High-Performance Numerical Tools for the Development and Scalability of High-End Computer Applications

An Overview of The DOE Advanced CompuTational Software Collection (ACTS)

Annual Conference
DOE Computational Science Graduate Fellowship
Washington DC, July, 25, 2012

Tony Drummond (LADrummond@lbl.gov)
Computational Research Division
Lawrence Berkeley National Laboratory

acts-support@nersc.gov

U.S. DEPARTMENT OF ENERGY | Office of Science

ACTS COLLECTION

BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

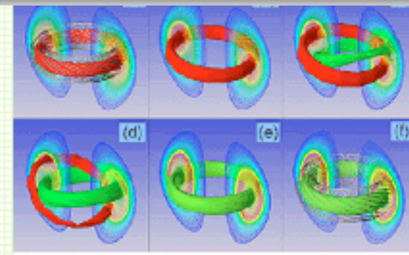
UNIVERSITY OF OREGON

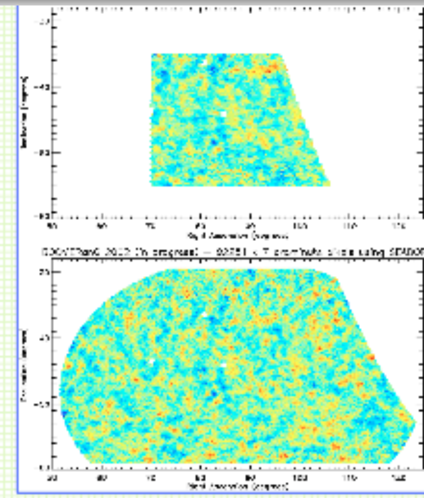
OUTLINE

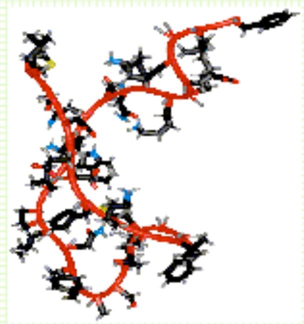
- Introduction to DOE ACTS Collection
- ACTS Numerical Functionality
- Use of the Tools in the ACTS Collection
- Sustainability through emerging hardware
- Summary

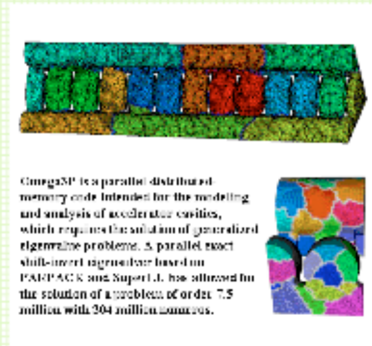
Motivation - HPC Applications

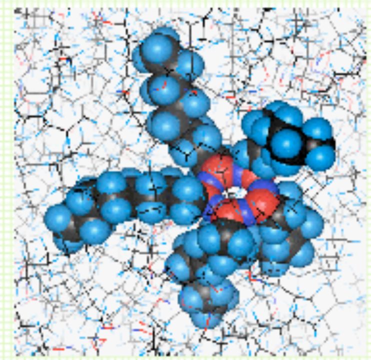
- Accelerator Science
- Astrophysics
- Biology
- Chemistry
- Earth Sciences
- Materials Science
- Nanoscience
- Plasma Science
-
-







