Advanced Analysis, Sensitivity, and Optimization Capabilities in the Trilinos Collection

Roscoe A. Bartlett
Department of Optimization & Uncertainty Estimation

Sandia National Laboratories

Frontiers of Geophysical Simulation

August 20, 2009

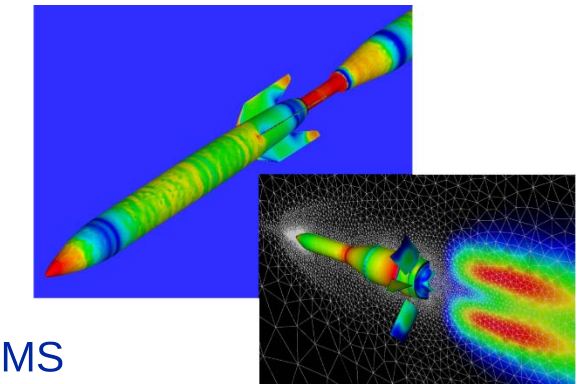
Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company,
for the United States Department of Energy under contract DE-AC04-94AL85000.

Overview of Trilinos

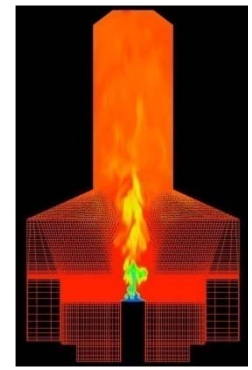
Sandia Physics Simulation Codes

- **Element-based**
  - **Finite element, finite volume, finite difference, network, etc…**
- **Large-scale**
  - **Billions of unknowns**
- **Parallel**
  - **MPI-based SPMD**
  - **Distributed memory**
- **C++**
  - **Object oriented**
  - **Some coupling to legacy Fortran libraries**

Fluids

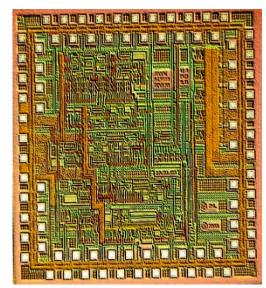


Combustion

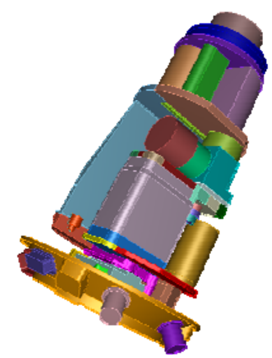


MEMS

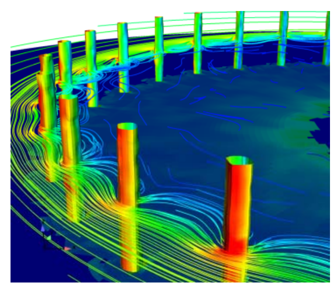
Circuits



Structures



Plasmas



Sandia National Laboratories

Motivation For Trilinos

- Sandia does LOTS of solver work.
- 11 years ago …
  - Aztec was a mature package. Used in many codes.
  - FETI, PETSc, DSCPack, Spooles, ARPACK, DASPK, and many other codes were (and are) in use.
  - New projects were underway or planned in multi-level preconditioners, eigensolvers, non-linear solvers, etc…
- The challenges:
  - Little or no coordination was in place to:
    - Efficiently reuse existing solver technology.
    - Leverage new development across various projects.
    - Support solver software processes.
    - Provide consistent solver APIs for applications.
  - ASCI was forming software quality assurance/engineering (SQA/SQE) requirements:
    - Daunting requirements for any single solver effort to address alone.

Sandia National Laboratories

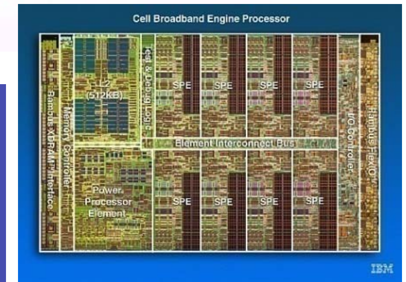
Evolving Trilinos Solution

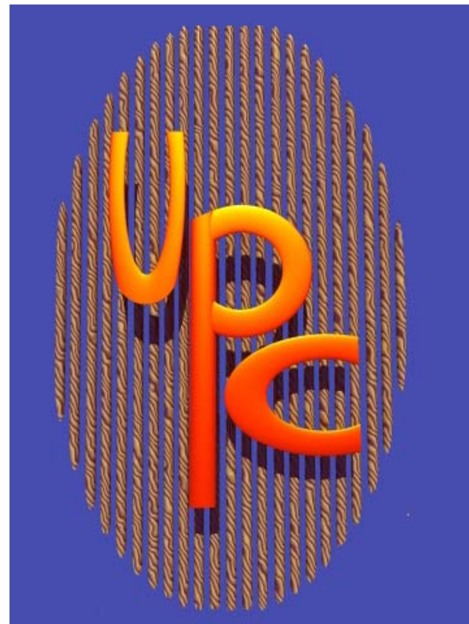
- **Trilinos[1] is an evolving framework to address these challenges:**
  - **Follow a TOOLKIT approach.**
  - **Fundamental atomic unit is a *package*.**
  - **Includes core set of vector, graph and matrix classes (Epetra/Tpetra packages).**
  - **Provides a common abstract solver API (Thyra package).**
  - **Provides a ready-made package infrastructure (new_package package):**
    - **Source code management (cvs, bonsai => Moving to git).**
    - **Build tools (CMake [New]).**
    - **Automated regression testing (CTest/CDash [New]).**
    - **Communication tools (mailman mail lists).**
  - **Specifies requirements and suggested practices for package SQA.**
- **In general allows us to categorize efforts:**
  - **Efforts best done at the Trilinos framework level (useful to most or all packages).**
  - **Efforts best done at a package level (peculiar or important to a package).**
  - Allows package developers to focus only on things that are unique to their package.

1. Trilinos loose translation: “A string of pearls”

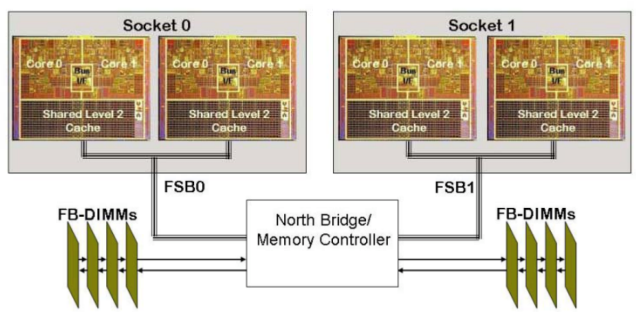
Sandia National Laboratories

Target Platforms: Any and All





- **Desktop: Development and more…**
  - Native MS Windows (Visual C++, CMake)
  - Native MAC (XCode, CMake)
- **Capability machines:**
  - Redstorm (XT3), Clusters
  - Roadrunner (Cell-based).
  - Multicore nodes.
- **Parallel software environments:**
  - MPI of course.
  - UPC, CAF, threads, vectors,…
  - Combinations of the above.
- **User "skins":**
  - C++/C, Python
  - Fortran
  - Web, CCA



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.

Algorithmic Goals

- ***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.

