed advanced numerical solver algorithms in Trilinos to build new predictive capabilities
  - **Impact:** Achieved vertical integration of more than 10 Trilinos algorithm packages from parallel linear algebra data structures to transient sensitivities and simulation-constrained optimization!
- **Goal:** Demonstrate new vertically integrated solver algorithms on relevant production applications
  - **Impact:** Solved steady-state parameter estimation problems and transient sensitivities on QASPR-related semiconductor devices => New capabilites!
- **Goal:** Deliver new vertically integrated solvers to ASC customers
  - **Impact:** Release of new packages in Trilinos 8.0!
- **Added Goal:** Explore refined models of collaboration between production application developers and algorithm researchers.
  - **Impact:** Closer collaboration between application and algorithm developers yielding better solvers and better applications

    => Daily Integration and Testing!

Sandia National Laboratories

Categories of Abstract Problems and Abstract Algorithms that Were Vertically Integrated for the Milestone

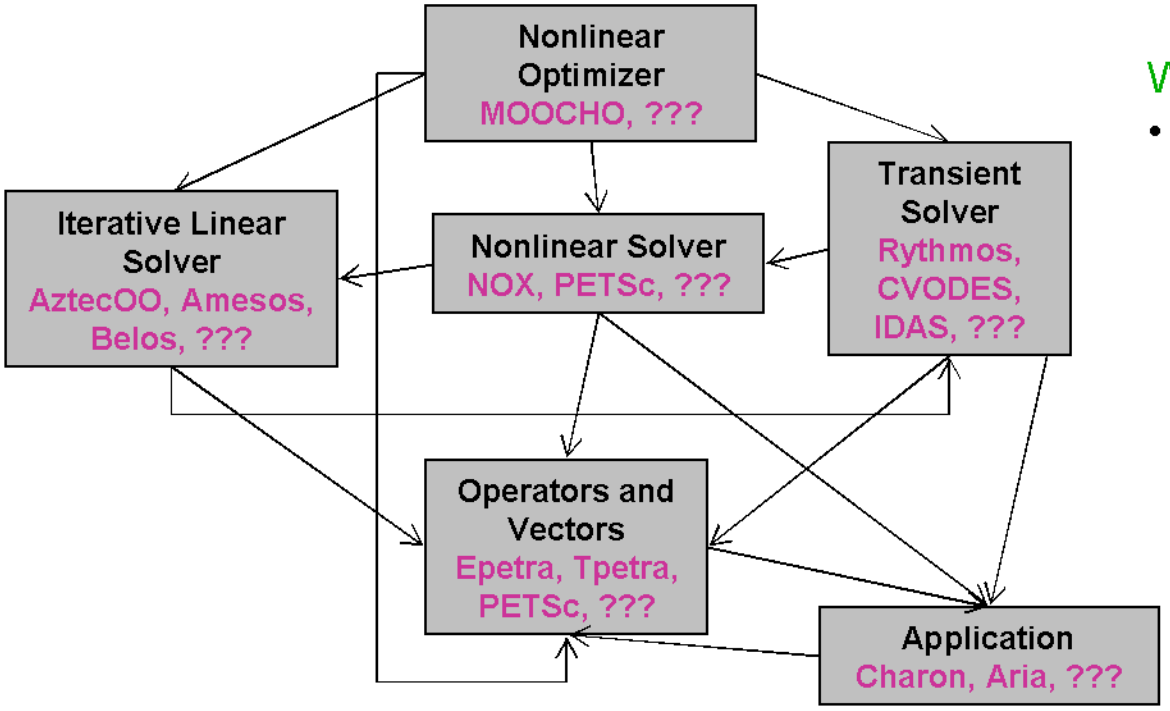
Trilinos Packages

- Linear Problems: Given linear operator (matrix) $A \in \mathbf{R}^{n \times n}$
  - Linear equations: Solve $Ax = b$ for $x \in \mathbf{R}^n$ — Belos
  - Eigen problems: Solve $Av = \lambda v$ for (all) $v \in \mathbf{R}^n$ and $\lambda \in \mathbf{R}$ — Anasazi
- Nonlinear Problems: Given nonlinear operator $f(x,p) \in \mathbf{R}^{n+m} \rightarrow \mathbf{R}^n$
  - Nonlinear equations: Solve $f(x) = 0$ for $x \in \mathbf{R}^n$ — NOX
  - Stability analysis: For $f(x,p) = 0$ find space $p \in \mathcal{P}$ such that $\frac{\partial f}{\partial x}$ is singular — LOCA
- 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 $x(t) \in \mathbf{R}^n, t \in [0,T]$ — Rythmos
- Optimization Problems:
  - Unconstrained: Find $p \in \mathbf{R}^m$ that minimizes $g(p)$ — MOOCHO
  - Constrained: Find $x \in \mathbf{R}^n$ and $p \in \mathbf{R}^m$ that:
    minimizes $g(x,p)$
    such that $f(x,p) = 0$

Sandia National Laboratories

Vertical Interoperability for Transient Optimization

Numerous interactions exist between layered 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
  - Operators/vectors
  - Linear Solvers
  - Nonlinear solvers
  - Transient solvers
  - etc.