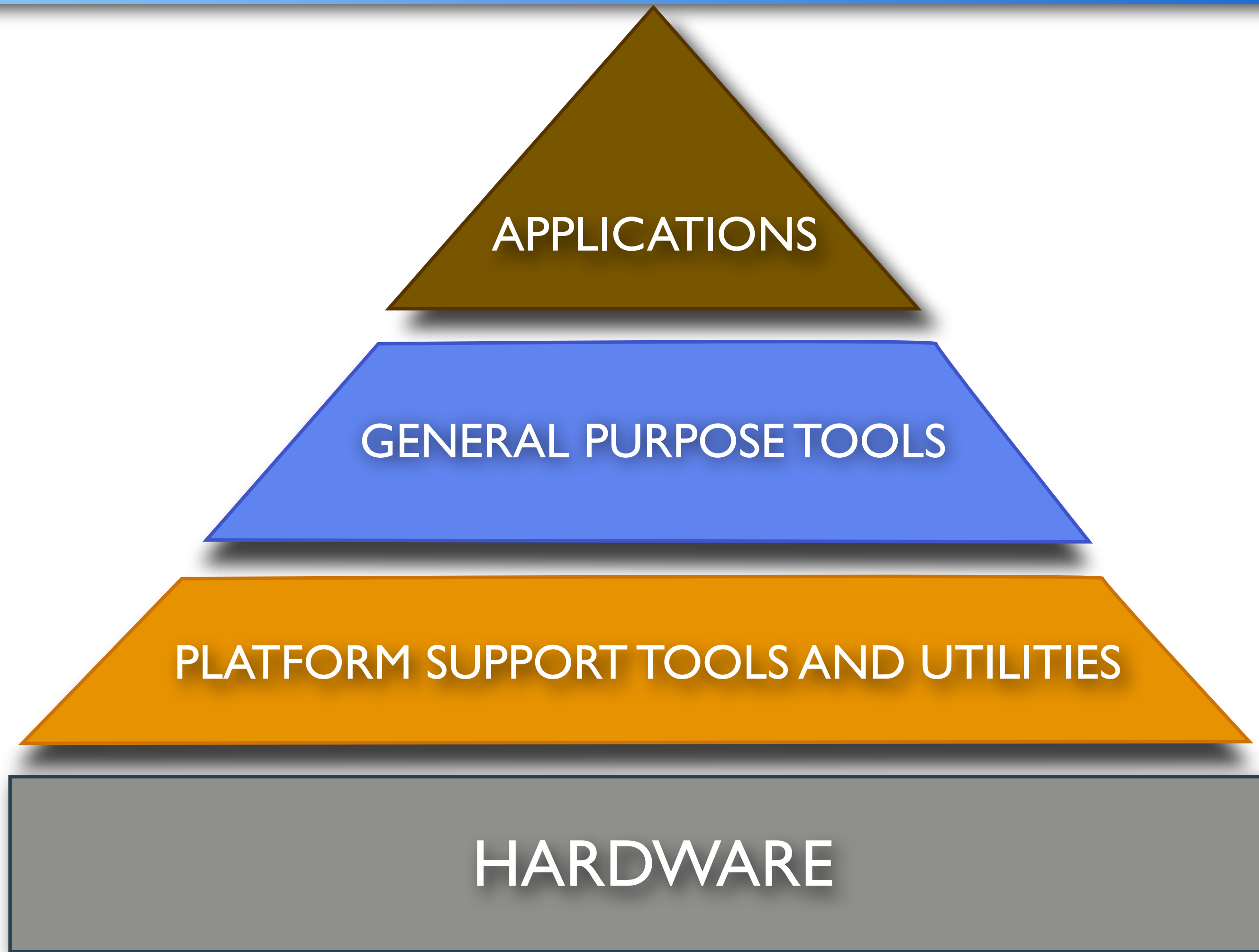




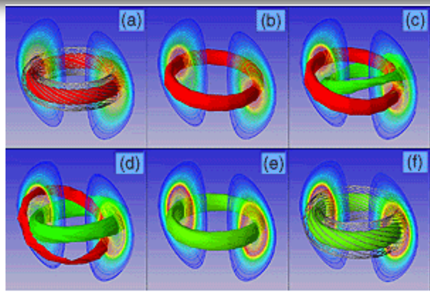
Commonalities:

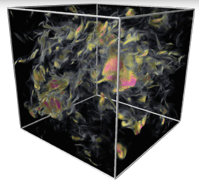
- Major advancements in Science
- Increasing demands for computational power
- Rely on available computational systems, languages, and software tools

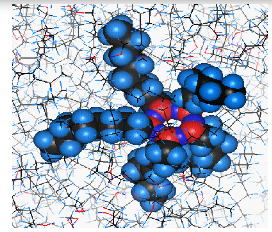
HPC Software Stack

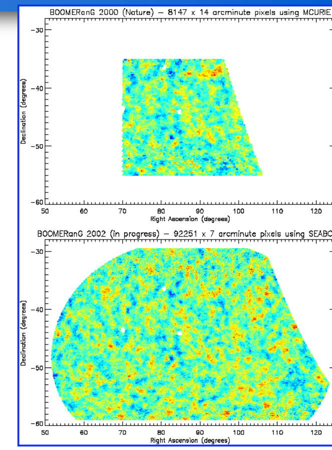


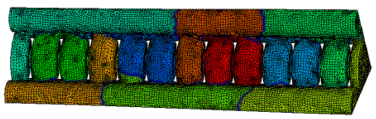
HPC Software Stack











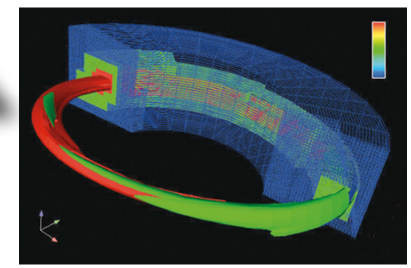
Omega3P is a parallel distributed-memory code intended for the modeling and analysis of accelerator cavities, which requires the solution of generalized eigenvalue problems. A parallel exact shift-invert eigensolver based on PARPACK and SuperLU has allowed for the solution of a problem of order 7.5 million with 304 million nonzeros.