Software Goals

Sandia National Laboratories

Trilinos Package Summary

	Objective	Package(s)
Discretizations	Meshing & Spatial Discretizations	phdMesh, Intrepid, Phalanx, Shards, Pamgen, Sundance, FEI
	Time Integration	Rythmos
Methods	Automatic Differentiation	Sacado
	Mortar Methods	Moertel
Core	Linear algebra objects	Epetra, Jpetra, Tpetra
	Abstract interfaces	Thyra, Stratimikos, RTOp
	Load Balancing	Zoltan, Isorropia
	“Skins”	PyTrilinos, WebTrilinos, Star-P, ForTrilinos, CTrilinos
	C++ utilities, I/O, thread API	Teuchos, EpetraExt, Kokkos, Triutils, TPI
Solvers	Iterative (Krylov) linear solvers	AztecOO, Belos, Komplex
	Direct sparse linear solvers	Amesos
	Direct dense linear solvers	Epetra, Teuchos, Pliris
	Iterative eigenvalue solvers	Anasazi
	ILU-type preconditioners	AztecOO, IFPACK
	Multilevel preconditioners	ML, CLAPS
	Block preconditioners	Meros
	Nonlinear system solvers	NOX, LOCA
	Optimization	MOOCHO, Aristos, GlobiPack, OptiPack, TriKota
	Stochastic PDEs	Stokhos

Sandia National Laboratories

Automatic Intra-Package Dependency Handling

Packages	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32	P33	P34	P35	P36	P37	P38	P39	P40	P41	P42	P43	P44	P45	P46	Packages
P01) Teuchos	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P01) Teuchos
P02) RTOp	LR	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P02) RTOp
P03) Kokkos	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P03) Kokkos
P04) Epetra	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P04) Epetra
P05) Stokhos	LR	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P05) Stokhos
P06) Sacado	LO	.	.	.	LO	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P06) Sacado
P07) Zoltan	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P07) Zoltan
P08) Shards	TO	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P08) Shards
P09) Intrepid	LR	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P09) Intrepid
P10) Triutils	.	.	.	LR	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P10) Triutils
P11) Tpetra	LR	.	.	ITO	.	.	.	.	.	TO	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P11) Tpetra
P12) EpetraExt	LR	.	.	LR	.	.	.	.	.	LO	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P12) EpetraExt
P13) Thyra	LR	LR	.	LO	.	.	.	.	.	ILO	.	LO	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P13) Thyra
P14) Isorropia	LR	.	.	LR	.	.	LR	.	.	ILO	.	LO	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P14) Isorropia
P15) Pliris	.	.	.	LR	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P15) Pliris
P16) Claps	.	.	.	LR	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P16) Claps
P17) AztecOO	LO	.	.	LR	.	.	.	.	.	LR	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P17) AztecOO
P18) Galeri	LR	.	.	LR	.	.	.	.	.	ILO	.	LO	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P18) Galeri
P19) Amesos	LR	.	.	LR	.	.	.	.	.	TO	.	LO	.	.	.	.	.	TO	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P19) Amesos
P20) Ifpack	LR	.	.	LR	.	.	.	.	.	ILO	.	ILO	.	.	.	.	LO	TO	LO	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P20) Ifpack
P21) Komplex	LR	.	.	LR	.	.	.	.	.	ILR	.	.	.	.	.	.	LR	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P21) Komplex
P22) ML	LO	.	.	LO	.	.	LO	.	.	ILO	.	LO	.	LO	.	.	LO	LO	LO	LO	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P22) ML
P23) Belos	LR	.	.	LR	.	.	ITO	.	.	TO	TO	TO	.	ITO	.	.	ITO	ITO	ITO	TO	.	TO	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P23) Belos
P24) Stratimikos	ILR	ILR	.	ILO	.	.	ILO	.	.	TO	.	TO	LR	ILO	.	.	LO	ILO	LO	LO	.	LO	LO	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P24) Stratimikos
P25) Meros	ILR	ILR	.	LR	.	.	ITO	.	.	ITR	.	ILO	LR	ITO	.	.	TR	ITO	ITO	ITO	.	ITO	ITO	TR	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P25) Meros
P26) FEI	LO	.	.	LO	.	.	ILO	.	.	ILO	.	ILO	.	ILO	.	.	LO	ILO	LO	LO	.	LO	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P26) FEI
P27) RBGen	LR	.	.	LO	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P27) RBGen
P28) Anasazi	LR	ILO	.	LO	.	.	.	.	.	TO	LO	TO	LO	.	.	.	TO	.	TO	TO	.	.	TO	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P28) Anasazi
P29) ThreadPool	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P29) ThreadPool
P30) Phalanx	LR	.	.	TO	ILO	LR	.	.	.	ITO	.	ITO	.	.	.	.	ITO	.	ITO	TO	.	.	TO	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P30) Phalanx
P31) Pamgen	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P31) Pamgen
P32) Phdmesh	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	LR	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	.	P32) Phdmesh
P33) NOX	LR	ILO	.	LO	ILO	LO	ILO	.	.	ILO	ILO	LO	LO	ILO	.	.	LO	ILO	LO	LO	.	LO	LO	TO	.	.	.	LO	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	.	P33) NOX
P34) Moertel	LR	.	.	LR	.	.	ILO	.	.	ILO	.	LR	.	ILO	.	.	ILO	ILO	LR	ILO	.	LR	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	.	.	.	.	P34) Moertel
P35) TrilinosCouplings	ILO	ILO	.	ILO	ILO	ILO	ILO	.	.	ILO	ILO	LO	ILO	LO	.	.	LO	ILO	LO	LO	.	LO	LO	.	.	.	.	ILO	.	.	.	.	LO	.	X	.	.	.	.	.	.	.	.	.	.	.	P35) TrilinosCouplings
P36) Rythmos	LR	ILR	.	ILO	ITO	TO	ITO	.	.	ILO	ITO	TO	LR	ITO	.	.	ITO	ITO	ITO	ITO	.	ITO	ITO	TO	.	.	.	ITO	.	.	.	.	TO	.	.	X	.	.	.	.	.	.	.	.	.	.	P36) Rythmos
P37) MOOCHO	LR	LR	.	ILO	.	.	ITO	.	.	ILO	.	TO	LO	ITO	.	.	ITO	ITO	ITO	ITO	.	ITO	ITO	TO	.	.	.	.	.	.	.	.	.	.	.	TO	X	.	.	.	.	.	.	.	.	.	P37) MOOCHO
P38) Aristos	LR	.	.	LR	.	.	ILO	.	.	TO	.	LO	.	ILO	.	.	LO	ILO	LO	LO	.	LO	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	.	.	P38) Aristos
P39) Sundance	LR	ILR	.	LR	ILO	ILO	ILO	.	.	ILR	ILO	ILO	LR	ILO	.	.	LR	ILO	LR	LR	.	LR	ILO	LR	.	.	.	LR	.	.	.	.	LR	.	.	.	LO	.	X	.	.	.	.	.	.	.	P39) Sundance
P40) TriKota	LR	.	.	LR	.	.	.	.	.	ILO	.	LR	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	.	P40) TriKota
P41) CTrilinos	LR	.	.	LR	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	.	.	.	P41) CTrilinos
P42) ForTrilinos	TR	.	.	TR	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	LR	X	.	.	.	.	P42) ForTrilinos
P43) PyTrilinos	LO	LO	LO	LO	ILO	LO	ILO	.	.	LO	LO	LO	LO	LO	.	LO	LO	LO	LO	LO	LO	LO	LO	LO	LO	.	.	LO	.	.	.	.	LO	LO	.	LO	LO	.	.	.	.	.	X	.	.	.	P43) PyTrilinos
P44) WebTrilinos	ITR	.	.	TR	.	.	ITO	.	.	ITR	.	ITO	.	ITO	.	.	TR	ITO	TR	TR	.	TR	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	.	.	P44) WebTrilinos
P45) Didasko	TO	ITO	.	TO	ITO	ITO	ITO	.	.	TO	TO	TO	ITO	ITO	.	.	TO	ITO	TO	TO	.	TO	TO	.	.	.	.	TO	.	.	.	.	TO	.	.	.	.	.	.	.	.	.	.	.	X	.	P45) Didasko
P46) NewPackage	LO	.	.	LR	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	X	P46) NewPackage
Packages	P01	P02	P03	P04	P05	P06	P07	P08	P09	P10	P11	P12	P13	P14	P15	P16	P17	P18	P19	P20	P21	P22	P23	P24	P25	P26	P27	P28	P29	P30	P31	P32	P33	P34	P35	P36	P37	P38	P39	P40	P41	P42	P43	P44	P45	P46	Packages