Thyra is being developed to address interoperability of ANAs

Key Points

- Higher level algorithms, like optimization, require a lot of interoperability
- Interoperability and layering must be "easy" or these configurations will not be achieved in practice

Sandia National Laboratories

Example of Vertical Interoperability : Rythmos

Solve $f(\dot{x}(t), x(t), t) = 0, t \in [t_0, t_f],\ x(t_0) = x_0,\ \dot{x}(t_0) = \dot{x}_0$

for $x(t) \in \mathbf{R}^n, t \in [t_0, t_f]$

(Time) Stepper

Advance $x(t_k)$ to $x(t_{k+1})$

where $t_{k+1} = t_k + \Delta t_k$

Implicit Backward Euler method

Solve $f\left(\frac{x_{k+1}-x_k}{\Delta t_k}, x_{k+1}, t_{k+1}\right) = 0$ for x_{k+1}

Nonlinear equations

Solve $r(z) = 0$ for $z \in \mathbf{R}^n$

Newton's method (e.g. NOX)

Choose initial guess z_0, tolerance η

for $k = 0, 1, \ldots$

If "converged" **Stop!**

Solve $\frac{\partial r(z_k)}{\partial z}\delta z_k = -r(z_k)$ for δz

Choose α using a line search method

$z_{k+1} = z_k + \alpha \delta z_k$

end for

Linear equations

Solve $Ax = b$ for $x \in \mathbf{R}^n$

Preconditioned GMRES

Iterate to "convergence"

$PAx = Pb$

Operator and Preconditioner applications

Apply $y = Ax$

Apply $y = Px$

Matrix-free or Matrix?

Preconditioners can be defined in many different ways

Sandia National Laboratories

ASC FY07 Level-2 Vertical Integration Milestone (Targets)

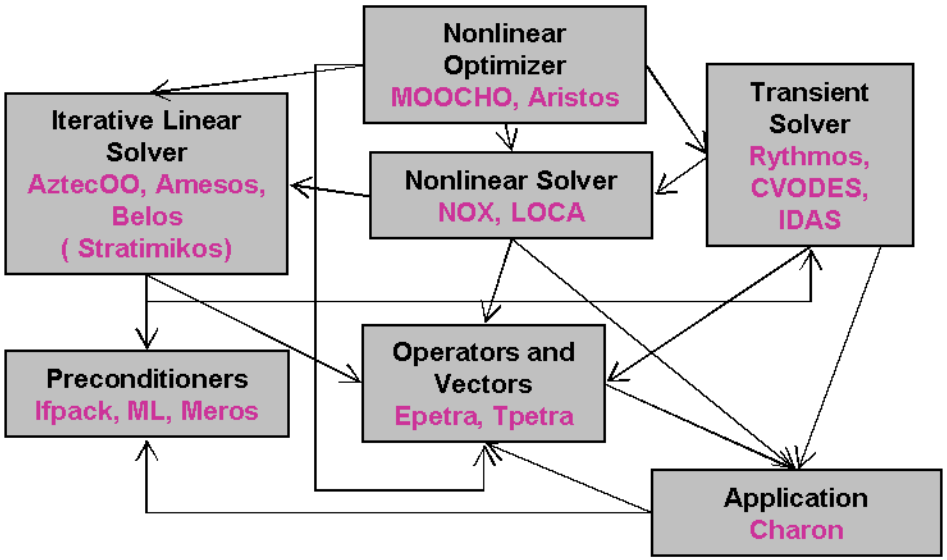
Steady-State Simulation-Constrained Optimization:

Find $x \in \mathbf{R}^n$ and $p \in \mathbf{R}^m$ that:
minimizes $g(x,p)$
such that $f(x,p) = 0$

x: state variables
p: optimization parameters

Transient Simulation-Constrained Optimization:

Find $x(t) \in \mathbf{R}^n$ in $t \in [0,T]$ and $p \in \mathbf{R}^m$ that:
minimizes $\int_0^T g(x(t),p)$
such that $\dot{x} = f(x(t),p,t) = 0$, on $t \in [0,T]$
where $x(0) = x_0$



Targeted Demonstration Problems:

- Steady-state and transient model calibration against experimental data (part of QASPR project) modeled with Charon using parameter estimation optimization using Rythmos, MOOCHO, …

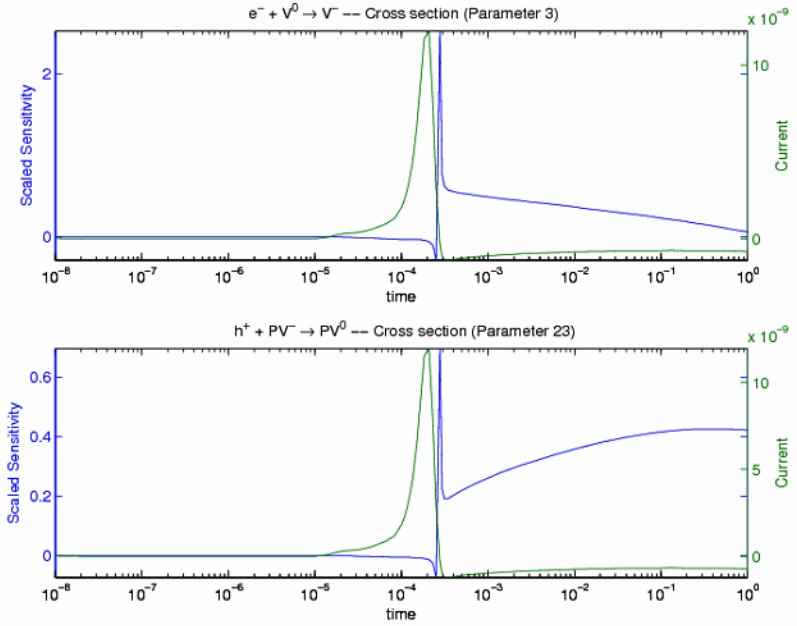
Goal: Demonstrate fully vertical integration from linear algebra all the way through to optimization

