The DOE ACTS Collection

Functionality and Lessons Learned

Changing the Face of Mathematical Software
NIST - June, 2004

ACTS
COLLECTION

Tony Drummond
Lawrence Berkeley National Laboratory
acts-support@nersc.gov
LADrummond@lbl.gov

Berkeley Lab
Office of Science
U.S. DEPARTMENT OF ENERGY

Who is in ACTS?

- Founded by DOE MICS office
- Osni Marques (PI) and Tony Drummond (co-PI)
- 18 development teams from ANL, LANL, LBNL, LLNL, ORNL, PNL, SNL, UCB, U. of Oregon, UTK, other universities

What is ACTS?

- Advanced CompuTational Software Collection
- Tools for developing parallel applications
- ACTS started as an “umbrella” project

Some of the Goals

- *Extended support for experimental software*
- *Provide technical support (acts-support@nersc.gov)*
- *Maintain ACTS information center (http://acts.nersc.gov)*
- *Coordinate efforts with other supercomputing centers*
- *Enable large scale scientific applications*
- *Educate and train*

Upcoming Events: ACTS Workshop

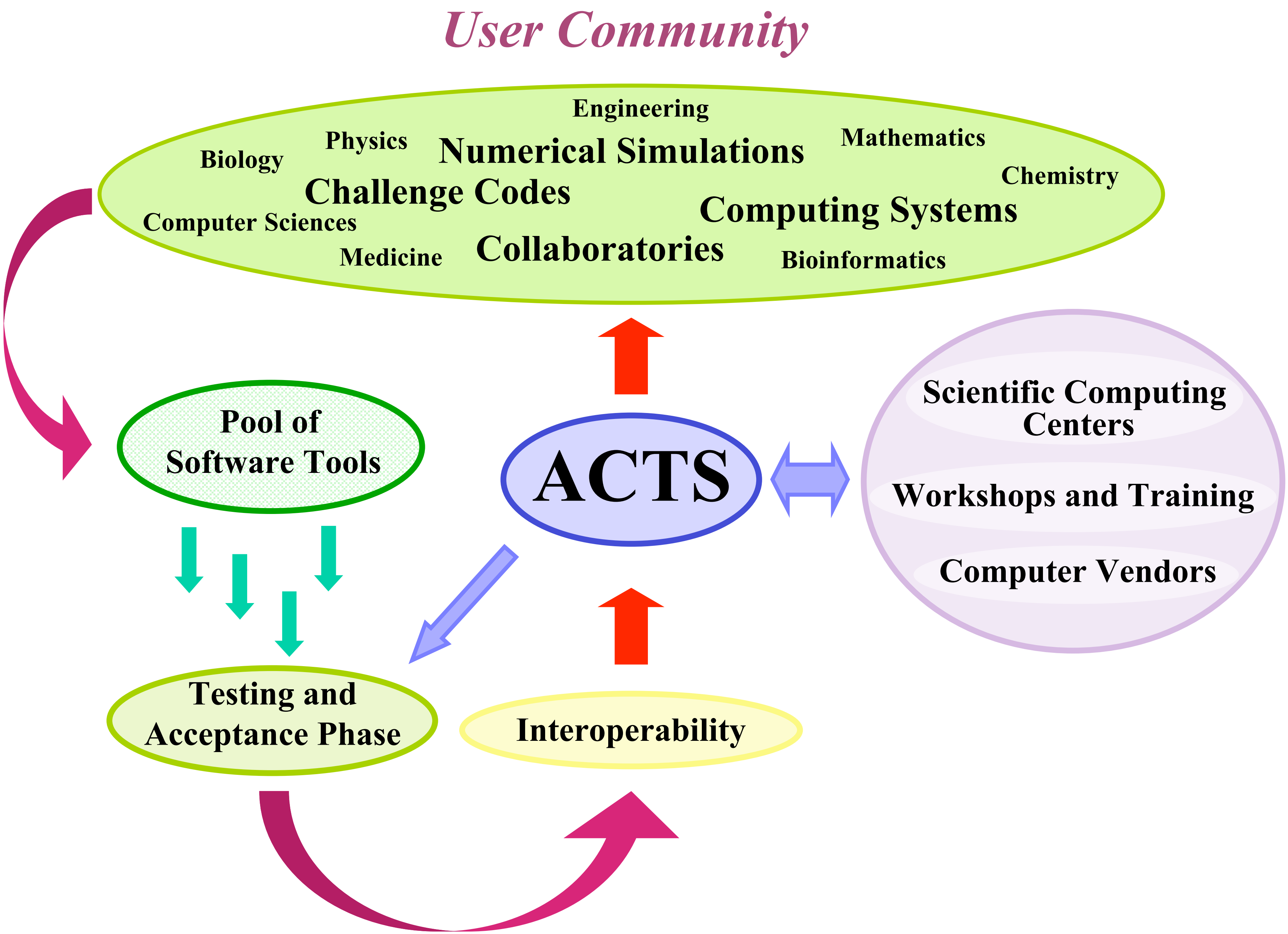
For more information:

http://acts.nersc.gov/events/Workshop2004

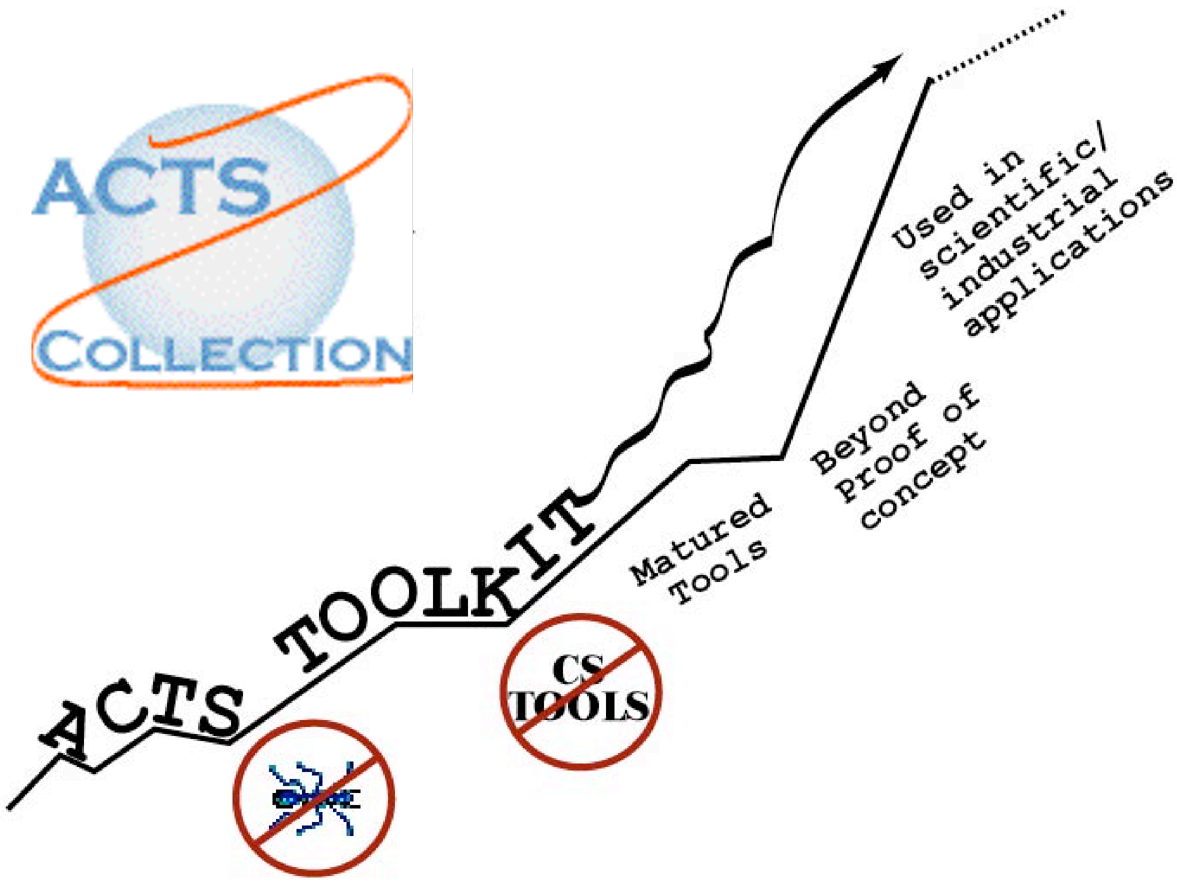
Enabling Technologies For High End Computer Simulations

Lawrence Berkeley National Laboratory

August 24-27, 2004


User Community
Engineering
Physics
Numerical Simulations
Mathematics
Biology
Chemistry
Challenge Codes
Computing Systems
Computer Sciences
Medicine
Collaboratories
Bioinformatics
Pool of Software Tools
ACTS
Scientific Computing Centers
Workshops and Training
Computer Vendors
Testing and Acceptance Phase
Interoperability

The Way to Users



Requirements for Mathematical Software

min[*time_to_first_solution*] (prototype)

→ **min**[*time_to_solution*] (production)

- Outlive Complexity
  - Increasingly sophisticated models
  - Model coupling
  - Interdisciplinary

  } (Software Evolution)

- Sustained Performance
  - Increasingly complex algorithms
  - Increasingly complex architectures
  - Increasingly demanding applications

  } (Long-term deliverables)