- Trilinos/cmake/python/data/TrilinosPackageDependenciesTable.html

Sandia National Laboratories

(Partial) List of People Who Have Developed Trilinos

Chris Baker
Developer of Anasazi, RBGen, Tpetra

Ross Bartlett
Lead Developer of Thyra, Stratimikos and MOOCHO
Developer of Rythmos, Teuchos

Pavel Bochev
Project Lead and Developer of Intrepid

Paul Boggs
Developer of Thyra

Eric Boman
Lead Developer of Isorropia
Developer of Zoltan

Todd Coffey
Lead Developer of Rythmos

David Day
Developer of Komplex and Intrepid

Karen Devine
Lead Developer of Zoltan

Clark Dohrmann
Developer of CLAPS

Michael Gee
Developer of ML, NOX

Bob Heaphy
Lead Developer of Trilinos SQA

Mike Heroux
Trilinos Project Leader
Lead Developer of Epetra, AztecOO,
Kokkos, Komplex, IFPACK, Thyra, Tpetra
Developer of Amesos, Belos, EpetraExt, Jpetra

Ulrich Hetmaniuk
Developer of Anasazi

Robert Hoekstra
Lead Developer of EpetraExt
Developer of Epetra, Thyra, Tpetra

Russell Hooper
Developer of NOX

Vicki Howle
Lead Developer of Meros
Developer of Belos and Thyra

Jonathan Hu
Developer of ML

Sarah Knepper
Developer of Komplex

Tammy Kolda
Lead Developer of NOX

Joe Kotulski
Lead Developer of Pliris

Rich Lehoucq
Developer of Anasazi and Belos

Kevin Long
Lead Developer of Thyra, Sundance
Developer of Teuchos

Roger Pawlowski
Lead Developer of NOX, Phalanx
Developer of Shards, LOCA

Michael Phenow
Trilinos Webmaster
Lead Developer of New_Package

Eric Phipps
Lead Developer of Sacado
Developer of LOCA, NOX

Denis Ridzal
Lead Developer of Aristos and Intrepid

Marzio Sala
Lead Developer of Didasko and IFPACK
Developer of ML, Amesos

Andrew Salinger
Lead Developer of LOCA

Paul Sexton
Developer of Epetra and Tpetra

Bill Spotz
Lead Developer of PyTrilinos
Developer of Epetra, New_Package

Ken Stanley
Lead Developer of Amesos and New_Package

Heidi Thornquist
Lead Developer of Anasazi, Belos, RBGen, and Teuchos

Ray Tuminaro
Lead Developer of ML and Meros

Jim Willenbring
Developer of Epetra and New_Package.
Trilinos library manager

Alan Williams
Lead Developer of Isorropia
Developer of Epetra, EpetraExt, AztecOO, Tpetra

Sandia National Laboratories

Overview of Nonlinear Analysis, Sensitivity and Optimization Capabilities

Trilinos Nonlinear Solver and Analysis Packages

Trilinos Packages

- 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
  - ODE/DAE Sensitivities …

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

  MOOCHO

Sandia National Laboratories

Full Vertical Solver Coverage

Optimization **Unconstrained:** **Constrained:**	Find $u \in \Re^n$ that minimizes $g(u)$ Find $x \in \Re^m$ and $u \in \Re^n$ that minimizes $g(x,u)$ s.t. $f(x,u) = 0$	**Derivatives** **(Automatic Differentiation: Sacado)**	**MOOCHO**
Bifurcation Analysis	Given nonlinear operator $F(x,u) \in \Re^{n+m}$ For $F(x,u) = 0$ find space $u \in U \ni \frac{\partial F}{\partial x}$		**LOCA**
Transient 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 $x(t) \in \Re^n, t \in [0,T]$		**Rythmos**
Nonlinear Problems	Given nonlinear operator $F(x) \in \Re^m \to \Re$ Solve $F(x) = 0 \quad x \in \Re^n$		**NOX**
Linear Problems **Linear Equations:** **Eigen Problems:**	Given Linear Ops (Matrices) $A, B \in \Re^{m \times n}$ Solve $Ax = b$ for $x \in \Re^n$ Solve $A\nu = \lambda B \nu$ for (all) $\nu \in \Re^n, \quad \lambda \in$		**AztecOO** **Belos** **Ifpack, ML, etc...** **Anasazi**
Distributed Linear Algebra **Matrix/Graph Equations:** **Vector Problems:**	Compute $y = Ax; A = A(G); A \in \Re^{m \times n}, G \in \Im^{m \times n}$ Compute $y = \alpha x + \beta w; \alpha = \langle x, y \rangle; x, y \in \Re^n$		**Epetra** **Tpetra**

Rythmos

Suite of time integration (discretization) methods

- Includes: backward Euler, forward Euler, explicit Runge-Kutta, and implicit BDF at this time
- Preliminary implicit Runge-Kutta methods (no local error control, other limitations)
- Binary checkpoints and restarting (new in Trilinos 10.0)
- Native support for operator split methods
- Highly modular
- Forward sensitivity computations
- Adjoint sensitivities being developed

Developers: Todd Coffey, Roscoe Bartlett

Why Sensitivities and Optimization?

The Standard Steady-State Forward Simulation Problem

For a given set of input parameters $p \in \mathbf{R}^{n_p}$, solve the square state equations

$$f(x,p) = 0$$

for the state variables $x \in \mathbf{R}^{x_x}$ then compute observation(s) $g(x)$.

Example applications

- Discretized PDEs (e.g. finite element, finite volume, discontinous Gelerkin, finite difference, ...)
- Network problems (e.g. circuit simulation, power grids, ...)
- ...

Why is a forward solver is not enough?

- A forward solve $p \rightarrow g(x(p),p)$ can only give point-wise information, it can't tell you what you ultimately want to know:
  - How to a characterize the error in my model so that it can be improved? $\rightarrow$ Error estimation
  - What is the uncertainty in x or $g(x(p),p)$ given uncertainty in p ? $\rightarrow$ UQ
  - What is the "best" value of p so that my model $f(x,p)=0$ fits exp. data? $\rightarrow$ Param. Estimation
  - What is the "best" value for p to achieve some goal? $\rightarrow$ Optimization

What are some of the tools that we need to answer these higher questions?

- Sensitivities and Optimization!

Sandia National Laboratories

Derivatives and Sensitivity Computations

Steady-State Simulation-Constrained Sensitivities

Steady-State Simulation-Constrained Response

Compute $g(x,p) \in \mathbf{R}^{n_x} \times \mathbf{R}^{n_p} \rightarrow \mathbf{R}^{n_g}$

such that $f(x,p) = 0$ (where $f(x,p) \in \mathbf{R}^{n_x} \times \mathbf{R}^{n_p} \rightarrow \mathbf{R}^{n_x}$)

Nonlinear elimination: $f(x,p) = 0$ → $p \rightarrow x(p)$ → Reduced Response Function: $p \rightarrow g(x(p),p) \rightarrow \hat{g}(p)$

Steady-State Sensitivities

State Sensitivity:

$$\frac{\partial x}{\partial p} = -\frac{\partial f}{\partial x}^{-1}\frac{\partial f}{\partial p}$$

Well suited for Newton Methods

Reduced Response Function Sensitivity:

$$\frac{\partial \hat{g}}{\partial p} = \frac{\partial g}{\partial x}\frac{\partial x}{\partial p} + \frac{\partial g}{\partial p}$$

Forward (Direct) vs. Adjoint Sensitivities

Complexity

Forward (Direct) Sensitivity Method:

$$\frac{\partial \hat{g}}{\partial p} = \frac{\partial g}{\partial x}\left(-\frac{\partial f}{\partial x}^{-1}\frac{\partial f}{\partial p}\right) + \frac{\partial g}{\partial p}$$

O(n_p)

Adjoint Sensitivity Method:

$$\frac{\partial \hat{g}}{\partial p}^T = \frac{\partial f}{\partial p}^T \boxed{\left(-\frac{\partial f}{\partial x}^{-T}\frac{\partial g}{\partial x}^T\right)} + \frac{\partial g}{\partial p}^T$$

Adjoint variables λ

O(n_g)

Uses for Sensitivities: Derivative-based optimization, UQ, error estimation etc ...