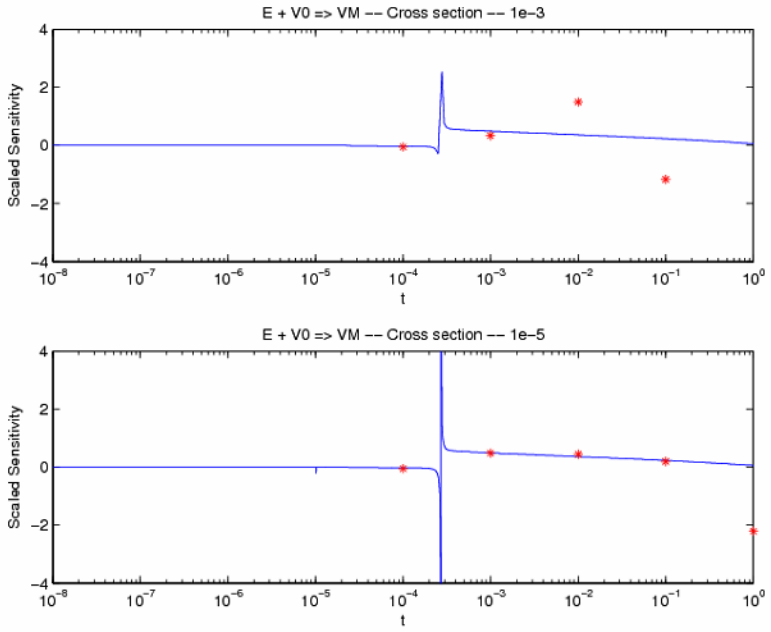
Approach: Create standard interfaces and implementations to break N-to-N coupling using Thyra

Sandia National Laboratories

Transient Sensitivities for Neutron Damage in Stockpile Bipolar Junction Transistor

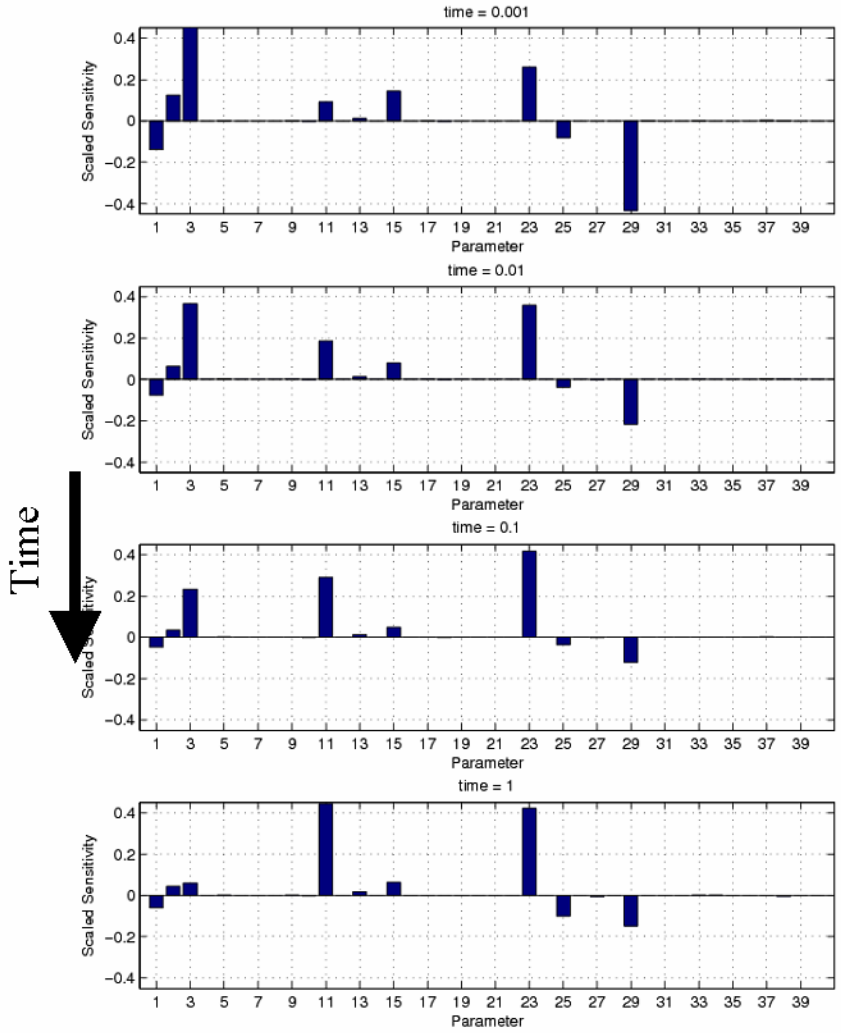
- At least 10 times faster using transient sensitivity solver than finite differences. Even better for full chemistry and substantially more accurate!
- One run (~3 times the cost of single transient run) to generate all 40 sensitivities over the time range (would require 40 individual simulations for (less accurate) finite difference calculation
- We see the expected dominance and time behavior of the (V^0,e):(V^-,h) and (VP^0,e):(VP^-,h) cross-section pairs
- Additionally, the $V^{--} \rightarrow e+V^-$ is shown to have significance





Sandia National Laboratories

Neutron damage mechanism transient sensitivities for stockpile BJT



- Dominant reaction mechanisms vary over time (circled in red)

