Outline

- **A quick survey of Dense Symmetric Eigensolvers**
- **New Eigensolvers in ScaLAPACK**
- **New Eigensolvers in LAPACK**
- **Templates for Eigenproblems**

Survey of Dense Symmetric Eigensolvers

- **Reduce to tridiagonal form ($O(n^3)$), then**
  - **Holy Grail: $O(n^2)$, embarrassingly parallel**
  - $\boxed{\boxed{\textbf{Bisection}}}$ **(+ accelerators)**
  - $\boxed{\boxed{\textbf{Inverse iteration}}}$
  - $\boxed{\textbf{QR}}$
  - $\boxed{\textbf{Fernando-Parlett-Dhillon}}$
  - **Divide-and-Conquer**
  - **QD**
  - **Homotopy**
- **Partial or no reduction**
  - **Band QR**
  - **Invariant Subspace Decomposition - PRISM**
  - **Successive band reduction**
  - **Jacobi**
- **Different algorithms in LAPACK and ScaLAPACK – for now**

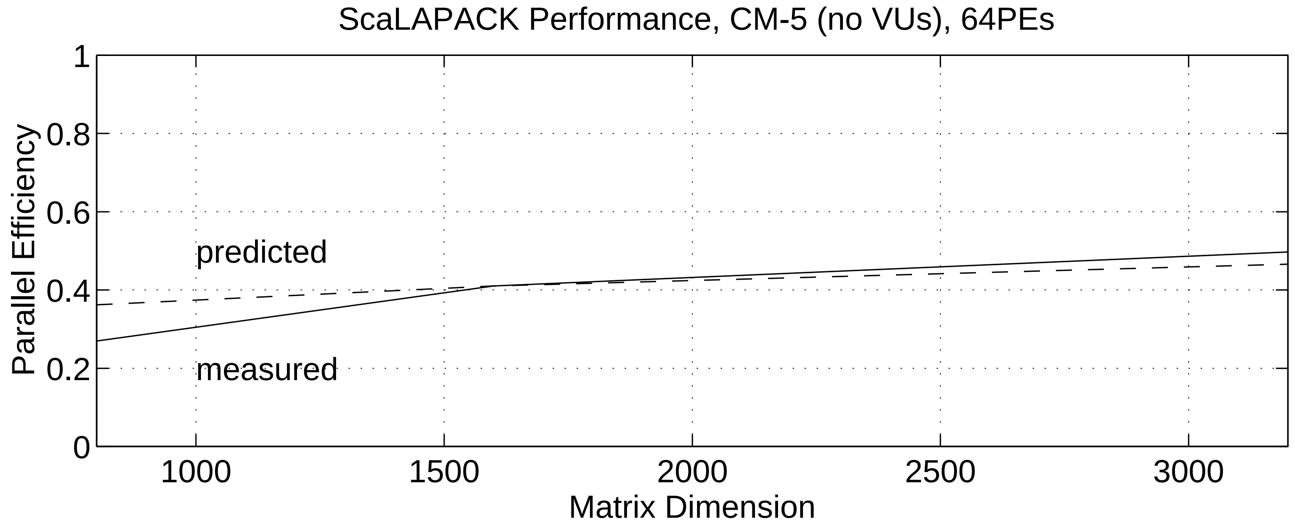
ScaLAPACK

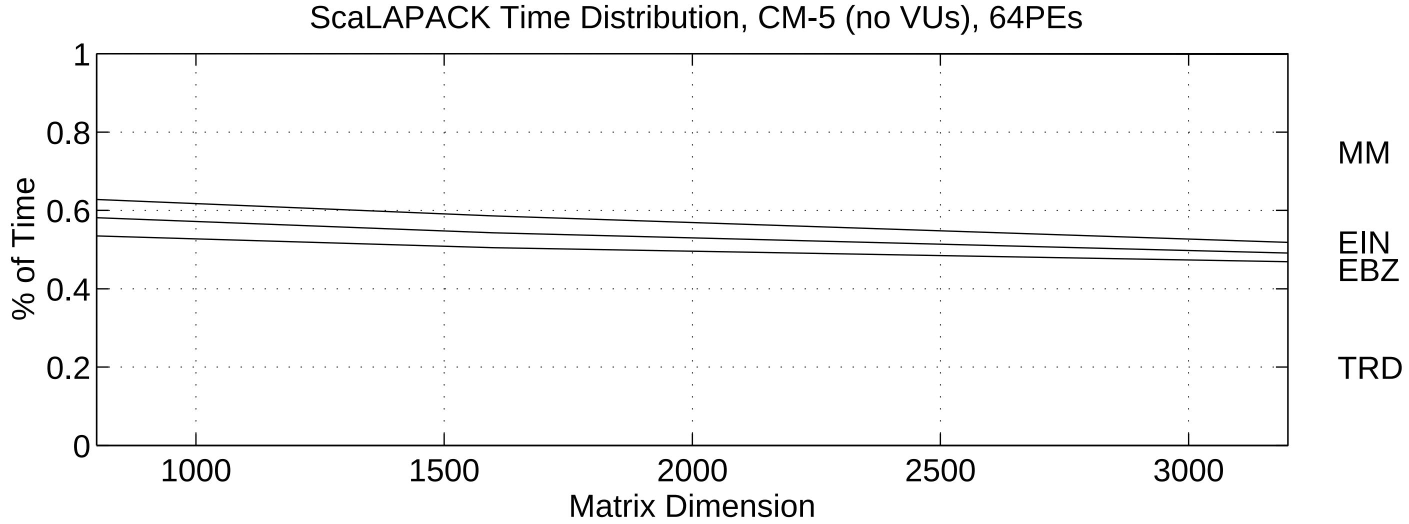
- Targets distributed memory machines
- Release 2.0 Feb 28 1995
- Direct and iterative methods
- Less coverage than LAPACK
- Software and templates
- Performance models