Transient Sensitivities: State Equations with Responses

Fully Implicit ODE/DAE State Equations

$$\begin{aligned} f\left(\dot{x}(t), x(t), p, t\right) &= 0,\ t \in \left[t_0, t_f\right] \\ x(0) &= x_0(p) \\ \dot{x}(0) &= \dot{x}_0(p) \end{aligned}$$

Composite Response Function

$$d(x,p) = \int_{t_0}^{t_f} g(\dot{x}(t), x(t), p, t)dt + h(\dot{x}(t_f), x(t_f), p)$$

Reduced Composite Response Function

$$\widehat{d}(p,v) = \int_{t_0}^{t_f} \widehat{g}(p,t)dt + \widehat{h}(p)$$

where:

$$\widehat{g}(p,t) = g(\dot{x}(p,t), x(p,t), p, t))$$

$$\widehat{h}(p) = h(\dot{x}(p,t_f), x(p,t_f), p))$$

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

Sandia National Laboratories

Transient Sensitivities: Forward Sensitivity Method

Forward Sensitivity Equations:

$$\begin{aligned} \frac{\partial f}{\partial \dot{x}}\left(\frac{\partial \dot{x}}{\partial p}\right) + \frac{\partial f}{\partial x}\left(\frac{\partial x}{\partial p}\right) + \frac{\partial f}{\partial p} &= 0,\ t \in [t_0, t_f] \\ \frac{\partial x(t_0)}{\partial p} &= \frac{\partial x_0}{\partial p} \\ \frac{\partial \dot{x}(t_0)}{\partial p} &= \frac{\partial \dot{x}_0}{\partial p} \end{aligned}$$

Forward Reduced Gradient

$$\frac{\partial \hat{d}}{\partial p} = \int_{t_0}^{t_f} \left(\frac{\partial g}{\partial \dot{x}}\frac{\partial \dot{x}}{\partial p} + \frac{\partial g}{\partial x}\frac{\partial x}{\partial p} + \frac{\partial g}{\partial p} \right) dt + \left(\frac{\partial h}{\partial \dot{x}}\frac{\partial \dot{x}}{\partial p} + \frac{\partial h}{\partial x}\frac{\partial x}{\partial p} + \frac{\partial h}{\partial p} \right)\Bigg|_{t=t_f}$$

Forward Sensitivity Methods:

- The forward sensitivities d(x)/d(p) are integrated right along with the forward state equation
- Can be solved using explicit or implicti time integration mehtods
- $O(n_p)$ extra storage and computation per time step
- Reuse of forward solver integrator infrastructure, Jacobian/preconditioner storage

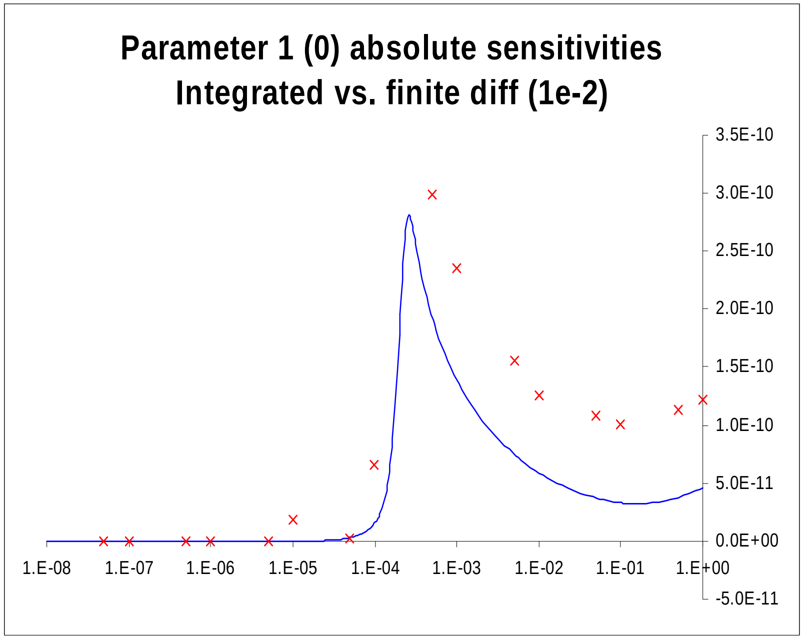
http://www.cs.sandia.gov/~rabartl/TransientSensitivitiesDerivation.pdf

Sandia National Laboratories

Forward Transient Sensitivities with Charon/Rythmos

QASPR transient current sensitivities w.r.t. reaction parameters for an irradiated semiconductor device modeled with Charon

- Embedded sensitivities with AD/Sacado (Phipps) & Rythmos
- Finite differences (steplen=1e-2) (optimal steplen=1e-1)
- Embedded sensitivities vs. finite diff.
  - Much more accurate and robust!
  - 10x faster for 40 parameters!



Bartlett, Roscoe, Scott Collis, Todd Coffey, David Day, Mike Heroux, Rob Hoekstra, Russell Hooper, Roger Pawlowski, Eric Phipps, Denis Ridzal, Andy Salinger, Heidi Thornquist, and Jim Willenbring. *ASC Vertical Integration Milestone*. SAND2007-5839, Sandia National Laboratories, 2007 [http://www.cs.sandia.gov/~rabartl/publications.html]

Optimization

Steady-State Simulation-Constrained Optimization

Basic Steady-State Simulation-Constrained Optimization Problem:

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

Trilinos Optimization Packages

- MOOCHO (R. Bartlett)
- Aristos (D. Ridzal)

Basic example optimization formations

- Parameter estimation / data reconciliation
- Optimal design
- Optimal control
- ...

Define Lagrangian: $L(x,p,\lambda) = g(x,p) + \lambda^T f(x,p)$

First-order necessary optimality conditions

State equation: $$\frac{\partial L}{\partial \lambda}^T = f(x,p) = 0$$

Adjoint equation: $$\frac{\partial L}{\partial x}^T = \frac{\partial g}{\partial x}^T + \frac{\partial f}{\partial x}^T \lambda = 0$$

Gradient equation: $$\frac{\partial L}{\partial p}^T = \frac{\partial g}{\partial p}^T + \frac{\partial f}{\partial p}^T \lambda = 0$$

$$\frac{\partial \hat{g}}{\partial p}^T = -\frac{\partial f}{\partial p}^T \frac{\partial f}{\partial x}^{-T} \frac{\partial g}{\partial x}^T + \frac{\partial g}{\partial p}^T = 0$$

Reduced gradient!

Sandia National Laboratories

Simulation-Constrained Optimization Methods

Basic Steady-State Simulation-Constrained Optimization Problem:

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

Two broad approaches for solving optimization problems

- Non-invasive (decoupled) approach (simulation constraints always satisfied): **DAKOTA**

Find $p \in \mathbf{R}^{n_p}$ that:
minimizes $\hat{g}(p) = g(x(p),p)$

Optimization method never "sees" the state space!