APPLICATIONS

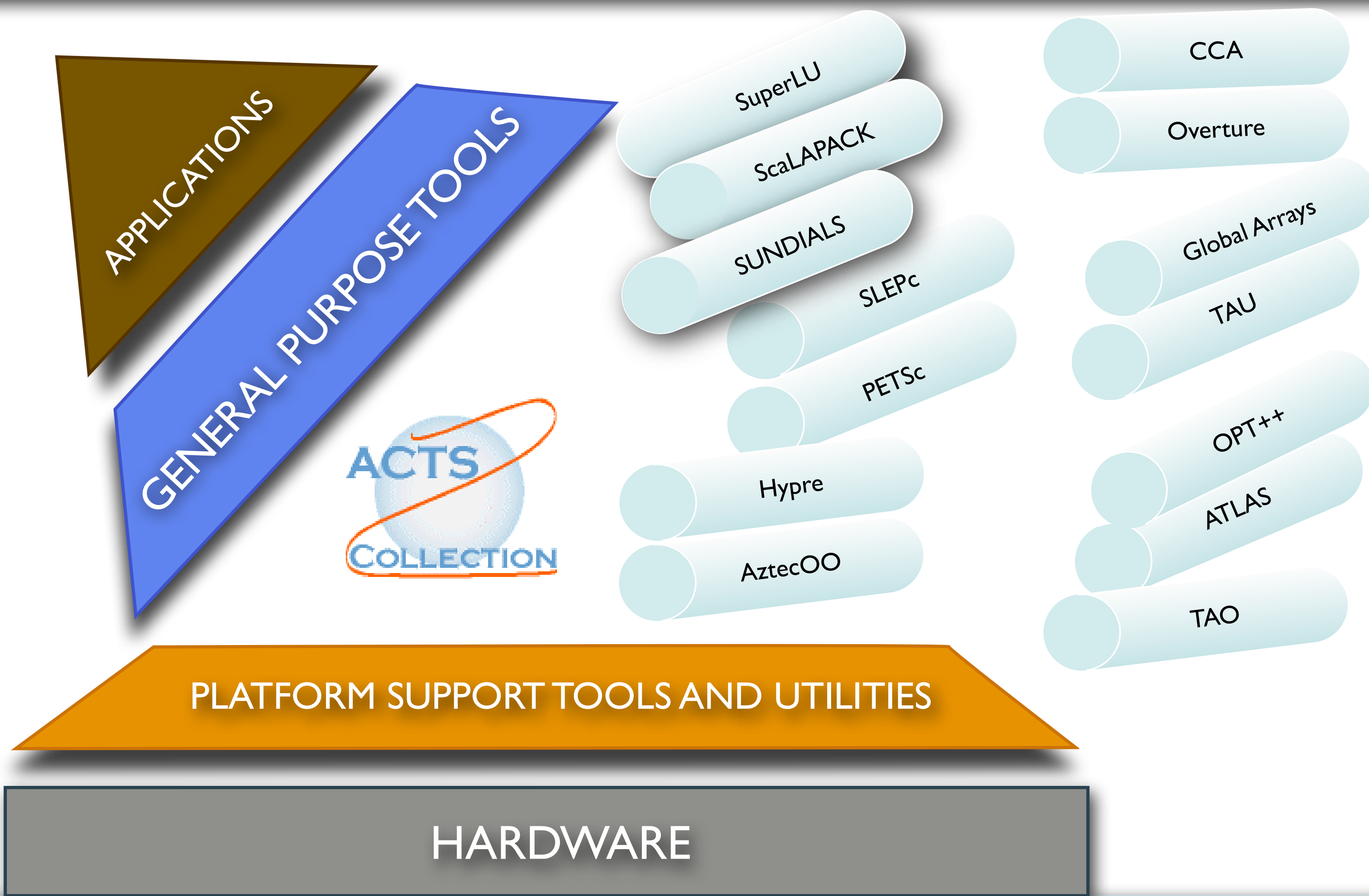
GENERAL PURPOSE TOOLS

PLATFORM SUPPORT TOOLS AND UTILITIES

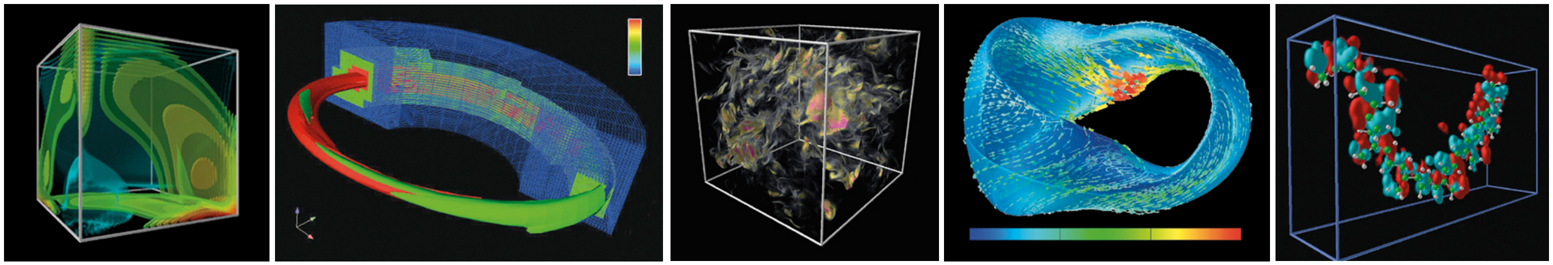
HARDWARE



HPC Software Stack



The DOE ACTS Collection