Symmetric Eigenroutines in ScaLAPACK

- PDSYEVX — Release Feb 28 1995
  - Algorithm:
    1. Tridiagonal reduction
    2. Bisection
    3. Inverse iteration with Partial Reorthogonalization
    4. Back Transformation
  - Static load balancing in Bisection
  - Exploit IEEE arithmetic in Bisection
  - Without clusters, steps 1, 4 dominate
  - Only guarantee orthogonality for "small" clusters, so cost $O(n^2/p)$, not $O(n^3)$
- PDSYEV — Future release
  - Algorithm:
    1. Tridiagonal reduction
    2. QR: tridiagonal part redundant
       update $1/p$-th of eigenvector matrix
  - Reliable, but slow
- Neither is Holy Grail

ScaLAPACK Performance on 64 Processor CM-5 (no VUs) Expert Driver - Bisection + Inverse iteration





LAPACK

- **Targets workstations, vector machines, shared memory**
- **Release 2.0 in Oct 1994**
- **2nd edition of manual from SIAM, WWW**
- **New feature of release 2.0**
  - **Faster routines for symmetric eigenproblems, based on divide-and-conquer**
  - **Faster bidiagonal SVD (s. values only)**
  - **Reducing a band matrix to bidiagonal form**
  - **Generalized nonsymmetric eigenproblem** $Ax = \lambda Bx$
  - **Generalized definite banded eigenproblem** $Ax = \lambda Bx$
  - **Generalized linear least squares problems (linear constraints; Gauss-Markov problem)**
  - **Generalized (quotient) SVD**
- **Next release - New SVD, least squares**

Outline of Cuppen's Divide and Conquer

1. **Goal: compute** $T = Q\Lambda Q^T$, $\Lambda = \mathbf{diag}(\lambda_i)$, $QQ^T = I$
2. $$T = \begin{bmatrix} a_1 & b_1 & & \\ b_1 & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & b_{n-1} & a_n \end{bmatrix} = \left[\begin{array}{c|c} T_1 & \\ \hline & T_2 \end{array}\right] + vv^T$$
3. **Solve** $T_1 = Q_1\Lambda_1 Q_1^T$ **and** $T_2 = Q_2\Lambda_2 Q_2^T$ **recursively**
4. $$T = \left[\begin{array}{c|c} Q_1\Lambda_1 Q_1^T & \\ \hline & Q_2\Lambda_2 Q_2^T \end{array}\right] + vv^T = \begin{bmatrix} Q_1 & \\ & Q_2 \end{bmatrix} \cdot (D + zz^T) \cdot \begin{bmatrix} Q_1^T & \\ & Q_2^T \end{bmatrix}$$
   where
   $$D = \begin{bmatrix} \Lambda_1 & \\ & \Lambda_2 \end{bmatrix} \text{ and } z = \begin{bmatrix} Q_1^T & \\ & Q_2^T \end{bmatrix} v$$
5. **Eigenvalues** λ **of** $D + zz^T$ **are roots of "secular equation"**
   $$1 + \sum_i \frac{z_i^2}{\lambda - d_{ii}} = 0$$
6. **Eigenvectors** v_j **of** $D + zz^T$ **are** $(D - \lambda_j I)^{-1} z$
7. **Eigenvectors of** T **are** $\begin{bmatrix} Q_1 & \\ & Q_2 \end{bmatrix} \cdot V$ **(BLAS3)**