1

A close look at ACTS Numerical Software Tools Interfaces and Functionality

Software 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 )
```

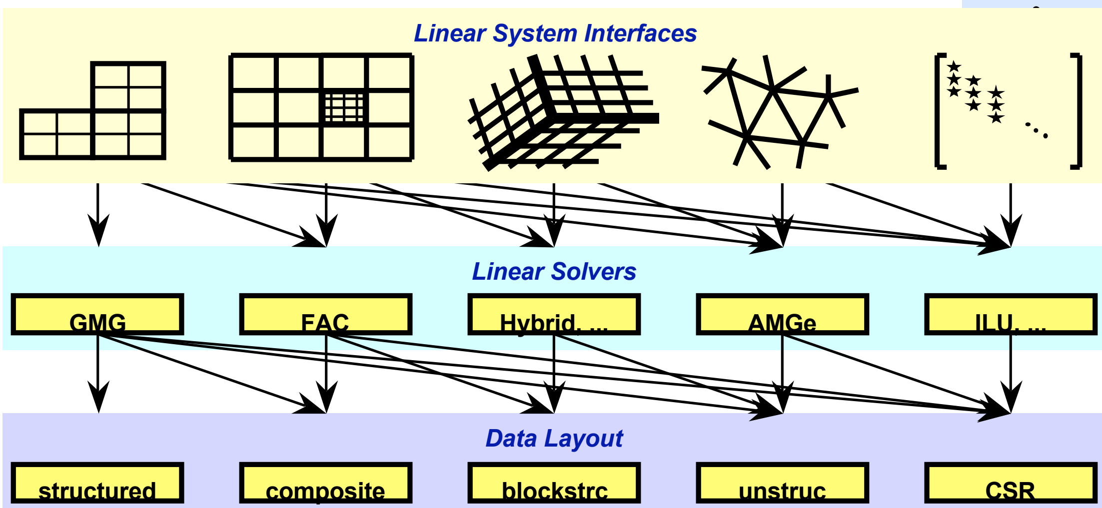
Library Calls

Command lines

- -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>
- -pc_asm_type [basic,restrict,interpolate,none]



Problem Domain

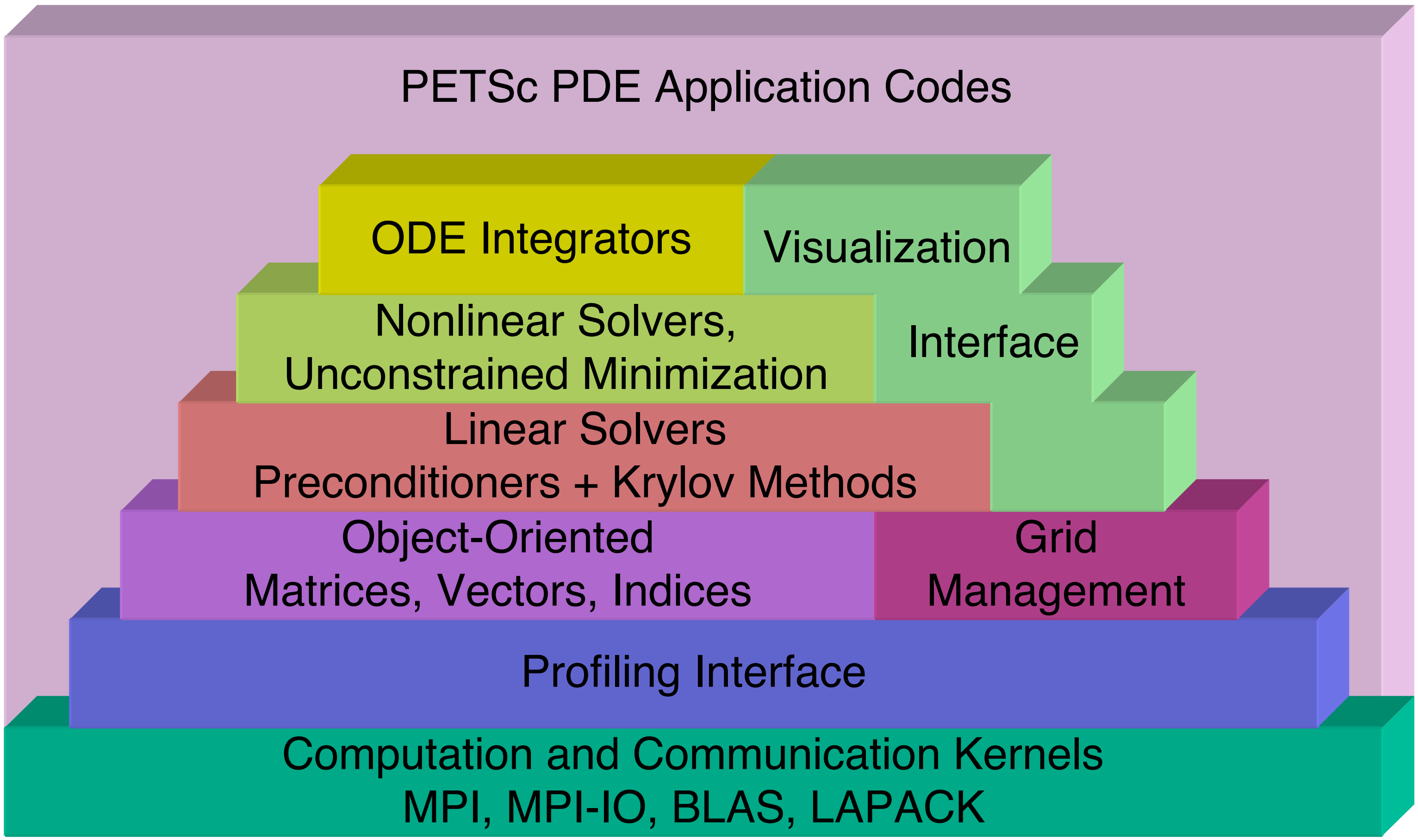
ACTS Numerical Tools: *Functionality*

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

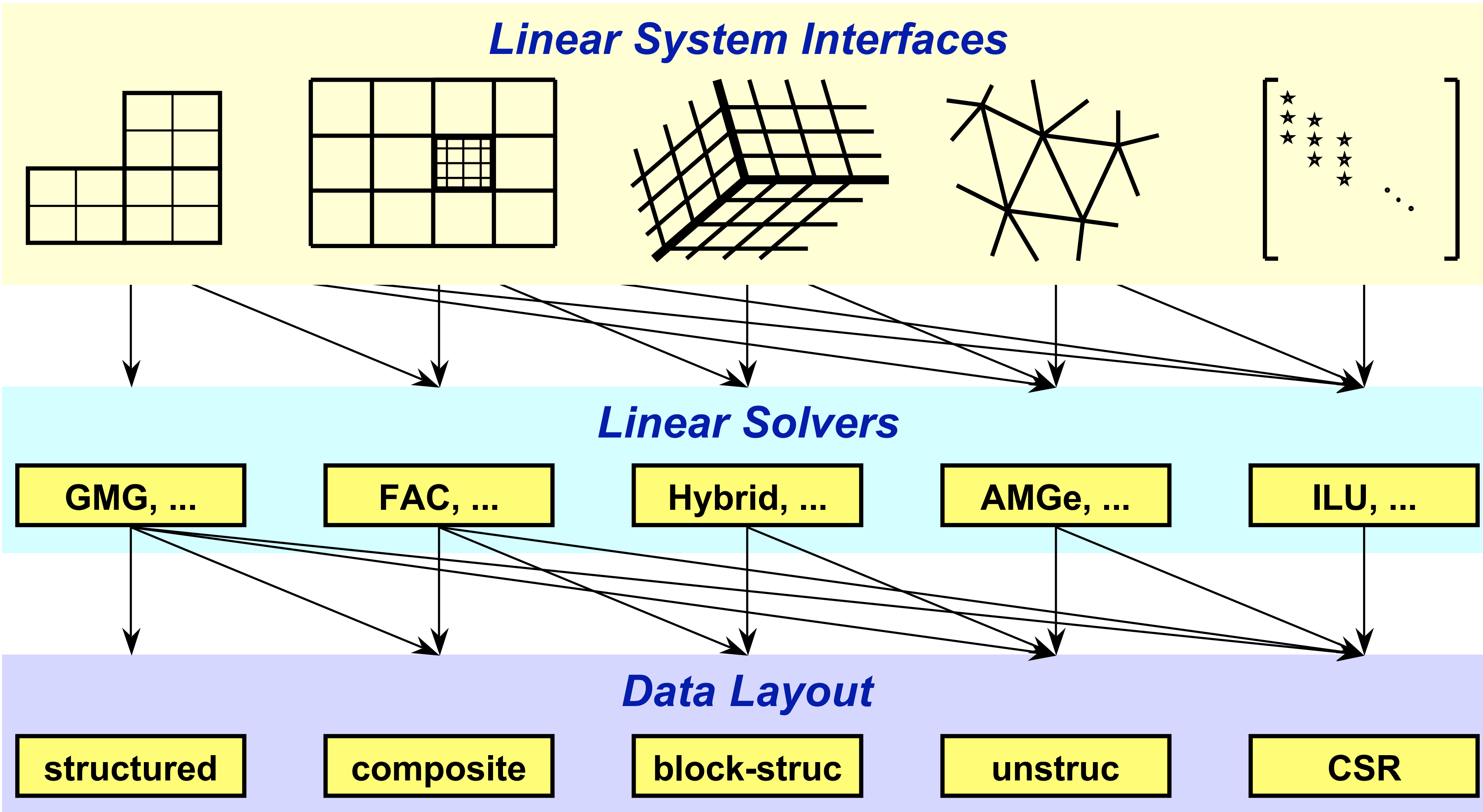
ACTS Numerical Tools: *Functionality*

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

Structure of PETSc



Hypre Conceptual Interfaces



INTERFACE TO SOLVERS

List of Solvers and Preconditioners per Conceptual Interface

Solvers	System Interfaces Struct	SStruct	FEI	IJ
Jacobi	X			
SMG	X			
PFMG	X			
BoomerAMG	X	X	X	X
ParaSails	X	X	X	X
PILUT	X	X	X	X
Euclid	X	X	X	X
PCG	X	X	X	X
GMRES	X	X	X	X

ACTS Numerical Tools: *Functionality*

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

ACTS Numerical Tools: *Functionality*

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

ACTS Numerical Tools: *Functionality*

Computational Problem	Methodology	Algorithm	Library