- Embedded (coupled) approach (converges optimality and feasibility together): **MOOCHO**

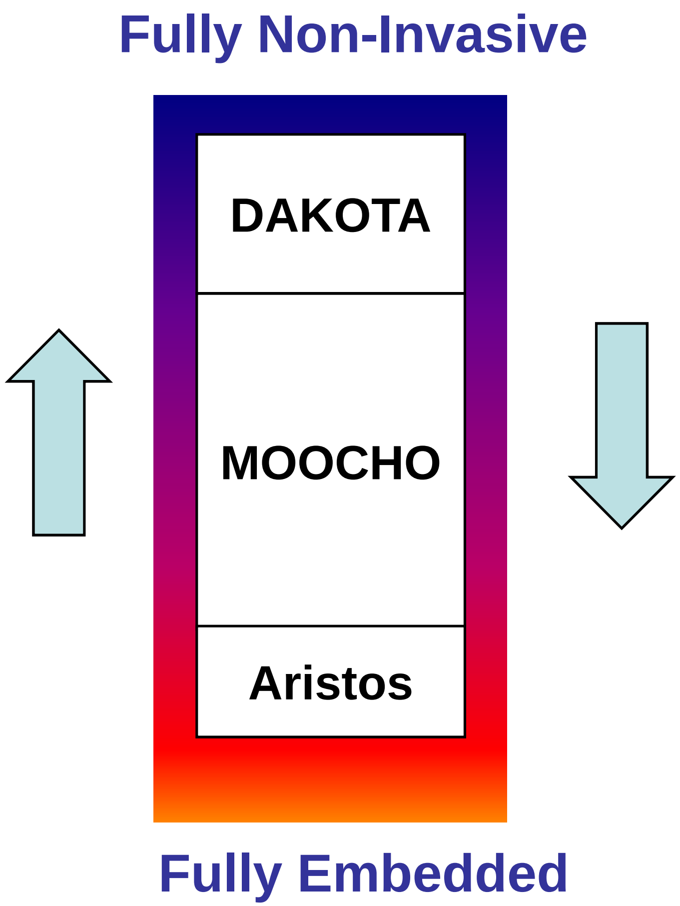
Find $x \in \mathbf{R}^{n_x}$ and $p \in \mathbf{R}^{n_p}$ that:
minimizes $g(x,p)$
such that $f(x,p) = 0$

- Optimization method deals with the (parallel) state space and the parameter space together!
- Requires special globalization methods to converge to a minimum!

Sandia National Laboratories

A Spectrum of Optimization Methods form Decoupled to Coupled

- Decreased impact to existing app code
- Ease of interfacing



- Better scalability to large parameter spaces
- More accurate solutions
- Less computer time

Sandia National Laboratories

Scalable Optimization Test Problem

Example: Parallel, Finite-Element, 2D, Diffusion + Reaction (GL) Model

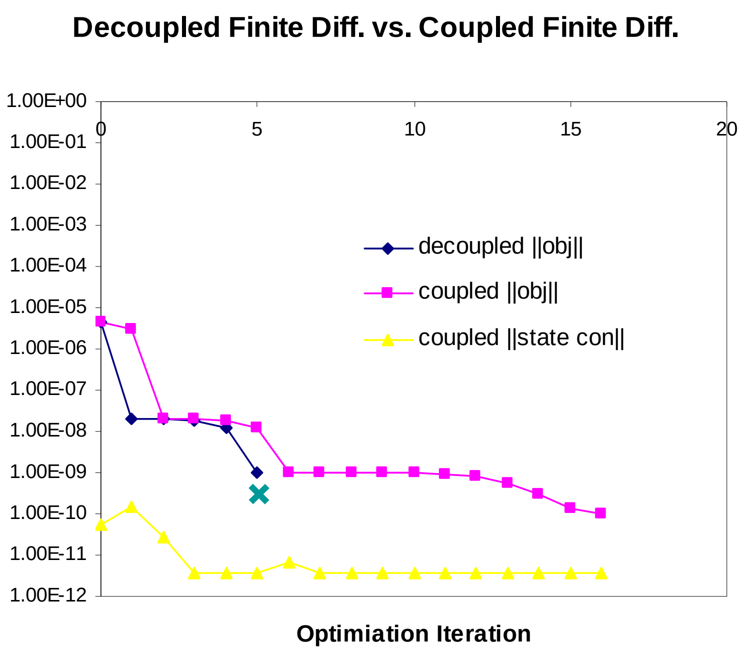
$$\begin{aligned} \min \quad & \tfrac{1}{2}\int_{\Omega}(x(y)-x^*(y))^2 dy & \\ \text{s.t.} \quad & \nabla^2 x + \alpha(x - x^3) = r(y) & y \in \Omega \\ & \frac{\partial x(y)}{\partial n} = q(p,y) & y \in \partial\Omega \end{aligned} \quad \Rightarrow \quad \begin{aligned} \min \quad & g(x,p) \\ \text{s.t.} \quad & f(x,p) = 0 \end{aligned}$$

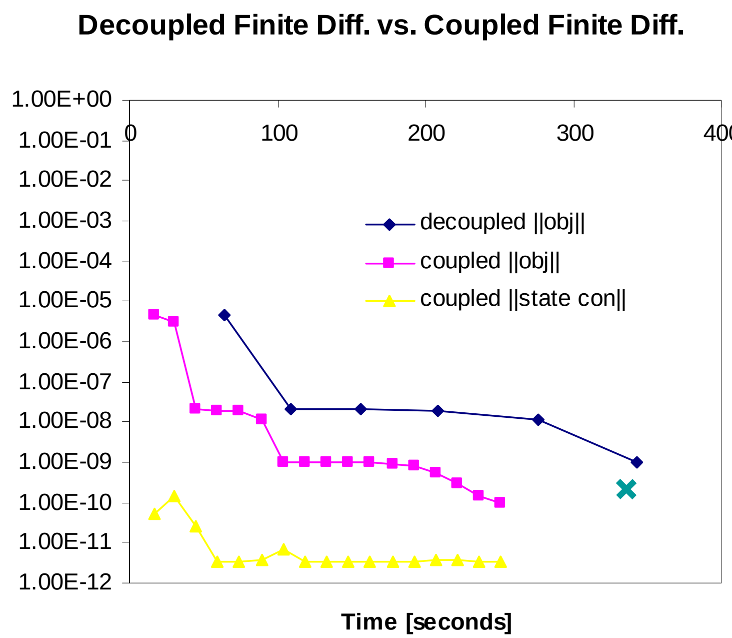
- State PDE: Scalar Ginzburg-Landau equations (based on Denis Ridzal's Ph.D. code)
- Discretization:
  - Second-oder FE on triangles
  - n_x = 110,011 state variables and equations
- Optimization variables:
  - Sine series basis
  - n_p = 8 optimization variables
  - Note: df/dp is constant in this problem!!!
- Iterative Linear Solver : ILU (Ifpack), (GMRES) AztecOO

Key Points

- Simple physics but leads to very nonlinear state equations
- Inverse optimization problem is very ill posed in many instances