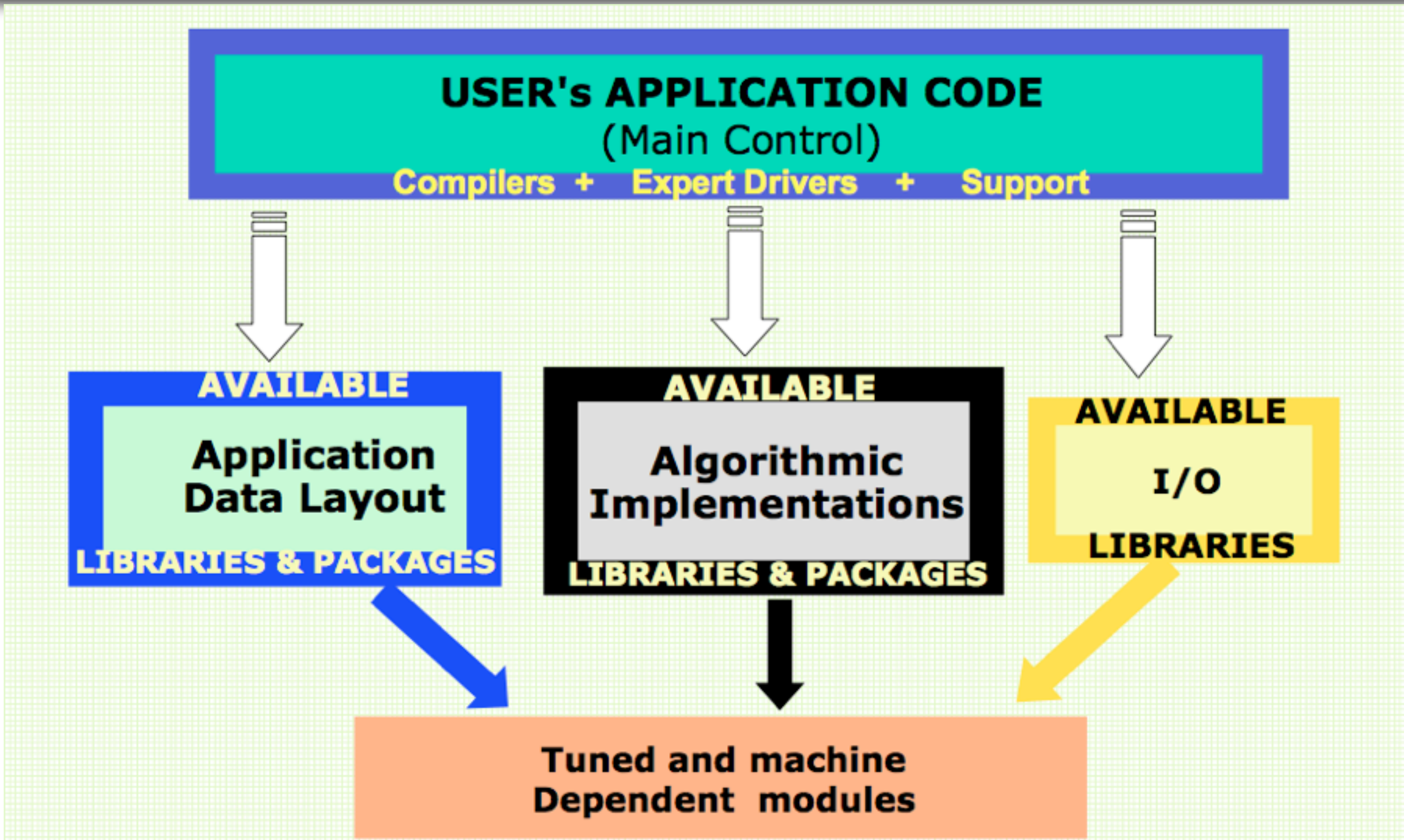


Goal: The Advanced CompuTational Software Collection (ACTS) makes reliable and efficient software tools more widely used, and more effective in solving the nation's engineering and scientific problems.