Linear Least Squares Problems	Least Squares	$min_x \|\| b - Ax \|\|_2$	ScaLAPACK
	Minimum Norm Solution	$min_x \|\| x \|\|_2$	ScaLAPACK
	Minimum Norm Least Squares	$min_x \|\| b - Ax \|\|_2$ $min_x \|\| x \|\|_2$	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)

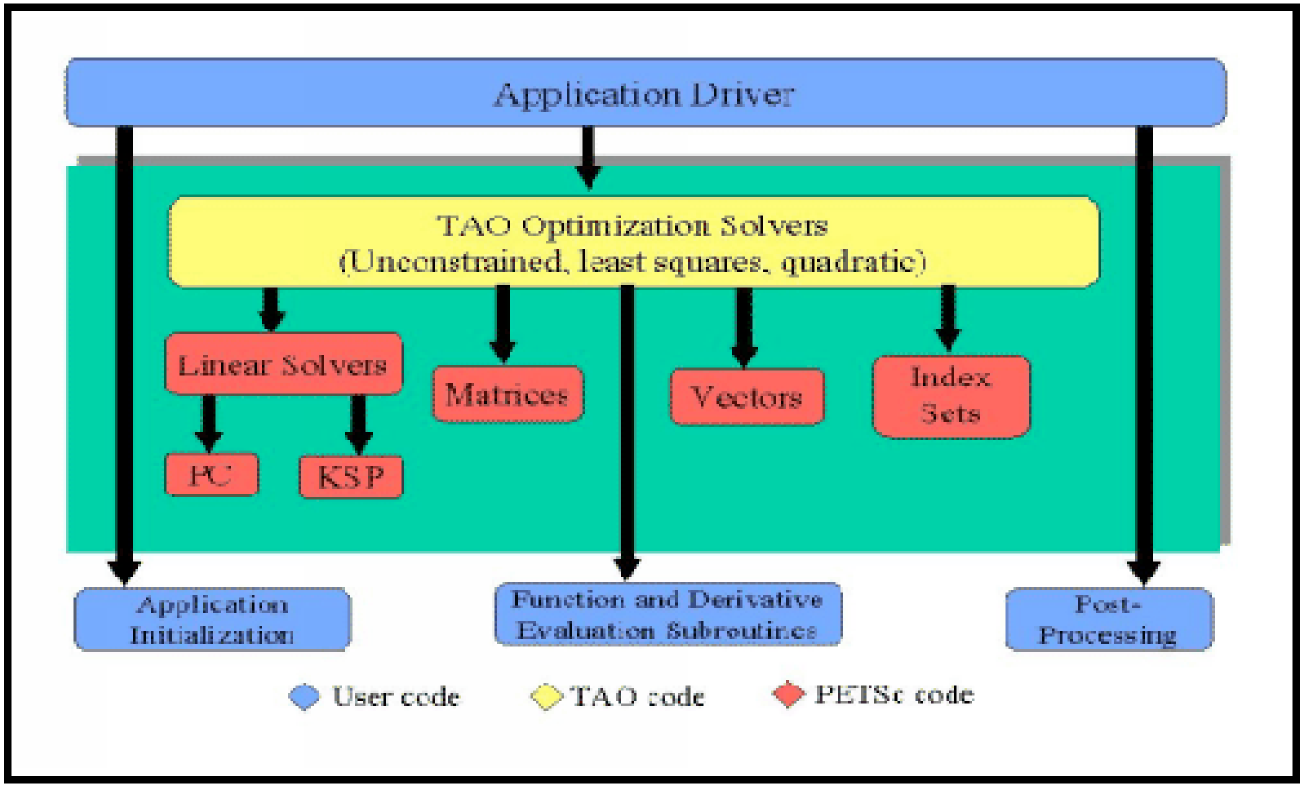
ACTS Numerical Tools: *Functionality*

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

ACTS Numerical Tools: *Functionality*

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++

TAO - Interface with PETSc



OPT++ Interfaces

- Four major classes of problems available
  - *NLF0(ndim, fcn, init_fcn, constraint)*
    - Basic nonlinear function, no derivative information available
  - *NLF1(ndim, fcn, init_fcn, constraint)*
    - Nonlinear function, first derivative information available
  - *FDNLF1(ndim, fcn, init_fcn, constraint)*
    - Nonlinear function, first derivative information approximated
  - *NLF2(ndim, fcn, init_fcn, constraint)*
    - Nonlinear function, first and second derivative information available

ACTS Numerical Tools: *Functionality*

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++

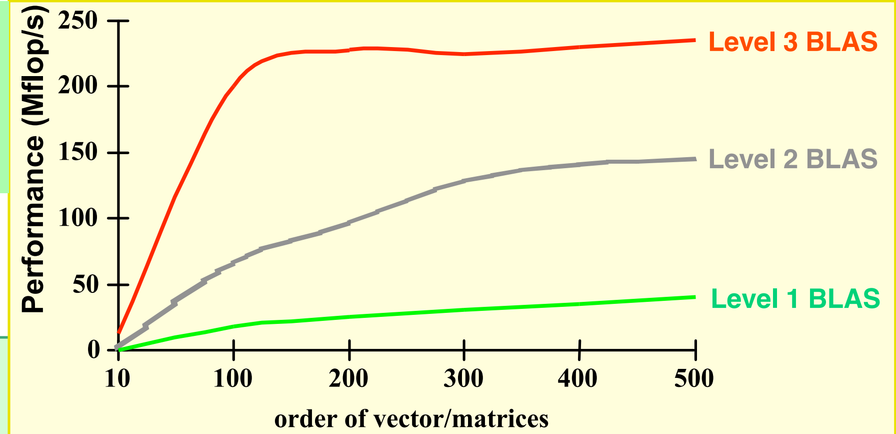
ACTS Numerical Tools: *Functionality*

