The Future of LAPACK and ScaLAPACK

Jason Riedy, Yozo Hida, James Demmel

EECS Department
University of California, Berkeley

November 18, 2005

Outline

Survey responses: What users want

Improving LAPACK and ScaLAPACK

Improved Numerics
Improved Performance
Improved Functionality
Improved Engineering and Community

Two Example Improvements

Numerics: Iterative Refinement for $Ax = b$
Performance: The MRRR Algorithm for $Ax = \lambda x$

Survey: What users want

- Survey available from
  http://www.netlib.org/lapack-dev/.
- 212 responses, over 100 different, non-anonymous groups
- Problem sizes:

100	1K	10K	100K	1M	(other)
8%	26%	24%	12%	6%	(24%)

- $>$80% interested in small-medium SMPs
- $>$40% interested in large distributed-memory systems
- Vendor libs seen as faster, buggier
- over 20% want $>$ double precision, 70% out-of-core
- Requests: High-level interfaces, low-level interfaces, parallel redistribution* and tuning

Participants

- UC Berkeley
  - Jim Demmel, Ming Gu, W. Kahan, Beresford Parlett, Xiaoye Li, Osni Marques, Christof Vömel, David Bindel, Yozo Hida, Jason Riedy, Jianlin Xia, Jiang Zhu, undergrads, ...
- U Tennessee, Knoxville
  - Jack Dongarra, Julien Langou, Julie Langou, Piotr Luszczek, Stan Tomov, ...
- Other Academic Institutions
  - UT Austin, UC Davis, U Kansas, U Maryland, North Carolina SU, San Jose SU, UC Santa Barbara, TU Berlin, FU Hagen, U Madrid, U Manchester, U Umeå, U Wuppertal, U Zagreb
- Research Institutions
  - CERFACS, LBL, UEC (Japan)
- Industrial Partners
  - Cray, HP, Intel, MathWorks, NAG, SGI

You?

Improved Numerics

Improved accuracy with standard asymptotic speed: Some are faster!

- Iterative refinement for linear systems, least squares
  Demmel / Hida / Kahan / Li / Mukherjee / Riedy / Sarkisyan
- Pivoting and scaling for symmetric systems
  - Definite and indefinite
- Jacobi SVD (and faster) — Drmač / Veselić
- Condition numbers and estimators
  Higham / Cheng / Tisseur
- Useful approximate error estimates

Improved Performance

Improved performance with at least standard accuracy

- MRRR algorithm for eigenvalues, SVD
  Parlett / Dhillon / Vömel / Marques / Willems / Katagiri
- Fast Hessenberg QR & QZ
  Byers / Mathias / Braman, Kågström / Kressner
- Fast reductions and BLAS2.5
  van de Geijn, Bischof / Lang, Howell / Fulton
- Recursive data layouts
  Gustavson / Kågström / Elmroth / Jonsson
- generalized SVD — Bai, Wang
- Polynomial roots from semi-separable form
  Gu / Chandrasekaran / Zhu / Xia / Bindel / Garmire / Demmel
- Automated tuning, optimizations in ScaLAPACK, ...

Improved Functionality

Algorithms

- Updating / downdating factorizations — Stewart, Langou
- More generalized SVDs: products, CSD — Bai, Wang
- More generalized Sylvester, Lyupanov solvers
  Kågström, Jonsson, Granat
- Quadratic eigenproblems — Mehrmann
- Matrix functions — Higham

Implementations

- Add “missing” features to ScaLAPACK
- Generate LAPACK, ScaLAPACK for higher precisions

Improved Engineering and Community

Use new features without a rewrite

- Use modern Fortran 95, maybe 2003
  - `DO ... END DO`, recursion, allocation (in wrappers)
- Provide higher-level wrappers for common languages
  - F95, C, C++
- Automatic generation of precisions, bindings
  - Full automation (FLAME, *etc.*) not quite ready for all functions
- Tests for algorithms, implementations, installations

Open development

Need a community for long-term evolution.

`http://www.netlib.org/lapack-dev/`

Lots of work to do, research *and* development.

Two Example Improvements

Recent, locally developed improvements

Improved Numerics

Iterative refinement for linear systems $Ax = b$:

- Extra precision $\Rightarrow$ small error, dependable estimate
- Both normwise and componentwise
- (See LAWN 165 for full details.)

Improved Performance

MRRR algorithm for eigenvalue, SVD problems

- Optimal complexity: $O(n)$ per value/vector
- (See LAWNs 162, 163, 166, 167... for more details.)