Results: Decoupled vs. Coupled, Finite Differences





Key Points

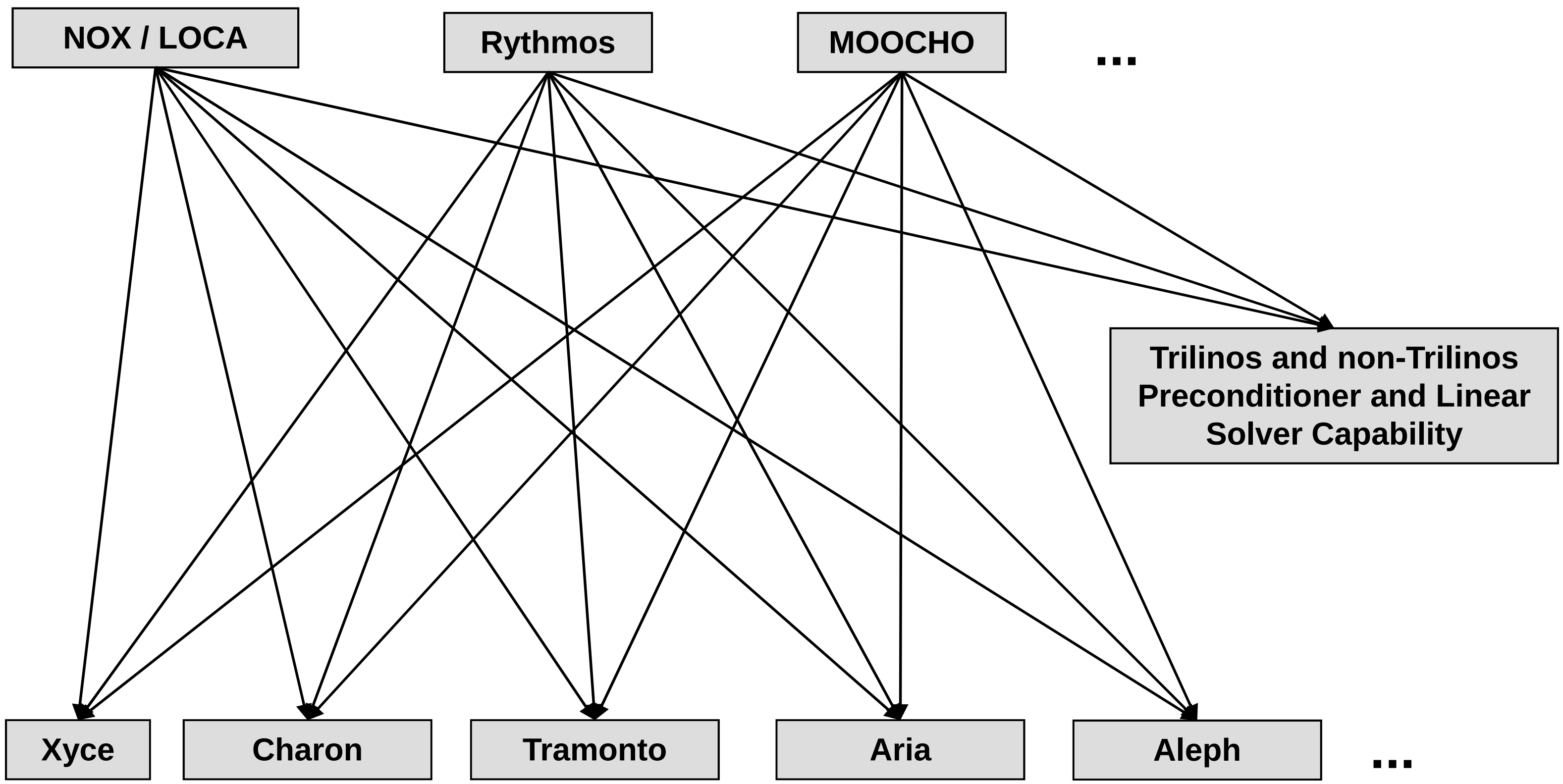
- Finite differencing the underlying functions is much more efficient than finite differencing entire simulation!
- Finite differencing the underlying functions is more accurate!
- Coupled approach requires (almost) no extra application requirements!

Sandia National Laboratories

ModelEvaluator Overview

Nonlinear Algorithms and Applications : Everyone for Themselves?

Nonlinear
ANA Solvers
in Trilinos



Sandia
Applications

Key Point

- BAD

Overview of Nonlinear Model Evaluator Interface

Motivation: An interface for nonlinear problems is needed that will support a variety of different types of problems

- Nonlinear equations (and senstitivities)
- Stability analysis and continuation
- Explicit ODEs (and sensitivities)
- DAEs and implicit ODEs (and sensitivities)
- Unconstrained optimization
- Constrained optimization
- Uncertainty quantification
- …

as well as different combinations of problem types such as:

- Uncertainty in transient simulations
- Stability of an optimum under uncertainty of a transient problem

Key Point

The number of combinations of different problem types is large and trying to statically type all of the combinations is not realistic

Approach: Develop a single, scalable interface to address all of these problems

- (Some) Input arguments:
  - State and differential state: $x \in \mathcal{X}$ and $\dot{x} = \frac{dx}{dt} \in \mathcal{X}$
  - Parameter sub-vectors: $p_l \in \mathcal{P}_l$ for $l = 0 \ldots N_p - 1$
  - Time (differential): $t \in \mathbf{R}$
- (Some) Output functions:
  - State function: $(\dot{x}, x, \{p_l\}, t) \Rightarrow f \in \mathcal{F}$
  - Auxiliary response functions: $(\dot{x}, x, \{p_l\}, t) \Rightarrow g_j \in \mathcal{G}_j$, for $j = 0 \ldots N_g - 1$
  - State/state derivative operator (LinearOpWithSolve): $(\dot{x}, x, \{p_l\}, t) \Rightarrow W = \alpha \frac{\partial f}{\partial \dot{x}} + \beta \frac{\partial f}{\partial x}$

Key Point

All inputs and outputs are optional and the model evaluator object itself decides which ones are accepted.

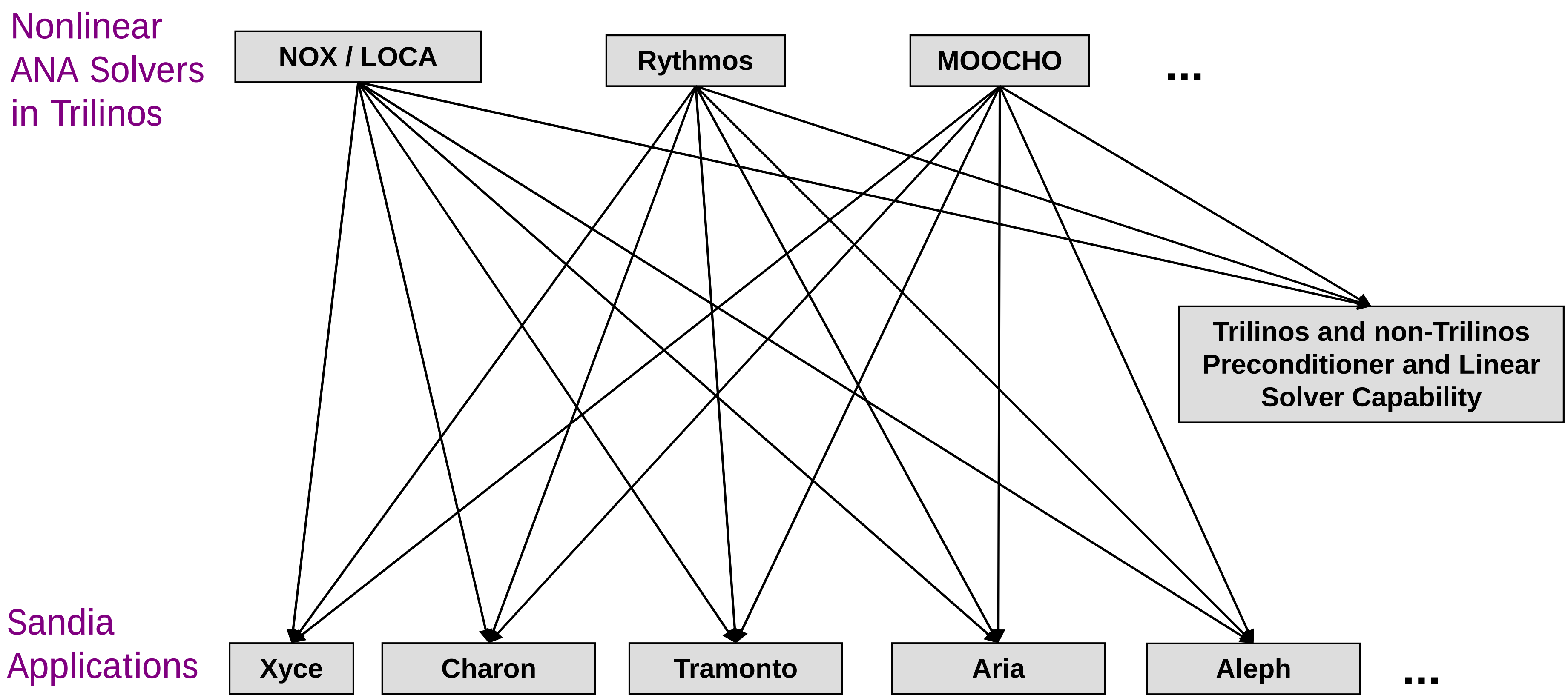
Sandia National Laboratories

Some Nonlinear Problems Supported by the ModelEvaluator

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
Explicit ODEs:	Solve $\dot{x} = f(x,t) = 0, t \in [0,T]$, $x(0) = x_0$, for $x(t) \in \mathbf{R}^n, t \in [0,T]$
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 $x(t) \in \mathbf{R}^n, t \in [0,T]$
Explicit ODE Forward Sensitivities:	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 $x(t) \in \mathbf{R}^n, t \in [0,T]$
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 $x(t) \in \mathbf{R}^n, t \in [0,T]$
Unconstrained Optimization:	Find $p \in \mathbf{R}^m$ that minimizes $g(p)$
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