References:

- L.A. Drummond, O. Marques: An Overview of the Advanced CompuTational Software (ACTS) Collection. ACM Transactions on Mathematical Software Vol. 31 pp. 282-301, 2005
- http://acts.nersc.gov

Speeding-Up Software Development



Current State of DOE ACTS Collection

Category	Tool	Functionalities
Numerical	AztecOO	Scalable linear and non-linear solvers using iterative schemes.
	Hypre	A family of scalable preconditioners.
	PETSc	Scalable linear and non-linear solvers and additional support for PDE related work.
	SUNDIALS	Solvers for the solution of systems of ordinary differential equations, nonlinear algebraic equations, and differential-algebraic equations.
	ScaLAPACK	High performance parallel dense linear algebra.
	SLEPc	Scalable algorithms for the solution of large sparse eigenvalue problems.
	SuperLU	Scalable direct solution of large, sparse, nonsymmetric linear systems of equations.
	TAO	Large-scale optimization software.
Code Development	Global Arrays	Supports the development of parallel programs.
	Overture	Supports the development of computational fluid dynamics codes in complex geometries.
Run Time Support	TAU	Portable and scalable performance analyzes and tracing tools for C, C++, Fortran and Java programs.
Library Development	ATLAS	Automatic generation of optimized numerical dense algebra for scalar processors.

Category	Tool	Functionalities
Tools in Consideration	ML	Multilevel Preconditioners from Trilinos
	BELOS	Krylov based solvers from Trilinos
	Zoltan	Parallel Partitional, Data-Management and Load Balancing
	pOSKI	Sparse auto-tuning library

Considering 4 more

User Interfaces

```
CALL BLACS_GET( -1, 0, ICTXT )
 CALL BLACS_GRIDINIT( ICTXT, 'Row-major', NPROW, NPCOL )
:
 CALL BLACS_GRIDINFO( ICTXT, NPROW, NPCOL, MYROW, MYCOL )
:
:
 CALL PDGESV( N, NRHS, A, IA, JA, DESCA, IPIV, B, IB, JB, DESCB,
 $              INFO )
```

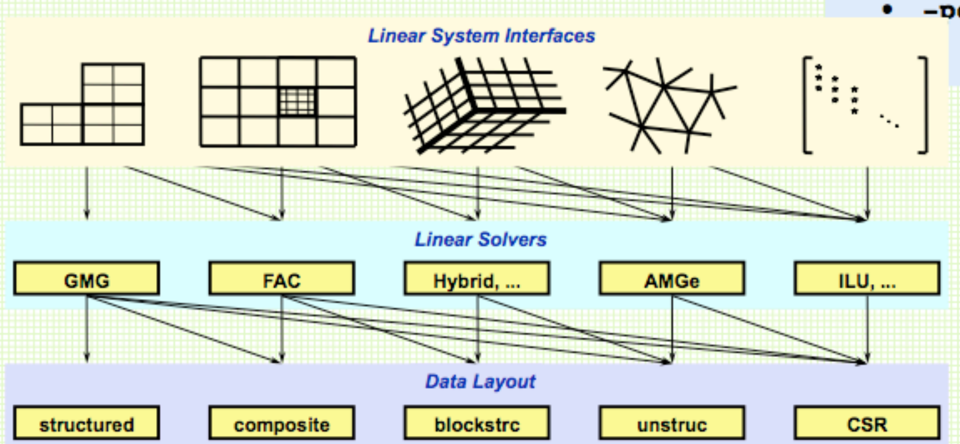
Library Calls

- `-ksp_type  [cg,gmres,bcgs,tfqmr,…]`
- `-pc_type  [lu,ilu,jacobi,sor,asm,…]`

More advanced:

- `-ksp_max_it  <max_iters>`
- `-ksp_gmres_restart  <restart>`
- `-pc_asm_overlap  <overlap>`

Command lines



Problem Domain

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations	Direct Methods	LU Factorization	ScaLAPACK(dense) SuperLU (sparse)
		Cholesky Factorization	ScaLAPACK
		LDL^T (Tridiagonal matrices)	ScaLAPACK
		QR Factorization	ScaLAPACK
		QR with column pivoting	ScaLAPACK
		LQ factorization	ScaLAPACK

Linear Solvers

- Solution of systems of linear equations may seem easy, but is at the heart of many computational problems.

- The choice of solution methods depends on matrix characteristics.
  - Symmetric vs nonsymmetric
  - Positive definite vs indefinite
  - Dimension
  - Sparsity
  - Special structures
    - Banded; block bordered diagonal
  - Conditioning

Linear Solvers

- Primary two favors of linear solvers:
  - Direct
  - Iterative
- Prefer to think of them as methods at opposite ends of a spectrum of linear solvers.
  - Preconditioned iterative methods are somewhere in between.
  - Where they are in the spectrum depends on the choice of preconditioners, including preconditioners constructed using techniques from sparse direct methods.