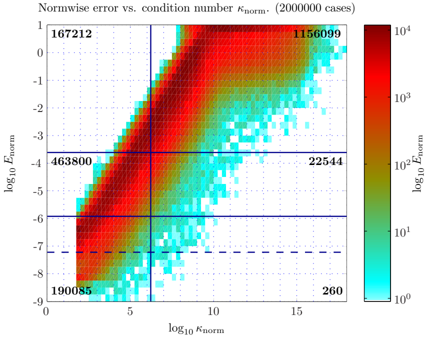
Numerics: Iterative Refinement

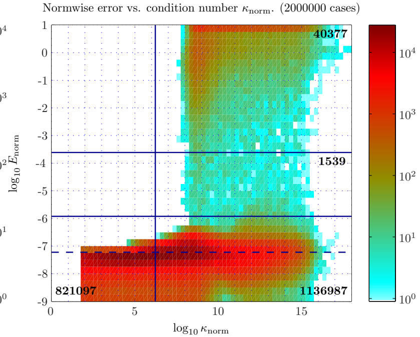
Improve solution to $Ax = b$

Repeat: $r = b - Ax$, $dx = A^{-1}r$, $x = x + dx$

Until: good enough

Not-too-ill-conditioned $\Rightarrow$ error $O(\sqrt{n}\,\varepsilon)$





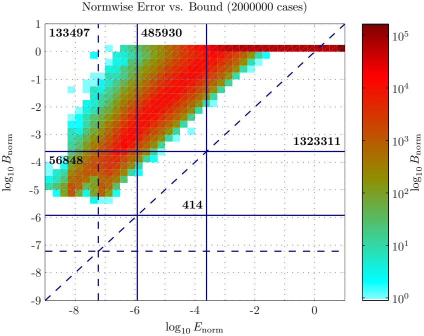
Numerics: Iterative Refinement

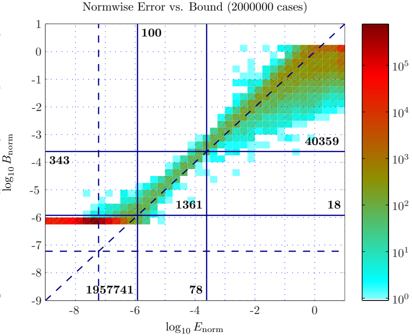
Improve solution to $Ax = b$

Repeat: $r = b - Ax$, $dx = A^{-1}r$, $x = x + dx$

Until: good enough

Dependable normwise relative error estimate





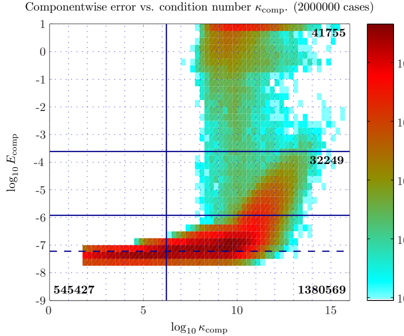
Numerics: Iterative Refinement

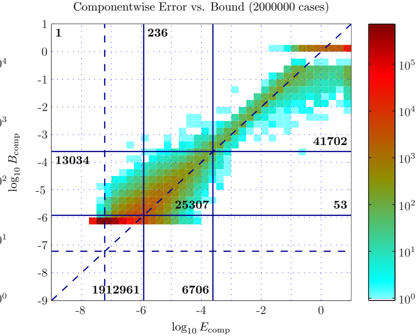
Improve solution to $Ax = b$

Repeat: $r = b - Ax$, $dx = A^{-1} r$, $x = x + dx$

Until: good enough

Also small componentwise errors and dependable estimates

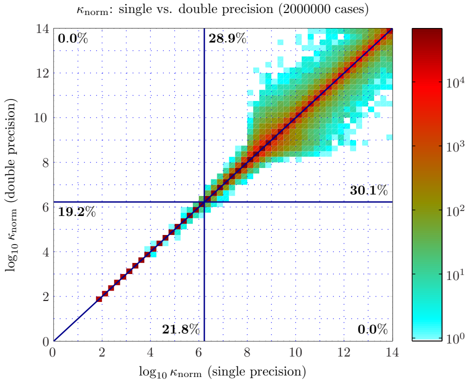




Relying on Condition Numbers

Need condition numbers for dependable estimates.

Picking the right condition number and estimating it well.

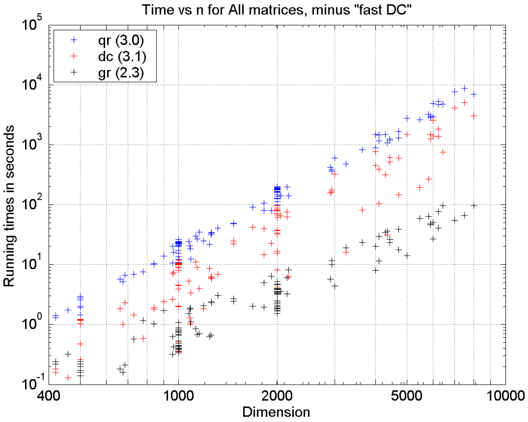


Performance: The MRRR Algorithm

Multiple Relatively Robust Representations

- 1999 Householder Award honorable mention for Dhillon
- Optimal complexity with small error!
  - $O(nk)$ flops for k eigenvalues/vectors of $n \times n$ tridiagonal matrix
  - Small residuals: $\|Tx_i - \lambda_i x_i\| = O(n\varepsilon)$
  - Orthogonal eigenvectors: $\|x_i^T x_j\| = O(n\varepsilon)$
- Similar algorithm for SVD.
- Eigenvectors computed independently $\Rightarrow$ naturally parallelizable
- (LAPACK r3 had bugs, missing cases)

Performance: The MRRR Algorithm



"fast DC": Wilkinson, deflate like crazy

Summary

- LAPACK and ScaLAPACK are open for improvement!
- Planned improvements in
  - numerics,
  - performance,
  - functionality, and
  - engineering.
- Forming a community for long-term development.