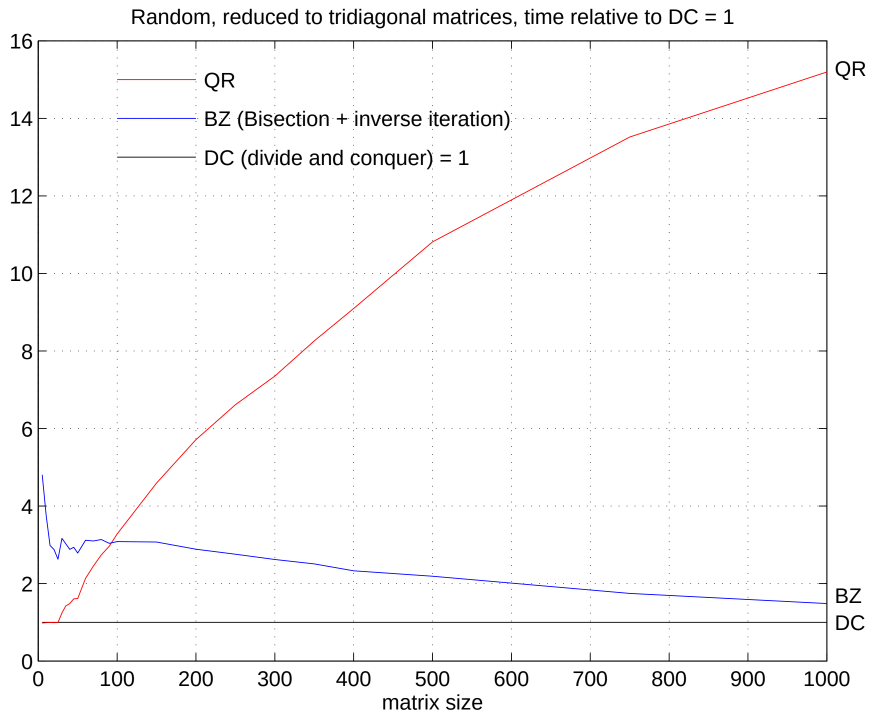
But $v_j = (D - \lambda_j I)^{-1} z$ is unstable!

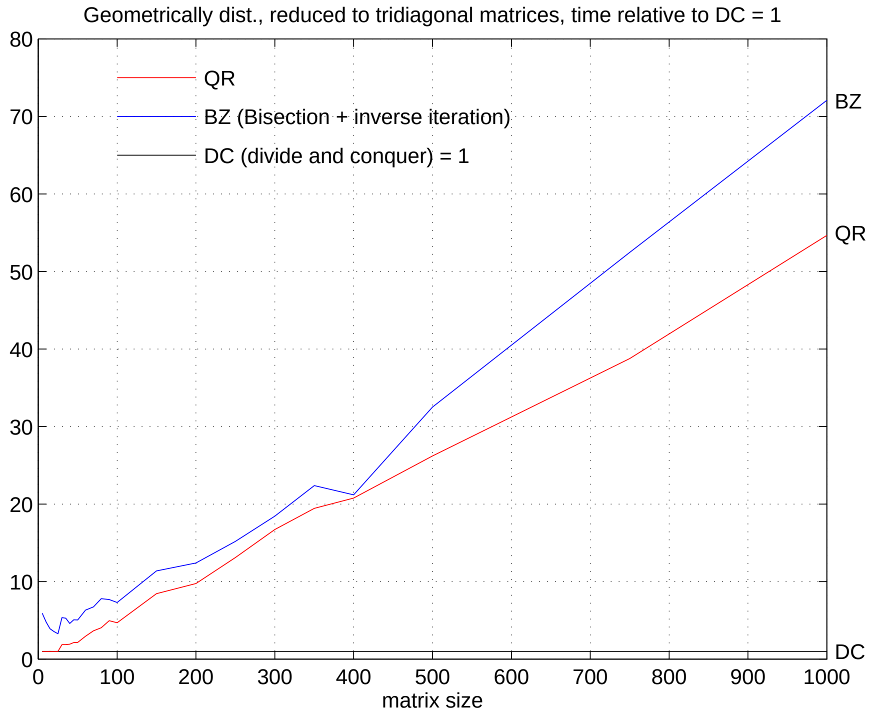
- **When two computed $\tilde{\lambda}_j$ are close, their v_j may not be orthogonal**
- **Idea of Sorensen, Tang, et al: compute $\tilde{\lambda}_j$ in double precision**
- **Idea of Gu, Eisenstat: find $\tilde{z}$ so that $\tilde{\lambda}_j$ is *exact* for $D + \tilde{z}\tilde{z}^T$, so $(D - \tilde{\lambda}_j)^{-1}\tilde{z}$ are accurate (orthogonal)**
- **Formula for $\tilde{z}$ (Löwner, too)**

$$\tilde{z}_i = \left[\prod_{j=1}^{i-1} \frac{\tilde{\lambda}_j - d_i}{d_j - d_i} \cdot \prod_{j=i}^{n-1} \frac{\tilde{\lambda}_{j+1} - d_i}{d_j - d_i} \cdot (\tilde{\lambda}_n - d_i) \right]^{1/2}$$