Reaction Cross-Section Parameters					
Index	Value	Reaction	Index	Value	Reaction
1	3.0e-16	$e^- + V^- \rightarrow V^{--}$	15	1.8e-13	$h^+ + V^- \rightarrow V^0$
2	3.0e-16	$V^{--} \rightarrow e^- + V^-$	16	1.8e-13	$V^0 \rightarrow h^+ + V^-$
3	1.8e-14	$e^- + V^0 \rightarrow V^-$	17	3.0e-15	$h^+ + V^0 \rightarrow V^+$
4	1.8e-14	$V^- \rightarrow e^- + V^0$	18	3.0e-15	$V^+ \rightarrow h^+ + V^0$
5	3.0e-14	$e^- + V^+ \rightarrow V^0$	19	3.0e-16	$h^+ + V^+ \rightarrow V^{++}$
6	3.0e-14	$V^0 \rightarrow e^- + V^+$	20	3.0e-16	$V^{++} \rightarrow h^+ + V^+$
7	3.0e-14	$e^- + V^{++} \rightarrow V^+$	21	3.0e-16	$h^+ + B^- \rightarrow B^0$
8	3.0e-14	$V^+ \rightarrow e^- + V^{++}$	22	7.5e-14	$B^0 \rightarrow h^+ + B^-$
9	3.0e-16	$e^- + P^+ \rightarrow P^0$	23	3.0e-14	$h^+ + PV^- \rightarrow PV^0$
10	1.5e-13	$P^0 \rightarrow e^- + P^+$	24	3.0e-14	$PV^0 \rightarrow h^+ + PV^-$
11	3.0e-15	$e^- + PV^0 \rightarrow PV^-$	25	1.0	$V^{--} + P^+ \rightarrow PV^-$
12	3.0e-15	$PV^- \rightarrow e^- + PV^0$	26	1.0	$V^- + P^0 \rightarrow PV^-$
13	3.0e-14	$h^+ + V^{--} \rightarrow V^-$	27	1.0	$V^- + P^+ \rightarrow PV^0$
14	3.0e-14	$V^- \rightarrow h^+ + V^{--}$	28	1.0	$V^0 + P^0 \rightarrow PV^0$

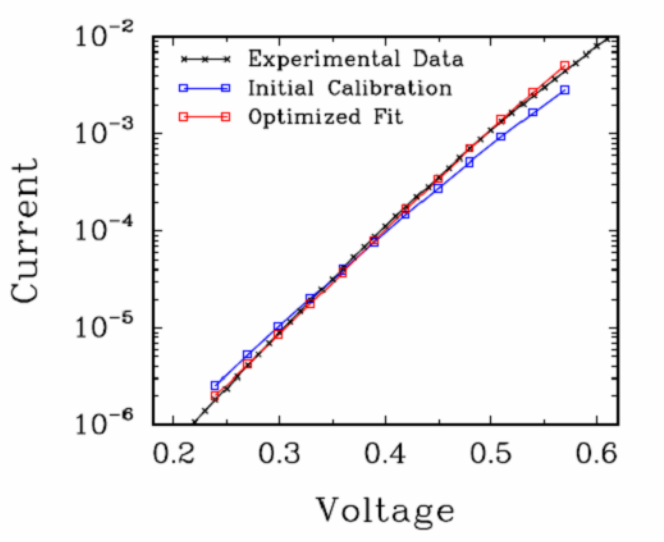
Reaction Activation Energy Parameters					
Index	Value	Reaction	Index	Value	Reaction
29	0.09	$V^{--} \rightarrow e^- + V^-$	35	1.03	$V^- \rightarrow h^+ + V^{--}$
30	0.40	$V^- \rightarrow e^- + V^0$	36	0.72	$V^0 \rightarrow h^+ + V^-$
31	1.07	$V^0 \rightarrow e^- + V^+$	37	0.05	$V^+ \rightarrow h^+ + V^0$
32	0.99	$V^+ \rightarrow e^- + V^{++}$	38	0.13	$V^{++} \rightarrow h^+ + V^+$
33	0.045	$P^0 \rightarrow e^- + P^+$	39	0.045	$B^0 \rightarrow h^+ + B^-$
34	0.44	$PV^- \rightarrow e^- + PV^0$	40	0.68	$PV^0 \rightarrow h^+ + PV^-$

Sandia National Laboratories

Steady-State Parameter Estimation

Minimize	Model current vs. target mismatch
Subject to:	Steady-state semiconductor defect physic FE model

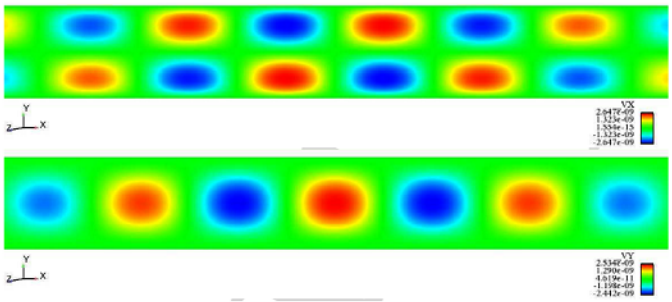
- Solve current matching optimization problem to calibrate model parameters against target currents
- MOOCHO (Bartlett) optimization solver converges FE model and optimality at same time
  - Faster and more robust than black-box optimization methods
  - More accurate solution
- Challenges
  - Extremely difficult nonlinear solver convergences on model convergence
    => Opportunities for algorithm research
  - Inability to match some data
    => May indicate incomplete FE model



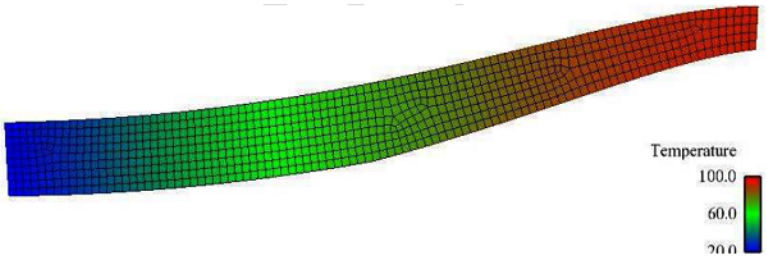
Sandia National Laboratories

Other Demonstrations

- Block eigensolves for reacting flows in Charon
  - Vertical integration of more than 8 Trilinos packages