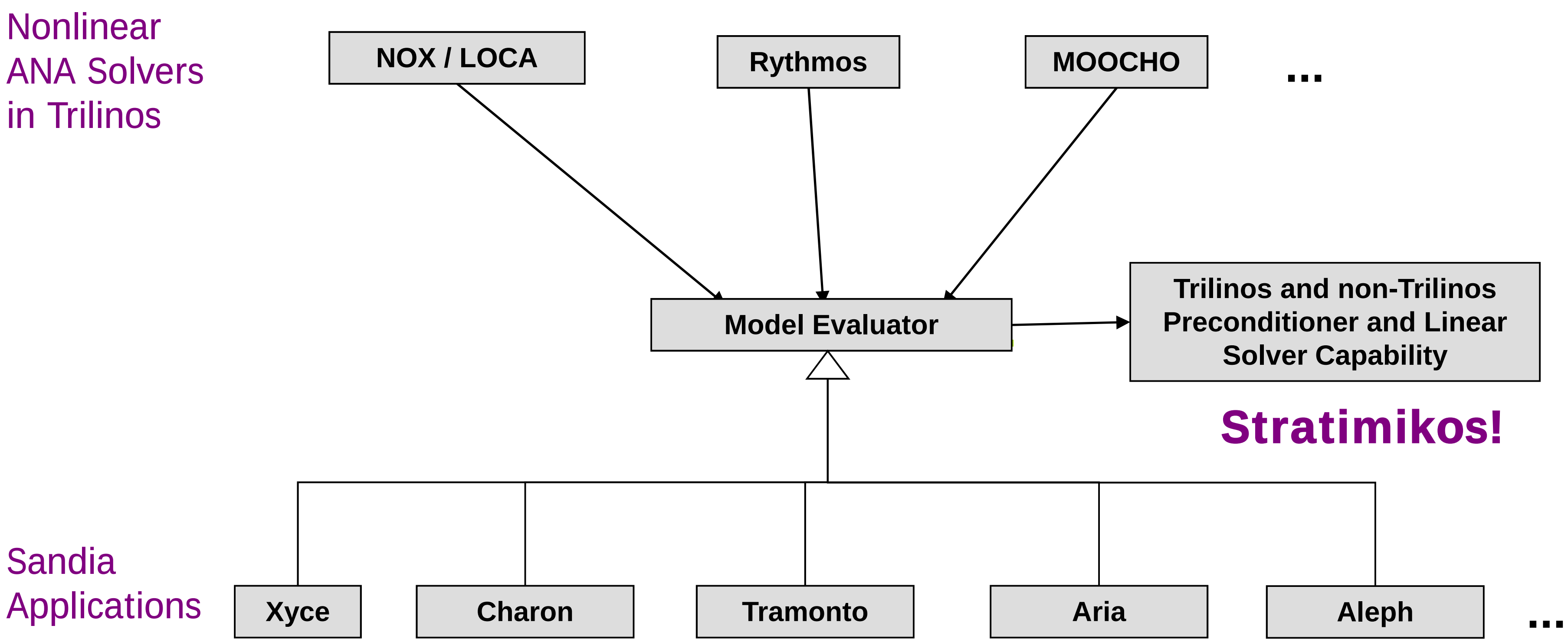
Nonlinear Algorithms and Applications : Everyone for Themselves?



Key Point

- BAD

Nonlinear Algorithms and Applications : Thyra & Model Evaluator!



Key Points

- Provide single interface from nonlinear ANAs to applications
- Provide single interface for applications to implement to access nonlinear ANAs
- Provides shared, uniform access to linear solver capabilities
- Once an application implements support for one ANA, support for other ANAs can quickly follow

Trilinos “Skins” and Interoperability with Other Software

Trilinos "Skins"

PyTrilinos provides Python access to Trilinos packages

- **Uses SWIG to generate bindings.**
- **Epetra, AztecOO, IFPACK, ML, NOX, LOCA, Amesos and NewPackage are supported.**
- **ModelEvaluator wrapper is being developed**
- **Developers: Bill Spotz**

CTrilinos provides C and Fortran 77 compatible wrappers to Trilinos

- **Currently wraps just part of Epetra**
- **More wrappers to come => ModelEvaluator …**
- **Provides basic C++/Fortran interoperability for ForTrilinos interfaces**
- **Developers: ???**

ForTrilinos developing full object-oriented interfaces to Trilinos

- **Based on basic wrappers in CTrilinos**
- **Uses new OO features of Fortran 2003**
- **Developers: Damian Rousan**
- **Not in Trilinos 10.0**

Sandia National Laboratories

Trilinos / PETSc Interoperability

- **Epetra_PETScAIJMatrix class**
  - Derives from Epetra_RowMatrix
  - Wrapper for serial/parallel PETSc aij matrices
  - Utilizes callbacks for matrix-vector product, getrow
  - No deep copies

- **Enables PETSc application to construct and call virtually any Trilinos preconditioner**

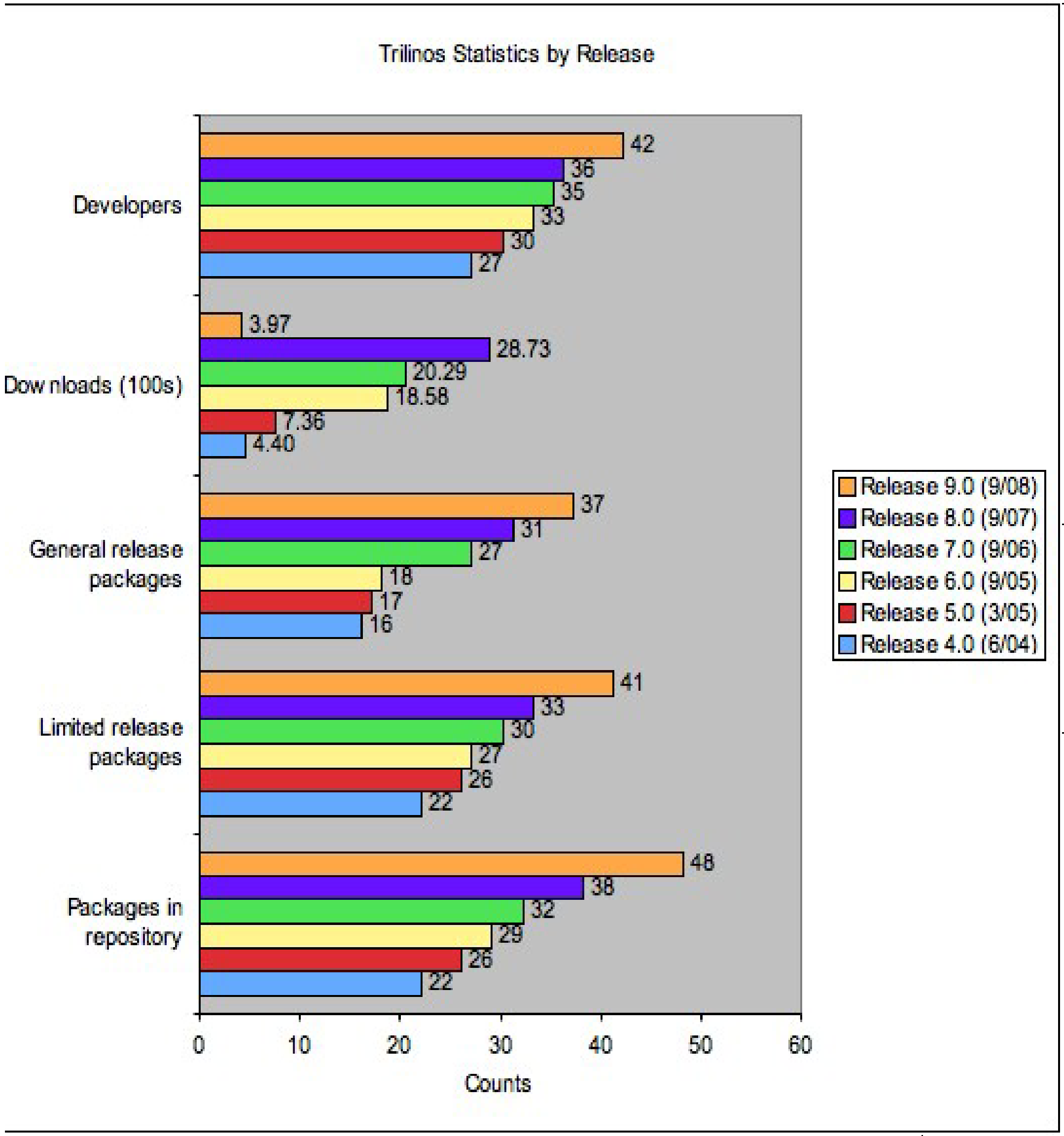
- **ML accepts fully constructed PETSc KSP solvers as smoothers**
  - Fine grid only
  - Assumes fine grid matrix is really PETSc aij matrix