Comparison Between Direct and Iterative Solvers

- Direct
  - Finite no. of ops
    - Doesn't depend on anything
  - Pivoting may be needed to maintain stability
  - Large memory requirement
  - Complex data structure
    - Banded structure need not be optimal
  - Harder to implement
  - More communication
  - More graph problems
    - Ordering, symbolic manipulation
  - Easy to handle multiple RHS

- Iterative
  - Unknown no. of ops
    - Depend on no. of iterations
  - Preconditioning may be needed to improve convergence
  - Low memory requirement
  - Simple data structure
  - Easier to implement
  - Less communication
  - Fewer graph problems
  - Handling multiple RHS may not be easy

Comparison Between Direct and Iterative Solvers

- Direct methods or iterative methods?
  - Depend on dimensions, sparsity, and conditioning
  - Sparse direct solvers have become very efficient.
    - Almost all sparse direct solvers are built on top of dense matrix operations.
  - Direct methods are desirable when
    - Poor conditioning
    - High accuracies are desired
    - Small dimensions … How small is "small"?
      - Really depend on memory requirement and time to solution
    - Solving multiple linear systems with the same matrix
      - Only one factorization required

Functionality in The DOE ACTS Collection

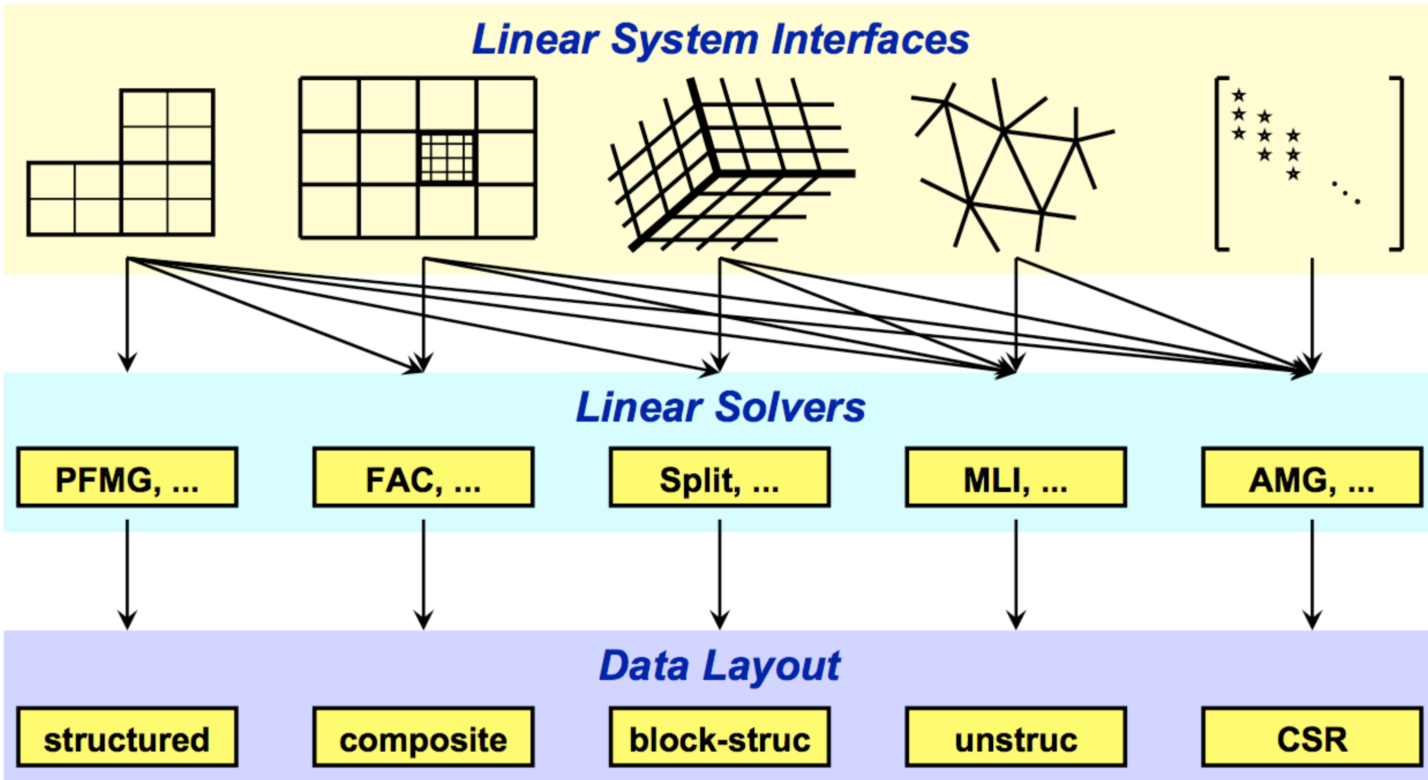
Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations	Direct Methods	LU Factorization	ScaLAPACK(dense) SuperLU (sparse)
		Cholesky Factorization	ScaLAPACK
		LDL^T (Tridiagonal matrices)	ScaLAPACK
		QR Factorization	ScaLAPACK
		QR with column pivoting	ScaLAPACK
		LQ factorization	ScaLAPACK