- Design optimization of a MEMS actuator using ARIA/SIERRA
  - Minimally invasive optimization algorithm in MOOCHO



Impact: Shows generality of vertically integrated algorithms

Sandia National Laboratories

Algorithm and Application Collaboration

- **Fact:** Software is the bridge between algorithms and application (APP) impact

- Current approach of yearly (or bi-yearly) release of algorithms in Trilinos do not always allow for sufficient synergy between algorithm and application developers

- Nightly testing and building of Dev versions of Trilinos and APP:
  - results in better production capabilities and better research,
  - brings algorithm developers and application developers closer together allowing for better exchange of ideas and concerns,
  - makes Trilinos a more customer focused effort,
  - helps drive algorithm development, and
  - reduces barriers to algorithm impact on production APPs

Impact: Charon + Trilinos should serve as example for other APPs

Sandia National Laboratories

New Trilinos Functionality Achieved During Milestone

- Thyra adapters expanded to other packages
  - NOX
  - LOCA
- New Thyra/Stratimikos support software
  - Multi-period model evaluator (never used in Charon through …)
  - Delayed linear solver creation (Thyra::DelayedLinearOpWithSolve[Factory])
  - ???
- Rythmos developments
  - Major refactoring of Rythmos interfaces
  - New integrator interface and subclasses
  - Forward sensitivity stepper
- Misc.
  - Major Belos refactoring
  - ???
- Is there anything I missed?

Sandia National Laboratories

Examples of Trilinos/Thyra Problems Encountered

- Parameter sublist handling error with Belos and ML
  - Removed embedded preconditioner sublist support
  - ToDo: We need a lot more tests for Stratimikos!
- Broken explicit transpose in Epetra
  - Mike Heroux fixed this once we identified the problem
- Memory leak in AztecOO/Thyra adapters caused by circular RCP references
  - I removed an unnecessary RCP
  - I massively improved debug detection of circular dependencies! (Teuchos_RCP.cpp)
  - ToDo: We need to include stress tests!
  - ToDo: We need to get valgrind testing going!
  - ToDo: Need to clean up the AztecOO/Thyra adapters … these are a mess …
- Unnecessary preconditioner construction in NOX/LOCA
  - Incompatible model of change management between Thyra::ModelEvaluator and NOX/LOCA Group
  - Developed new delayed linear solver creation for all Stratimikos solvers
  - ToDo: Need to change NOX/LOCA handing of linear solver creation
- Were there others that I did not list or don't know about?

Sandia National Laboratories

Enabling/Supporting New Production Capabilities in Charon

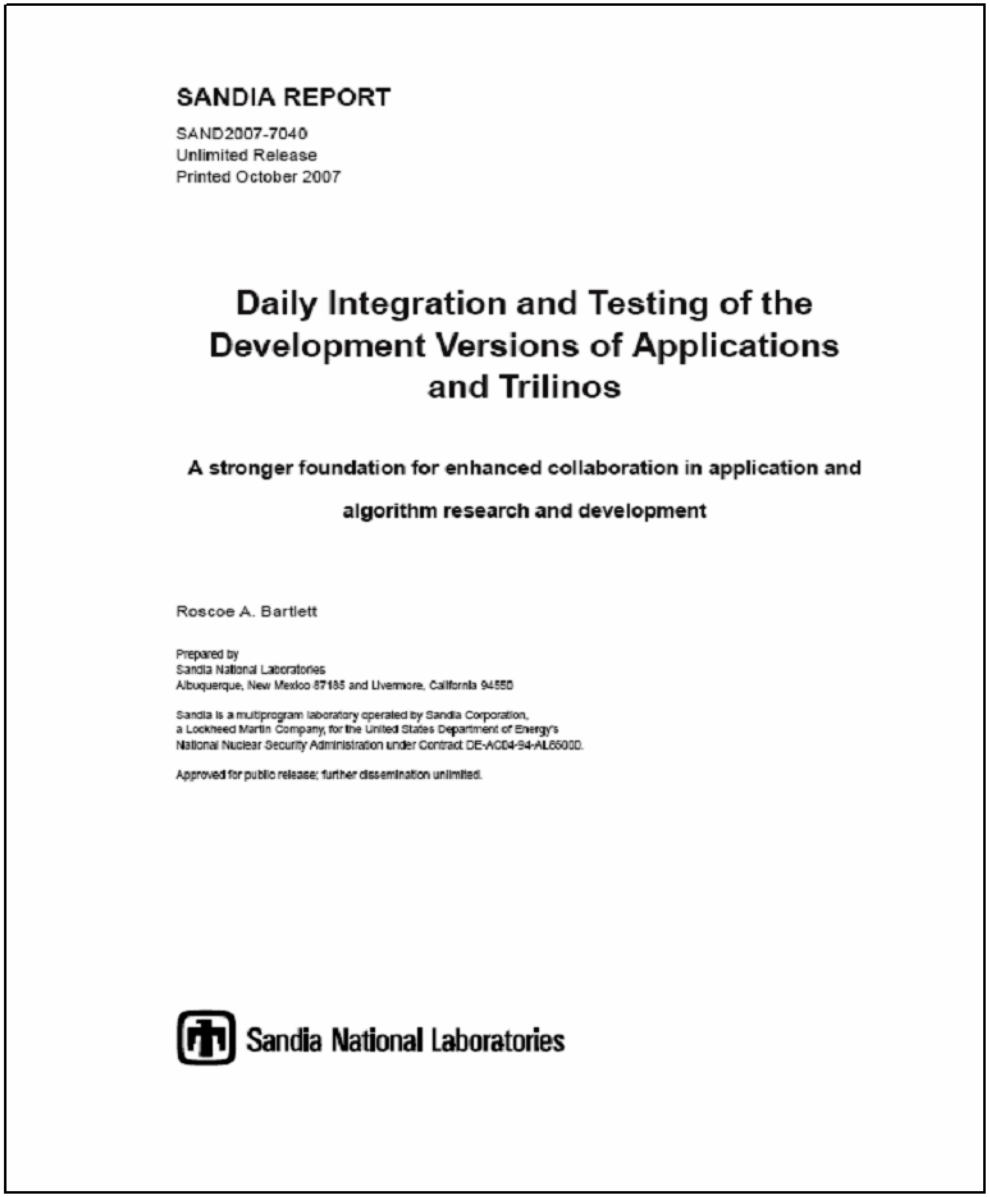
- Parameter list support
  - Complex sublists => Move to XML specification
  - Get Rythmos/Moocho sublists out of the NOX sublist
  - Specification of “model parameters” is confusing and inflexible
  - …
- Better support for transient sensitivities
  - Creation and outputting of sensitivities at end of each time step
  - Outputting sensitivites to the Exodus file
- Low hanging fruit?
  - Transient current matching and sensitivities?
  - Transient parameter estimation problems?
  - Steady-state doping profile matching (MOOCHO & Aristos)?
- Related R&D
  - Physics-based preconditioners in Charon using Thyra/Stratomikos?
  - Driving more optimization algorithm research?
- Is there anything else that I am not thinking of?