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)

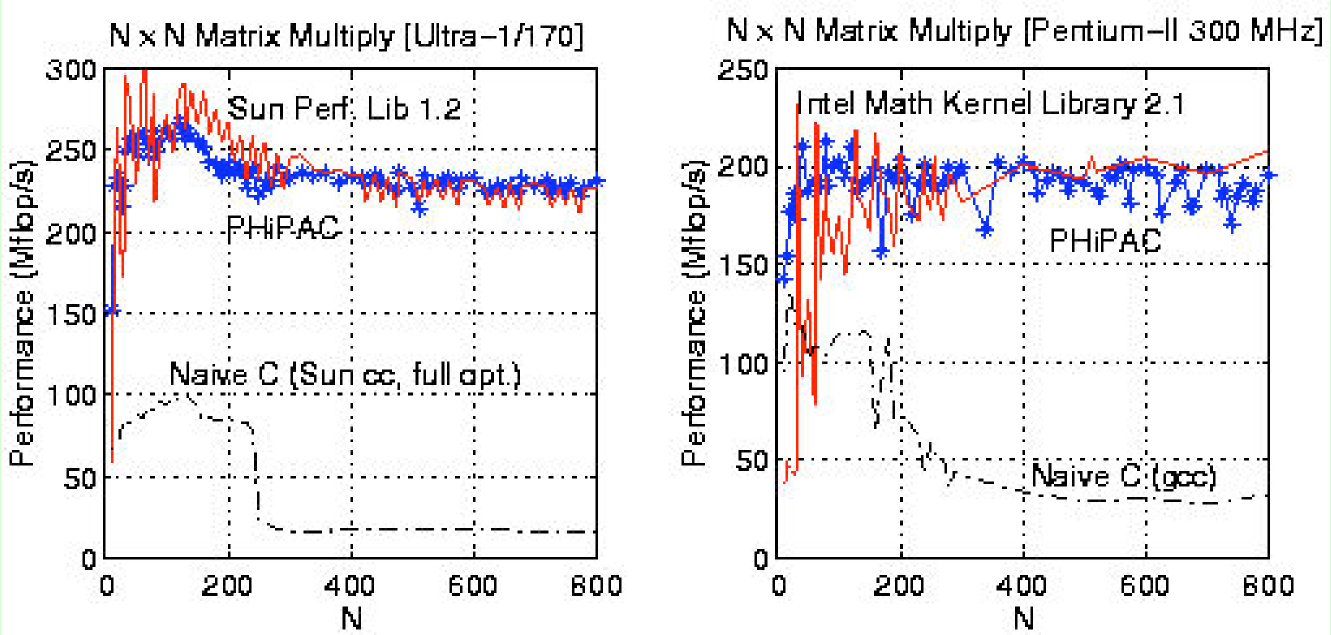
More ACTS functionality

Automatic Tuning

- PHiPAC: *www.icsi.berkeley.edu/~bilmes/phipac*
- ATLAS: *www.netlib.org/atlas*



PHiPAC: C = A * B



②

The Lessons Learned

Lessons Learned

- *There is still a gap between tool developers and application developers which leads to duplication of efforts.*

"We need to move away from a coding style suited for serial machines, where every macrostep of an algorithm needs to be thought about and explicitly coded, to a [illegible] compiler and library [illegible] And the remarkable t[illegible]her-level approach right now, even on today's machines, we will see immediate benefits in our productivity."

SciDAC
Scientific Discovery through Advanced Computing
DOE Office of Science ASCR BER BES FE HENP MICS

W. H. Press and S. A. Teukolsky, 1997
Numerical Recipes: Does This Paradigm Have a future?

Lessons Learned

- *Users demand long-term support of the tools.*

- ***High***
  - Intermediate level
  - Tool expertise
  - Conduct tutorials
- ***Intermediate***
  - Basic level
  - Higher level of support to users of the tool
- ***Basic***