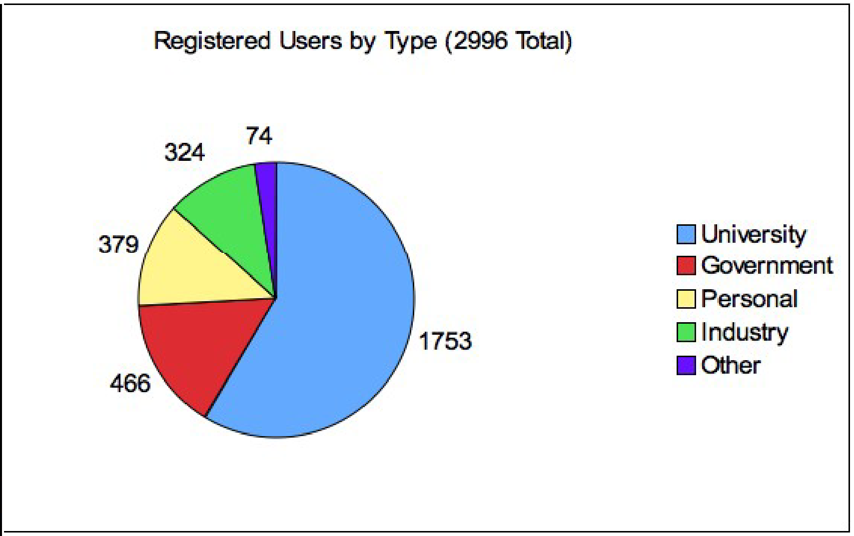
- **Complements Epetra_PETScAIJMatrix class**
  - For any smoother with getrow kernel, PETSc implementation should be *much* faster than Trilinos
  - For any smoother with matrix-vector product kernel, PETSc and Trilinos implementations should be comparable

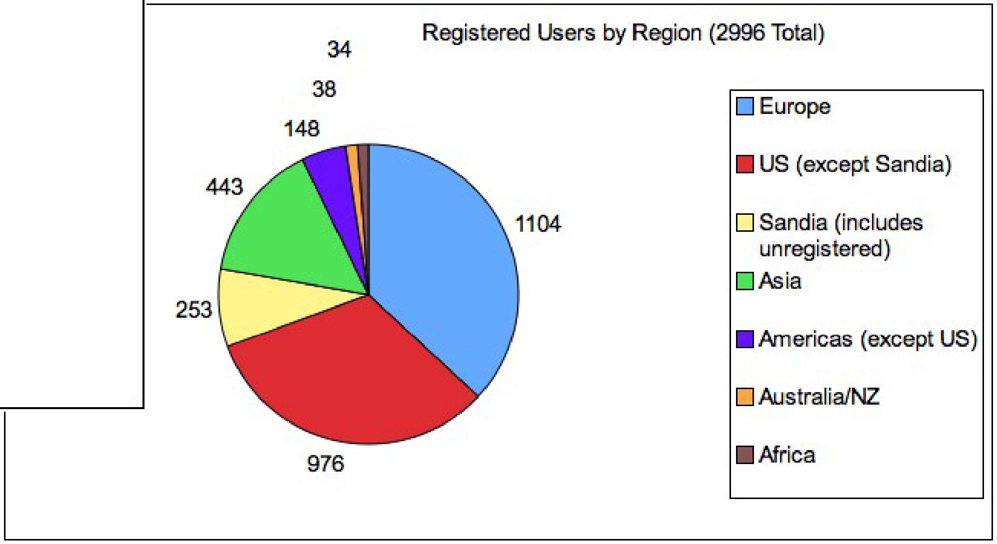
Sandia National Laboratories

Summary Wrap-up

Trilinos Statistics







Stats: Trilinos Download Page 10/20/2008.

External Visibility

- Awards: R&D 100, HPC SW Challenge (04).
- www.cfd-online.com:

> Trilinos ☺
> A project led by Sandia to develop an object-oriented software framework for scientific computations. This is an active project which includes several state-of-the-art solvers and lots of other nice things a software engineer writing CFD codes would find useful. Everything is freely available for download once you have registered. Very good!

- Industry Collaborations: Boeing, Goodyear, ExxonMobil, others.
- Linux distros: Debian, Mandriva, Ubuntu, Fedora.
- SciDAC TOPS-2 partner, IAA Algorithms (with ORNL).
- Over 8000 downloads since March 2005.
- Occasional unsolicited external endorsements such as the following two-person exchange on mathforum.org:

  > The consensus seems to be that OO has little, if anything, to offer
  > (except bloat) to numerical computing.
  I would completely disagree. A good example of using OO in numerics is
  Trilinos: http://software.sandia.gov/trilinos/

Trilinos Availability / Information

- Trilinos and related packages are available via LGPL.
- Current release (9.0) is "click release". Unlimited availability.
- Trilinos Release 10.0: September 2009.
  - CMake is now the supported build system
  - Autotools is no longer supported
  - Alpha release currently available to try CMake build system
- Trilinos Awards:
  - 2004 R&D 100 Award.
  - SC2004 HPC Software Challenge Award.
  - Sandia Team Employee Recognition Award.
  - Lockheed-Martin Nova Award Nominee.
- More information:
  - http://trilinos.sandia.gov
- 6th Annual Trilinos User Group Meeting in October 2008 @ SNL
  - talks available for download
- 7th Annual Trilinos User Group Meeting November 3-5, 2009

Sandia National Laboratories

Dependencies and Reusability

Using externally developed software can be as risk!

- **External software can be hard to learn**
- **External software may not do what you need**
- **Upgrades of external software can be risky:**
  - **Breaks in backward compatibility?**
  - **Regressions in capability?**
- **External software may not be well supported**
- **External software may not be support over long term**

What can reduce the risk of depending on external software?

- **Strong software engineering skill and processes (high quality, low defects, frequent releases)**
- **Strong organizational relationships**
- **Regulated backward compatibility and smooth upgrading**
- **Long term commitment (i.e. 10-30 years) to actively support the software**

Trilinos leaders and stakeholders recognize these issues and are committed to continual improvement!

Sandia National Laboratories

Useful Links

Trilinos website: **http://trilinos.sandia.gov**

Trilinos tutorial: **http://trilinos.sandia.gov/Trilinos8.0Tutorial.pdf**

Trilinos mailing lists: **http://trilinos.sandia.gov/mail_lists.html**

Trilinos User Group (TUG) meetings:
http://trilinos.sandia.gov/events/trilinos_user_group_2008
http://trilinos.sandia.gov/events/trilinos_user_group_2007