Sandia National Laboratories

Overall Lessons Learned

- Thyra (for the most part) worked just fine and did more that we where shooting for …
- Focus on the problem that the customer actually needs to solve even if it is really hard to solve
  - This makes V&V easier but this may make it hard to make progress
  - The difficulty is, how do we make progress in the mean time and how to we hit these targets?
- How much more effort should we put into making this a usable production capability for the customer
  - How user friendly and user-proof do we need to make this?
  - When do we hand over more responsibility to the customer?
- How can production applications drive our research without us being totally bogged down on solving their problems
  - APP + Trilinos Dev daily integration and testing …
  - Just because the APP wants lots of less research-directed work does not mean that we can actually do this in all cases?

APP + Trilinos Dev SAND Report



SANDIA REPORT

SAND2007-7040
Unlimited Release
Printed October 2007

Daily Integration and Testing of the Development Versions of Applications and Trilinos

A stronger foundation for enhanced collaboration in application and algorithm research and development

Roscoe A. Bartlett

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of Energy's
National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

Sandia National Laboratories

http://software.sandia.gov/SESS/common/AppPlusTrilinosDev.pdf

Sandia National Laboratories

Overview of Current APP + Trilinos Interactions

- Many active research collaborations have existed and continue to exist between applications (APPs) and Trilinos where APP + Trilinos Dev is developed throughout year and drives algorithm and APP development
- Currently, Trilinos has a major release once a year (typically in September, at the end of each Fiscal Year)
  - Some minor releases go out from time to time (e.g. Trilinos 7.1.x for ML)
  - Several minor minor releases (bug fixes) go out frequently (e.g. 7.0.9)
- Potential issues
  - Porting of some Trilinos packages to some customer platforms is only done right before a major release => Portability problems
  - May be very difficult to update the APP for new Trilinos release (e.g. porting problems, failing and diffing tests, etc.)
  - As much as year or more between introduction of new Trilinos capability and impact on end customer (through an APP release)
  - Post delivery maintenance after the introduction of a new capability

Obstacles to Algorithms Impacting Applications

- Technical Challenges
  - Features needed by algorithm from APP are difficult to implement
    e.g. Accurate parameter derivatives, in-core response functions
  - Algorithm is unproven on APP problem area
  - Inability of algorithm to scale to APP sizes (# DOFs and # processes)
  - Algorithm too complex to easily implement by APP developers
    => Have algorithm experts implement the algorithms =>Trilinos!
  - Others …
- Difficulty of collaboration between algorithm and APP developers
  - Difficult to access shared platforms
  - Difficult to build Dev versions algorithm and APP codes together
    - Software configuration management problems
    - Development versions of APP and algorithm codes incompatible

  => Even smallest overhead can kill a collaboration

Sandia National Laboratories

Obstacles to Algorithms Impacting Applications

- Examples of missed opportunities in FY07
  - Aria + Rythmos
    - Pat Notz (1514) had some funding to put Rythmos into Aria
    - There where some issue in the ModelEvaluator software
    - Could not easily access Dev version of Trilinos to address issues!
  - Alegra + Thyra/Stratimikos
    - Tom Brunner (1641) wanted to experiment with Thyra/Stratimikos to build some physics-based preconditioners

      PLUG: Thyra talk 3:30 PM Tuesday 11/6 at Trilinos User Group (TUG) Meeting in CSRI/90!

      http://trilinos.sandia.gov/events/trilinos_user_group_2007
    - Using released version of Trilinos and must port to Purple => Porting problems
    - Could not easily access Dev version of Trilinos to address issues!

Future of APP + Trilinos Interactions?

- Many applications are reluctant to take on direct Trilinos dependences
  - Example: APP will not allow use of Teuchos::RCP at application level
- Trilinos capability set is expanding:
  - Automatic differentiation (Sacado)
    - A fundamental type of dependency!
  - PDE discretization support (Intrepid)
  - Mesh support (phdMesh)
  - Finite element assembly (FEI)
  - …
- Will applications take on these new dependences in a significant way?
  - Can algorithm development in Trilinos lead to "Transformational" changes in APPs when treaded as a Third Party Library (TPL)?
- My assertion: For applications to take on more serious Trilinos dependences they must have more direct interaction with and control of these services!