  - Help with installation
  - Basic knowledge of the tools
  - Compilation of user's reports

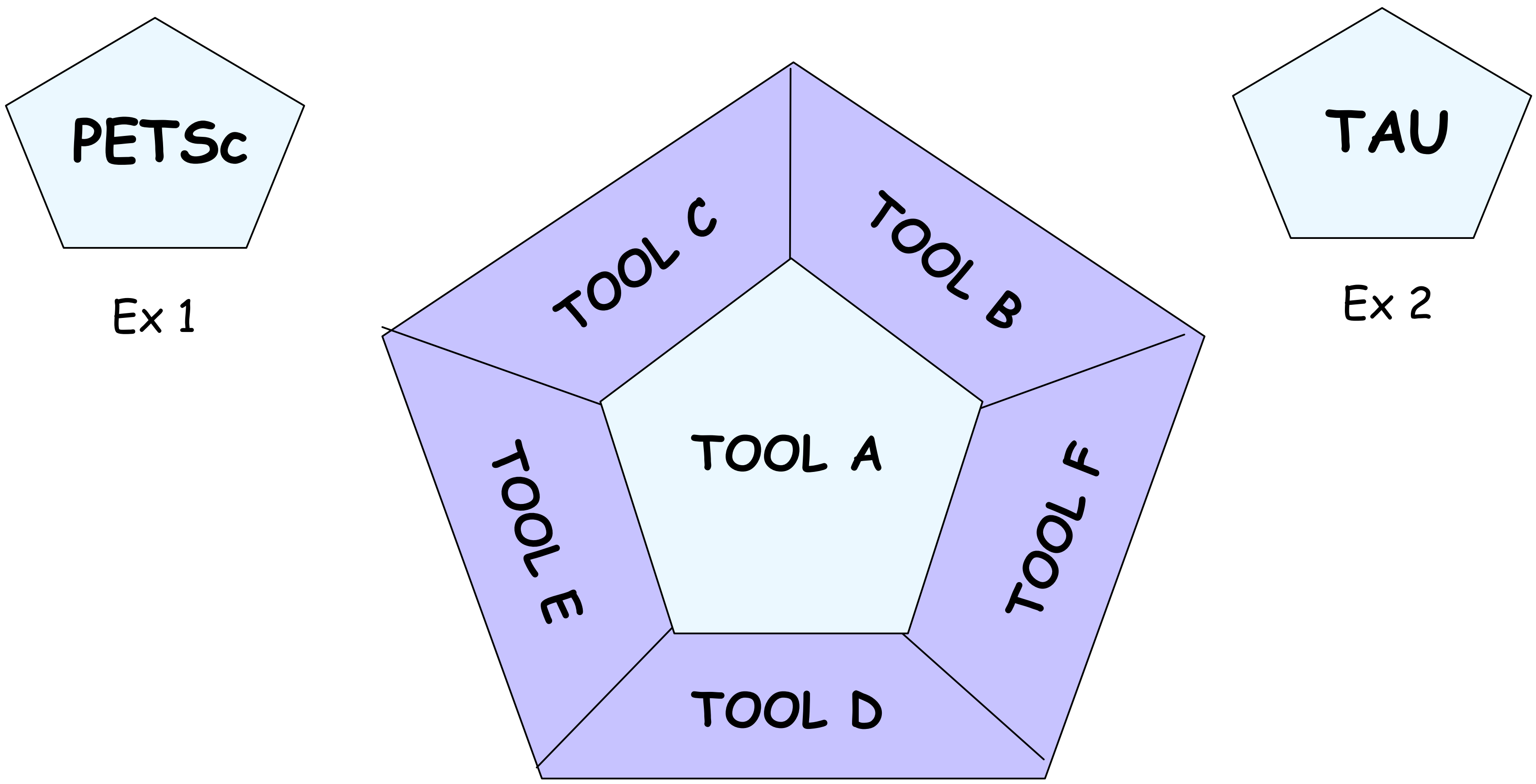
Lessons Learned

- *Tools evolve or are superseded by other tools*
  - Integration to frameworks - TRILINOS (Aztec → Aztec00)

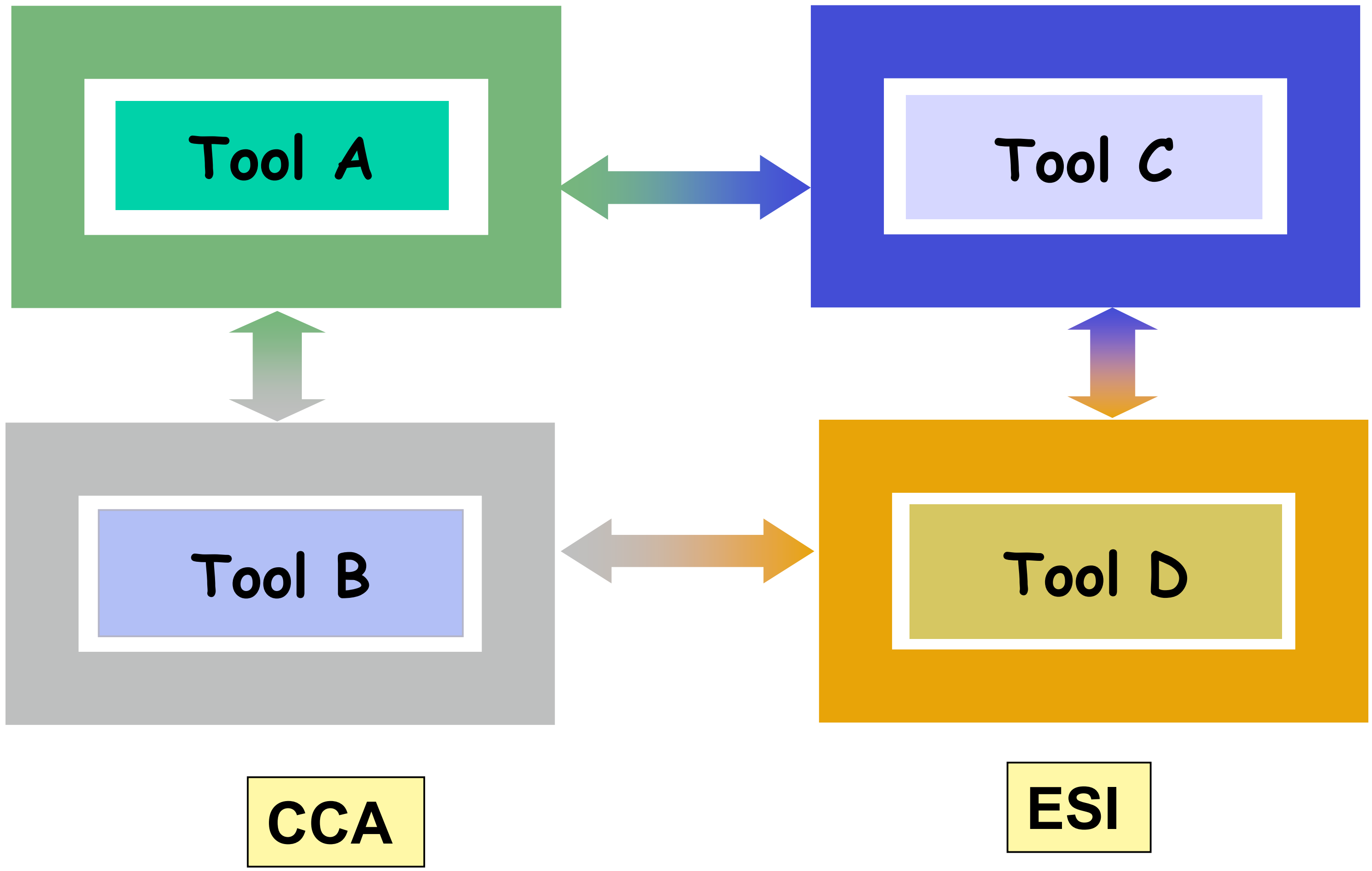
Lessons Learned

- *There is a demand for tool interoperability and more uniformity in the documentation and user interfaces*.
  - Tool-to-Tool
  - Software Frameworks
  - Problem Solving Environments

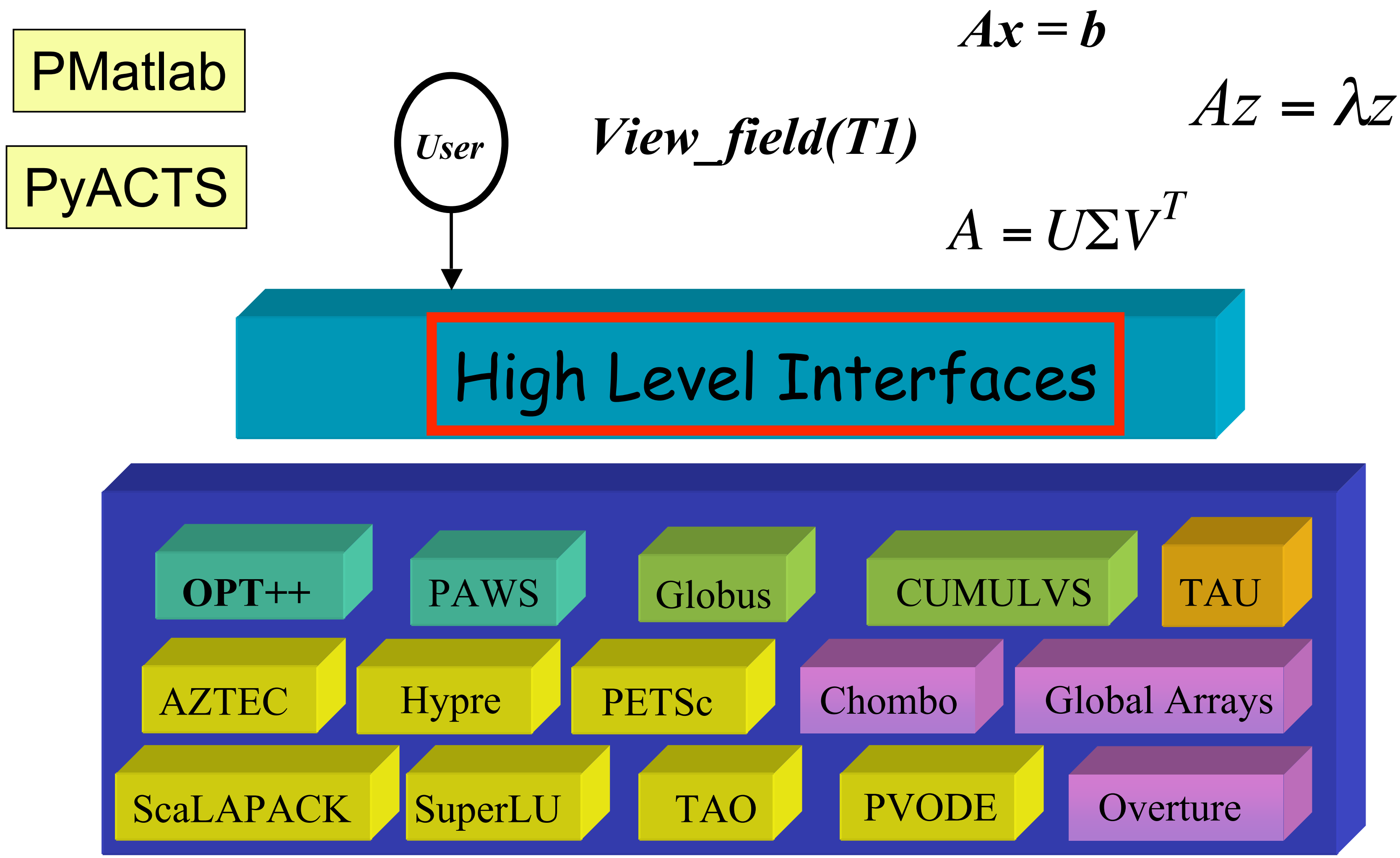
Tool Interoperability
Tool-to-Tool



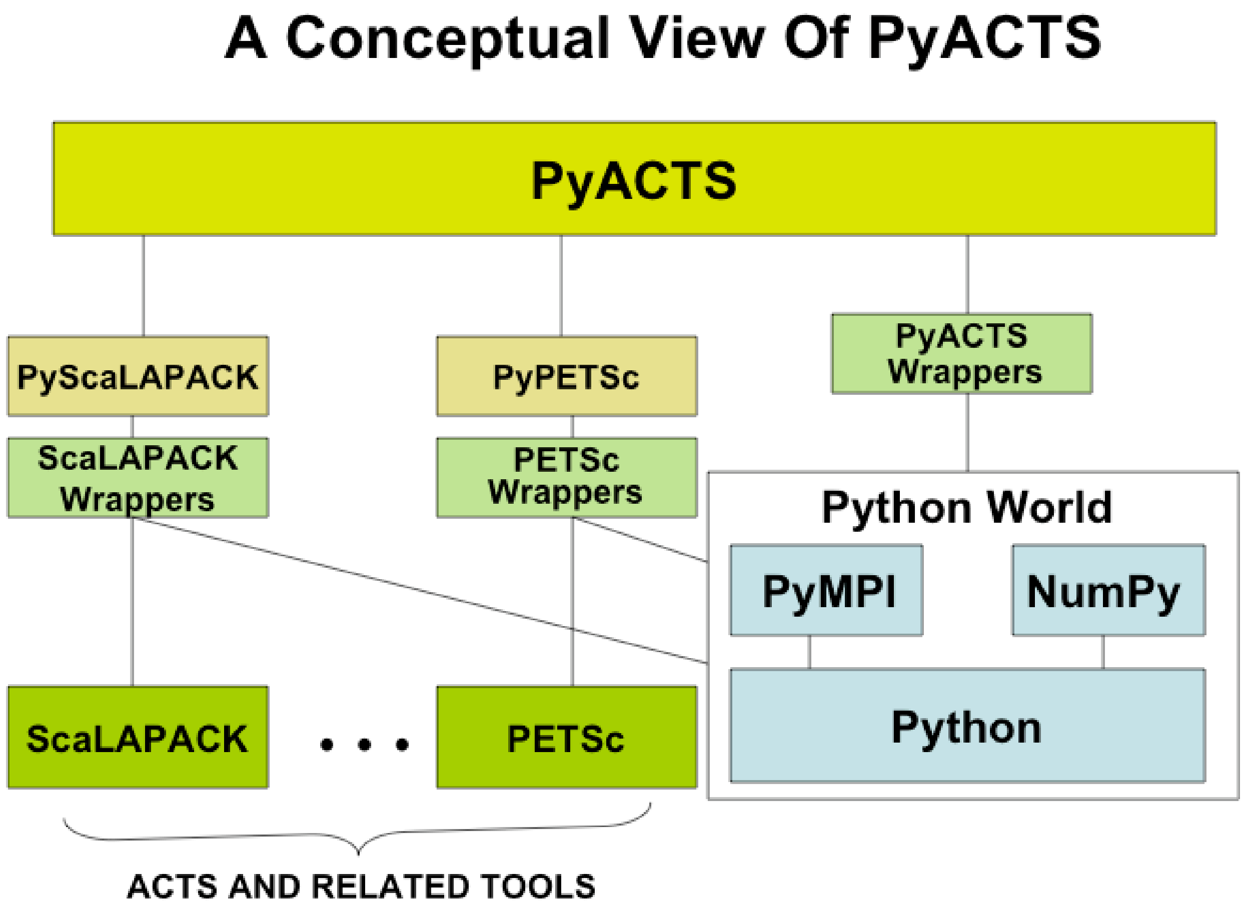
Component Technology



PSE's


PMatlab
PyACTS
User
View_field(T1)
$Ax = b$
$Az = \lambda z$
$A = U\Sigma V^T$
High Level Interfaces
OPT++
PAWS
Globus
CUMULVS
TAU
AZTEC
Hypre
PETSc
Chombo
Global Arrays
ScaLAPACK
SuperLU
TAO
PVODE
Overture

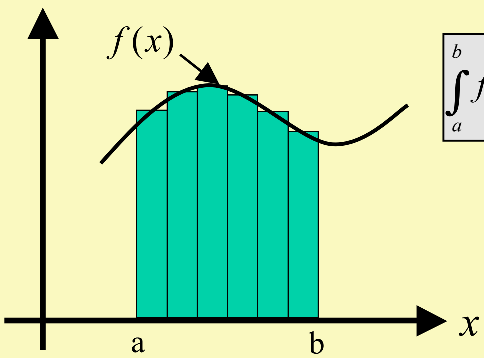
PyACTS


A Conceptual View Of PyACTS
PyACTS
PyScaLAPACK
PyPETSc
PyACTS Wrappers
ScaLAPACK Wrappers
PETSc Wrappers
Python World
PyMPI
NumPy
Python
ScaLAPACK
. . .
PETSc
ACTS AND RELATED TOOLS

Software Engineering Based in Components

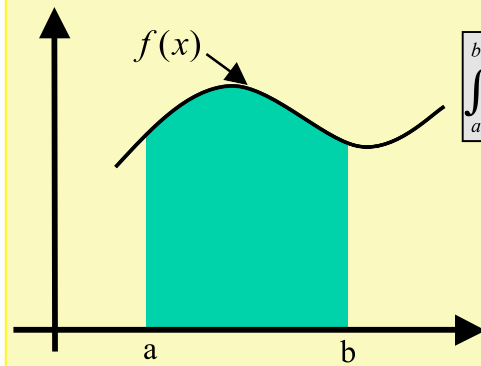
http://www.cca-forum.org