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations *(cont..)*	Iterative Methods	Conjugate Gradient	AztecOO (Trilinos) PETSc
		GMRES	AztecOO PETSc Hypre
		CG Squared	AztecOO PETSc
		Bi-CG Stab	AztecOO PETSc
		Quasi-Minimal Residual (QMR)	AztecOO
		Transpose Free QMR	AztecOO PETSc

Hypre

Inside Hypre (developed at LLNL)



- Algorithmic Implementations of Numerical Schemes
- Optimized platform support tools and libraries

Hypre use of Conceptual Interfaces

Inside Hypre

Solvers	System Interfaces			
	Struct	SStruct	FEI	IJ
Jacobi	✓	✓		
SMG	✓	✓		
PFMG	✓	✓		
Split		✓		
SysPFMG		✓		
FAC		✓		
Maxwell		✓		
AMS		✓	✓	✓
BoomerAMG		✓	✓	✓
MLI		✓	✓	✓
ParaSails		✓	✓	✓
Euclid		✓	✓	✓
PILUT		✓	✓	✓
PCG	✓	✓	✓	✓
GMRES	✓	✓	✓	✓
BiCGSTAB	✓	✓	✓	✓
Hybrid	✓	✓	✓	✓

Trilinos Framework

Full Vertical Solver Coverage

Trilinos

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. $f(x,u)=0$	Sensitivities (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 $F(x) \in \Re^m \rightarrow \Re$ 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

Trilinos Linear Solvers

Packages	Linear Solvers/Preconditioners
Amesos	• Direct sparse linear solvers
AztecOO	• Krylov based iterative linear solvers • ILU-type methods
Belos	• Krylov based iterative linear solvers
CLAPS	• Domain decomposition methods
Epetra	• Direct dense linear solver
IFPACK/TIFPACK	• Algebraic preconditioners • ILU type methods
Komplex	• Krylov based iterative linear solvers
Meros	• Block preconditioners
ML	• Multigrid methods
Teko	• Block preconditioners

Trilinos interoperability

Library	Functionality
SuperLU	• Direct sparse linear solvers
MUMPS	• Direct sparse linear solvers
PETSc	• Epetra_PETScAIJMatrix • ML accepts PETSc KSP for smoothers (fine grid only)
⋮	

PETSc's Algorithmic Functionalities

Nonlinear Solvers

Newton-based Methods		Other
Line Search	Trust Region	

Time Steppers

Euler	Backward Euler	Pseudo Time Stepping	Other

Krylov Subspace Methods

GMRES	CG	CGS	Bi-CG-STAB	TFQMR	Richardson	Chebychev	Other

Preconditioners

Additive Schwartz	Block Jacobi	Jacobi	ILU	ICC	LU (Sequential only)	Others

Matrices

Compressed Sparse Row (AIJ)	Blocked Compressed Sparse Row (BAIJ)	Block Diagonal (BDIAG)	Dense	Matrix-free	Other

Distributed Arrays

Index Sets

Indices	Block Indices	Stride	Other

Vectors

PETSc KSP Interface

- PETSc Linear System Solver Interface (KSP)
- Solve: Ax=b,
- Based on the Krylov subspace methods with the use of a preconditioning technique to accelerate the convergence rate of the numerical scheme.

KRYLOV SUBSPACE METHODS + PRECONDITIONERS
R. Freund, G. H. Golub, and N. Nachtigal. *Iterative Solution of Linear Systems,pp 57-100. ACTA Numerica. Cambridge University Press, 1992.*

$$(M_L^{-1} A M_R^{-1})(M_R x) = M_L^{-1} b,$$

- For left and right preconditioning matrices, M_L and M_R, respectively

For $M_R = I$

$$r_L \equiv M_L^{-1} b - M_L^{-1} Ax = M_L^{-1} r$$

PETSC Default

PETSc Interoperability with Other Packages

3. Use the runtime option: `-ksp_type preonly -pc_type <pctype> -pc_factor_mat_solver_package <packagename>`. For eg: `-ksp_type preonly -pc_type lu -pc_factor_mat_solver_package superlu_dist`.

MatType	PCType	MatSolverPackage	Package (-pc_factor_mat_solver_package)
baij	cholesky	MAT_SOLVER_DSCPACK	dscpack
seqaij	lu	MAT_SOLVER_ESSL	essl
seqaij	lu	MAT_SOLVER_LUSOL	lusol
seqaij	lu	MAT_SOLVER_MATLAB	matlab
aij	lu	MAT_SOLVER_MUMPS	mumps
sbaij	cholesky		
plapack	lu	MAT_SOVLER_PLAPACK	plapack
plapack	cholesky		
aij	lu	MAT_SOLVER_SPOOLES	spooles
sbaij	cholesky		
seqaij	lu	MAT_SOLVER_SUPERLU	superlu
aij	lu	MAT_SOLVER_SUPERLU_DIST	superlu_dist
seqaij	lu	MAT_SOLVER_UMFPACK	umfpack