Sandia National Laboratories

Charon + Trilinos Dev Daily Integration &Testing

- Motivation for nightly building and testing of Charon + Trilinos Dev
  - Avoid backslides in Vertical Integration Milestone capabilities
    - Before nightly testing, backslides of capability happened!
  - Proof of Milestone capabilities
  - Preservation of milestone problems

Sandia National Laboratories

Manifesto for Agile Software Development

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items on
the right, we value the items on the left more.

Kent Beck (XP)
Mike Beedle
Arie van Bennekum
Alistair Cockburn (Crystal Clear)
Ward Cunningham
Martin Fowler (UML, Refactoring)
James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick
Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas

http://agilemanifesto.org

Important Aspects of Agile (and other) Software Methods

- Frequent Integration of the Software Product is Critical

  "Whatever integration strategy you select, a good approach to integrating the software in the "**daily** build and smoke test"."

  Steve McConnell, *Code Complete: Second Edition*

- Close Customer Involvement and Interaction is Key

  "In order for a project to be agile, there must be **significant and frequent** interaction between the customers, developers, and stakeholders"

  Robert C. Martin, *Agile Software Development*

- Frequent Delivery of Capability to Customers is a Top Priority

  "The single most important property of any project, large or small, agile or not, is that of delivering running, tested code to real users **every few months**. The advantages are so numerous that it is astonishing that any team doesn't do this."

  Alistair Cockburn, *Crystal Clear*

Questions: Where does Trilinos and APP fit in? Who is the "customer", "developer", "stakeholder"?

Sandia National Laboratories

Where does Trilinos Fit It?

- The term “Agile” has been kicked around for research-driven software
  - All “Agile” methods involve Frequent Integration and Frequent Delivery
  - What does this mean for Trilinos?
- What is Trilinos with respect to “the product”, “the customer”, software integration, and software delivery?
  - Is Trilinos an integrated project just within itself?

    =>Just integrate Trilinos with itself and we are Agile?
  - Is Trilinos critical enabling technology that is used in real products?

    =>Trilinos needs to frequently integrate with APPs to be Agile?
  - Do APP requirements continue to drive Trilinos development?

    =>Trilinos needs to frequently integrate to let APPs drive Trilinos?
- I would argue that for Trilinos to be “Agile”, we have to frequently integrate with at least our immediate APP customers (even when active research collaborations are not taking place)
- What about frequent delivery?

Sandia National Laboratories

Principles for APP + Trilinos Dev Daily Integration & Testing

- Support active collaboration between APP and Trilinos developers
- Guard against destabilizing the separate development of APP and Trilinos
- Minimize unnecessary interactions between APP and Trilinos Developers
- Streamline communication between APP and Trilinos developers when such communication is needed
- Accountability for keeping APP + Trilinos Dev working

Sandia National Laboratories

Outline for APP + Trilinos Dev Integration & Testing

- APP developers work mainly with APP + Trilinos Release

  => Minimize unnecessary interaction with Trilinos developers

- Trilinos developers work mainly with Trilinos Dev

  => Minimize unnecessary interaction with APP developers

- Hide APP + Trilinos Dev “research” work in APP behind ifdefs

  => Keep both versions of the code close together (no branches)

- Segregate “production” and “research” tests

  => Differentiate between APP and Trilinos defects

- Perform nightly building and testing of APP + Trilinos Release and APP + Trilinos Dev

- Release APP + Trilinos together or staged

- Continue APP + Trilinos Dev nightly building and testing after APP upgrades to new Trilinos release

Sandia National Laboratories

Outline for APP + Trilinos Dev Integration & Testing

- Perform nightly building and testing of APP + Trilinos Release/Dev
  - APP + Trilinos Release tested against "production" tests
    - Send "production" failures only to APP developers
  - APP + Trilinos Dev tested against "research" and "production" tests
    - Only send "production" failures that did not also fail in APP + Trilinos Release to Trilinos developers
    - All "research" failures go to appropriate Trilinos (or APP) developers

Sandia National Laboratories

Outline for APP + Trilinos Dev Integration & Testing

- Release APP + Trilinos together or staged
  - Combined tagging and release of APP + Trilinos
    - Requires instant releasability of Trilinos
    - Advantage: Most up to date algorithms to customers
    - Disadvantage: Supporting multiple Trilinos releases
  - Staged releases of Trilinos and APP
    - Two approaches to handle Trilinos release
      - a) APP updates to new Trilinos Release right away
        - Build & test against i) Trilinos Release and ii)Trilinos Dev
      - b) APP delays update to new Trilinos release
        - Build & test against i) old Trilinos release, ii) current Trilinos release, and iii) Trilinos Dev
    - Next APP release is performed against latest Trilinos Release
    - Advantage: Fewer, more predictable Trilinos releases
    - Disadvantage: More delay in getting new Trilinos capabilities to end customers

Sandia National Laboratories

Research & Production Advantages for APP + Trilinos Dev

- Research Advantages
  - Reduces overhead for initial algorithm integration
  - Improves chances that new algorithms will have impact
  - Preserves interesting/challenging problems
- Production Advantages
  - Expands testing for Trilinos
  - Enables better scalability testing for Trilinos
  - Reduces time to detection of defects
  - Reduces release time and effort
  - Allows for more aggressive refactorings and code improvements
  - Better address customer needs
  - Reduces all kinds of risk

Sandia National Laboratories

Outline of Suggested Practices for APP + Trilinos Dev

- Separate ``production'' and ``research'' tests
- Refactor APP code to isolate and separate ifdefed code
- Maintain a dedicated machine for building and testing APP + Trilinos Dev
- Appoint a dedicated APP + Trilinos Representative
- Provide easy access for any Trilinos or APP developer to build, test, and develop APP + Trilinos Dev
- Fix failed builds of APP + Trilinos Dev ASAP
- Address failing ``research'' and ``production'' tests on a schedule appropriate for the APP + Trilinos collaboration
- Archive test results for sufficiently long periods of time
- Transition ``research'' to ``production'' appropriately after each Trilinos release
- Perform APP + Trilinos Release and APP + Trilinos Dev nightly testing on the same set of platforms