Numerical integration: *midpoint*

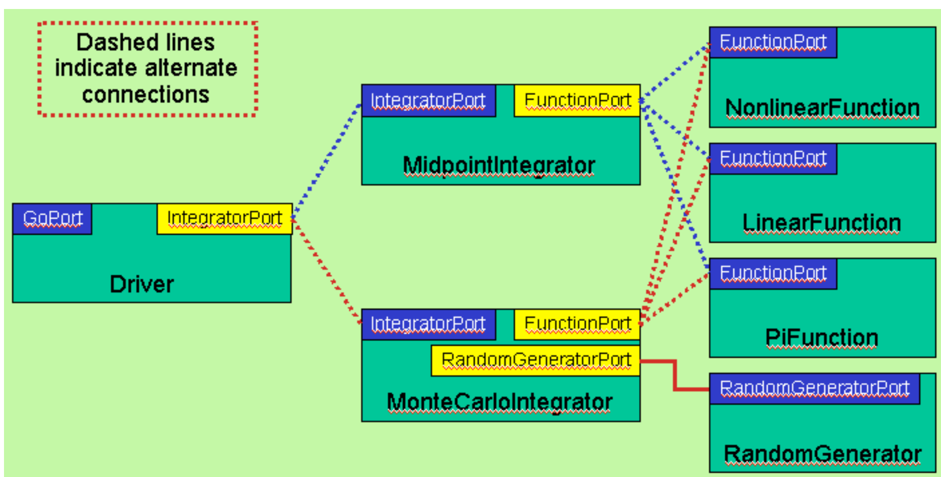


$$\int_a^b f(x)dx \approx \frac{b-a}{n}\sum_{j=1}^{n} f\left(\frac{x_{j-1}+x_j}{2}\right)$$

Numerical integration: *MonteCarlo*



$$\int_a^b f(x)dx \approx \frac{1}{b-a}\left(\frac{1}{N}\sum_{i=1}^{N} f(x_n)\right)$$



$$f_1(x) = x^2$$

$$f_2(x) = 2x$$

$$f_3(x) = \frac{4}{1+x^2}$$

Lessons Learned

- *Applications and users play an important role in making the tools mature*.

Matrix of Scientific Applications and Tools

Application	Computational Problem	Software Tools	Highlights
MADCAP	Matrix factorization and triangular solves	ScaLAPACK	• 50% peak performance on an IBM SP • Nearly perfect scalability on 1024, 2048, 3072 and 4096 processors • Fast implementation of numerical algorithms
3-Charged Particles	Solution of large, complex unsymmetric linear systems	SuperLU	• Solves systems of equations of order 8.4 million on 64 processors in 1 hour of wall clock time • 30 GFLOPs
NWChem	Distribute large data arrays, collective operations	Global Arrays and LAPACK	• Very good scaling for large problems

http://acts.nersc.gov/AppMat

Enabling sciences and discoveries… with high performance and scalability...

... More Applications …

FUN3D &	Unstructured grids, compressible and incompressible Euler and Navier-Stokes equations.	PETSc	• Parallelization of legacy code • Gordon Bell price, 0.23 Tflop/s on 3072 procs of ASCI Red
P-FLAPW	Eigenvalue problems	ScaLAPACK	• Study of systems up to 700 atoms (mat size=35,000) • Runs efficiently • Facilitated the study of new problems in materials science such as impurities and disordered systems.
NIMROD	Quad and triangular high order finite elements, semi-implicit time integration,sparse matrix solvers	SuperLU	• Code improvement of 5 fold, equivalent to 3-5 years progress in computing hardware.

Lessons Learned

- *The tools currently included in the ACTS Collection should be seen as dynamical configurable Collections and should be grouped into Collections upon user/application demand.*
- *There is a need for an intelligent and dynamic catalog/repository of high performance tools.*

FUTURE WORK

Future Plans

- Tool inclusion and *retirement*
  - *SciDAC ISIC products*
  - *Development outside the DOE umbrella*
- Long term support for maintenance/development
- Improve the Test and Evaluations
- Encourage more collaborations with computer vendors
- Improve ACTS information center