Table 5: Options for External Solvers

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations *(cont..)*	Iterative Methods *(cont..)*	SYMMLQ	PETSc
		Precondition CG	AztecOO PETSc Hypre
		Richardson	PETSc
		Block Jacobi Preconditioner	AztecOO PETSc Hypre
		Point Jocobi Preconditioner	AztecOO
		Least Squares Polynomials	PETSc

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithms	Library
Systems of Linear Equations (*cont..*)	Iterative Methods (*cont..*)	SOR Preconditioning	PETSc
		Overlapping Additive Schwartz	PETSc
		Approximate Inverse	Hypre
		Sparse LU preconditioner	AztecOO PETSc Hypre
		Incomplete LU (ILU) preconditioner	AztecOO
		Least Squares Polynomials	PETSc
	MultiGrid (MG) Methods	MG Preconditioner	PETSc Hypre
		Algebraic MG	Hypre
		Semi-coarsening	Hypre

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithm	Library
Linear Least Squares Problems	Least Squares	$min_x \parallel b - Ax \parallel_2$	ScaLAPACK
	Minimum Norm Solution	$min_x \parallel x \parallel_2$	ScaLAPACK
	Minimum Norm Least Squares	$min_x \parallel b - Ax \parallel_2$ $min_x \parallel x \parallel_2$	ScaLAPACK
Standard Eigenvalue Problem	Symmetric Eigenvalue Problem	$Az = \lambda z$ For $A=A^H$ or $A=A^T$	ScaLAPACK (dense) SLEPc (sparse)
Singular Value Problem	Singular Value Decomposition	$A = U \Sigma V^T$ $A = U \Sigma V^H$	ScaLAPACK (dense) SLEPc (sparse)
Generalized Symmetric Definite Eigenproblem	Eigenproblem	$Az = \lambda Bz$ $ABz = \lambda z$ $BAz = \lambda z$	ScaLAPACK (dense) SLEPc (sparse)

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithm	Library
Non-Linear Equations	Newton Based	Line Search	PETSc
		Trust Regions	PETSc
		Pseudo-Transient Continuation	PETSc
		Matrix Free	PETSc

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithm	Library
Non-Linear Optimization	Newton Based	Newton	OPT++
			TAO
		Finite-Difference Newton	OPT++
			TAO
		Quasi-Newton	OPT++
			TAO
		Non-linear Interior Point	OPT++
			TAO
	CG	Standard Non-linear CG	OPT++
			TAO
		Limited Memory BFGS	OPT++
		Gradient Projections	TAO
	Direct Search	No derivate information	OPT++

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithm	Library
Non-Linear Optimization	Newton Based	Newton	OPT++
			TAO
		Finite-Difference Newton	OPT++
			TAO
		Quasi-Newton	OPT++
			TAO
		Non-linear Interior Point	OPT++
			TAO
	CG	Standard Non-linear CG	OPT++
			TAO
		Limited Memory BFGS	OPT++
		Gradient Projections	TAO
	Direct Search	No derivate information	OPT++

U.S. DEPARTMENT OF ENERGY | Office of Science

ACTS COLLECTION

BERKELEY LAB
LAWRENCE BERKELEY NATIONAL LABORATORY

UNIVERSITY OF OREGON

Functionality in The DOE ACTS Collection

Computational Problem	Methodology	Algorithm	Library
Non-Linear Optimization (cont..)	Semismoothing	Feasible Semismooth	TAO
		Unfeasible semismooth	TAO
Ordinary Differential Equations	Integration	Adam-Moulton (Variable coefficient forms)	CVODE (SUNDIALS) CVODES
	Backward Differential Formula	Direct and Iterative Solvers	CVODE CVODES
Nonlinear Algebraic Equations	Inexact Newton	Line Search	KINSOL (SUNDIALS)
Differential Algebraic Equations	Backward Differential Formula	Direct and Iterative Solvers	IDA (SUNDIALS)

Functionality in The DOE ACTS Collection

Computational Problem	Support	Techniques	Library
Writing Parallel Programs	Distributed Arrays	Shared-Memory	Global Arrays
		Grid Generation	OVERTURE
		Structured Meshes	Hypre OVERTURE PETSc
		Semi-Structured Meshes	Hypre OVERTURE

Functionality in The DOE ACTS Collection

Computational Problem	Support	Technique	Library
Profiling	Algorithmic Performance	Automatic instrumentation	PETSc
		User Instrumentation	PETSc
	Execution Performance	Automatic Instrumentation	TAU
		User Instrumentation	TAU
Code Optimization	Library Installation	Linear Algebra Tuning	ATLAS

References

- L.A. Drummond, O. Marques: An Overview of the Advanced CompuTational Software (ACTS) Collection. ACM Transactions on Mathematical Software Vol. 31 pp. 282-301, 2005

- http://acts.nersc.gov

- http://acts.nersc.gov/events/Workshop2011