Sandia National Laboratories

Experience from Charon + Trilinos Dev from Milestone Work

- Timeline for Charon + Trilinos Dev Daily Integration and Testing
  - ???: Roger Pawlowski gets Charon building against Trilinos Dev
  - 10/30/06: First Charon+Trilinos Dev milestone-related test added
  - 12/2/06 - 1/7/07: Did not build or test Charon + Trilinos Dev
  - 1/7/07: Rebuild Charon + Trilinos Dev and a test failed!
  - 1/27/07: Deployed initial nightly building & testing of Charon + Trilinos Dev
  - 7/9/07: Branch for Trilinos 8.0
    - We did not build against Trilinos 8.0 (NOT GOOD)
  - 7/24/07: Charon updated to build only against Charon + Trilinos 8.0
    - We did not build against Trilinos Dev (NOT GOOD)
  - 9/21/07: Nightly testing updated to build against Trilinos 7.0, 8.0, and Dev
    - **This is the way to do it!**
  - 10/18/07: Nevada refactoring committed to Charon => Charon + Trilinos Dev goes down!
  - 10/25/07: Nightly testing of Charon + Trilinos 8.0 & Dev back up (Not 7.0!)
  - ???: Charon updated to Trilinos 8.0, drops support for Trilinos 7.0

Sandia National Laboratories

Experience from Charon + Trilinos Dev from Milestone Work

- A few observations Charon + Trilinos Dev Daily Integration and Testing
  - Only a few of weeks total developing and maintaining test scripts
  - Over the course of the milestone, I was able to diagnose most failed builds and failing tests and send out emails in less than 10 minutes
  - All major milestone capabilities are tested every night
  - Charon + Trilinos Dev Daily integration provides a solid foundation for future algorithmic work

APP + Trilinos Dev Checklist

- ❑ Do you have ifdefs in place in APP code to build against Trilinos Dev and against a stable release (or multiple releases) of Trilinos?
- ❑ Have you separated ``production'' and ``research'' tests?
- ❑ Have you appointed an official APP + Trilinos Representative?
- ❑ Have you set up a dedicated machine to do nightly building and testing of APP + Trilinos Dev?
- ❑ Have you provided easy access to APP and Trilinos developers to immediately build a private version of APP + Trilinos Dev?
- ❑ Do you fix failing builds of APP + Trilinos Dev right away, with no exceptions?
- ❑ Do you address failing ``production'' and ``research'' tests with an urgency that is appropriate for the nature of APP and the APP + Trilinos collaboration?
- ❑ Do you archive test results long enough to diagnose failing tests?
- ❑ After each major release of Trilinos, do you upgrade APP to the new release in a timely way?
- ❑ During the transitionary period between Trilinos branch and release do you build APP against the old Trilinos release, the current Trilinos release, and Trilinos Dev?

Conclusions

- Nightly building and testing of the development versions of the application and Trilinos:
  - results in better production capabilities and better research,
  - brings algorithm developers and application developers closer together allowing for a better exchange of ideas and concerns,
  - refocuses Trilinos developers on customer efforts,
  - helps drive continued research-quality algorithm development, and
  - reduces barriers for new algorithms to have impact on production applications.
- Other application projects and scientific support software projects should consider adopting the type of continuous integration that is used with Charon + Trilinos that was developed as part of the ASC Vertical Integration Milestone work.

Next Steps for APP + Trilinos Dev

- Charon + Trilinos Dev Daily Integration and Testing
  - Continuing Nightly building and tested indefinently?
  - Upgrade scripts to use Python?
  - Who will take over as Charon + Trilinos Representative?
  - What new algorithms research will this support?
- Aria/SIERRA + Trilinos Dev Daily Integration and Testing
  - Machine/enviornment being pursued by Russell Hooper (SIERRA + Trilinos Rep.) and Jim Stewart and others
  - Being set up to support Aria/Rythmos ASC Outer Core work and Fuego/ML LDRD work
- Others?
- Continuous integration Server? *Continuous Integration*, Paul Duvall et. al
  - Build and test Trilinos and APP + Trilinos several times a day
  - Avoid checking out broken code
  - …

Sandia National Laboratories

THE END

Sandia National Laboratories

Some of the Problems Solved During Milestone

Nonlinear equations:	Solve $f(x) = 0$ for $x \in \mathbf{R}^n$
Stability analysis:	For $f(x,p) = 0$ find space $p \in \mathcal{P}$ such that $\frac{\partial f}{\partial x}$ is singular
DAEs/Implicit ODEs:	Solve $f(\dot{x}(t), x(t), t) = 0, t \in [0,T]$, $x(0) = x_0$, $\dot{x}(0) = x'_0$ for $x(t) \in \mathbf{R}^n, t \in [0,T]$
DAE/Implicit ODE Forward Sensitivities:	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 $x(t) \in \mathbf{R}^n, t \in [0,T]$
Constrained Optimization:	Find $x \in \mathbf{R}^n$ and $p \in \mathbf{R}^m$ that: minimizes $g(x,p)$ such that $f(x,p) = 0$
ODE Constrained Optimization:	Find $x(t) \in \mathbf{R}^n$ in $t \in [0,T]$ and $p \in \mathbf{R}^m$ that: minimizes $\int_0^T g(x(t),p)$ such that $\dot{x} = f(x(t),p,t) = 0$, on $t \in [0,T]$ where $x(0) = x_0$

Sandia National Laboratories