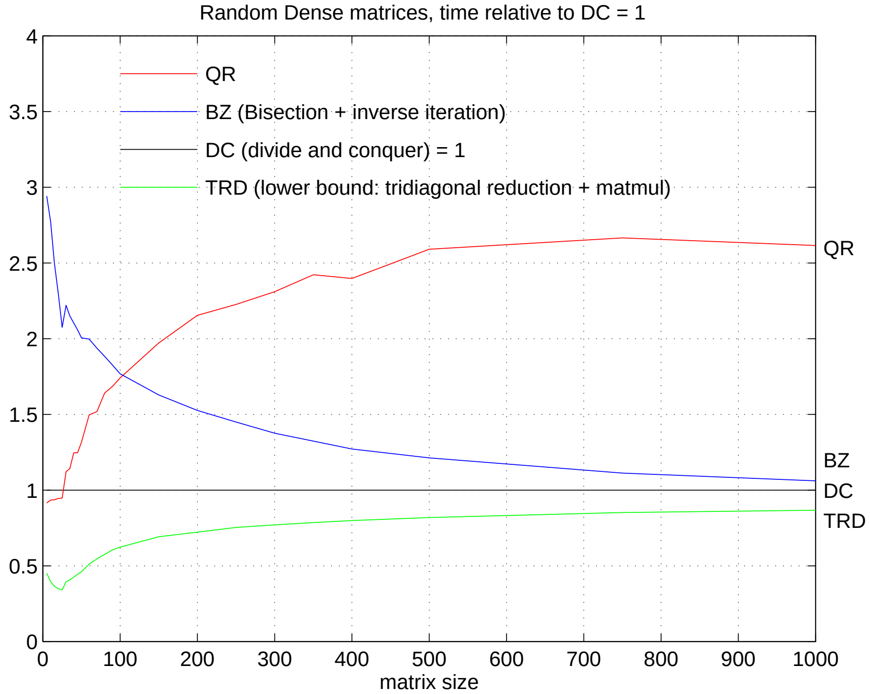
- **Can compute $\tilde{z}_i$ to high relative accuracy (but Cray...)**

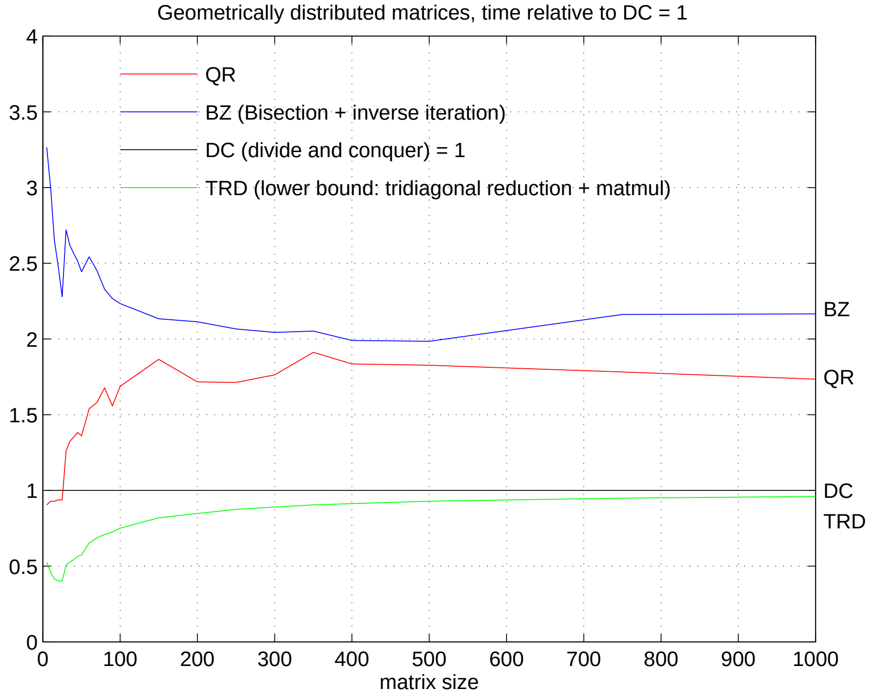
Speedups with Divide-and-conquer Tridiagonal Matrices on DEC Alpha



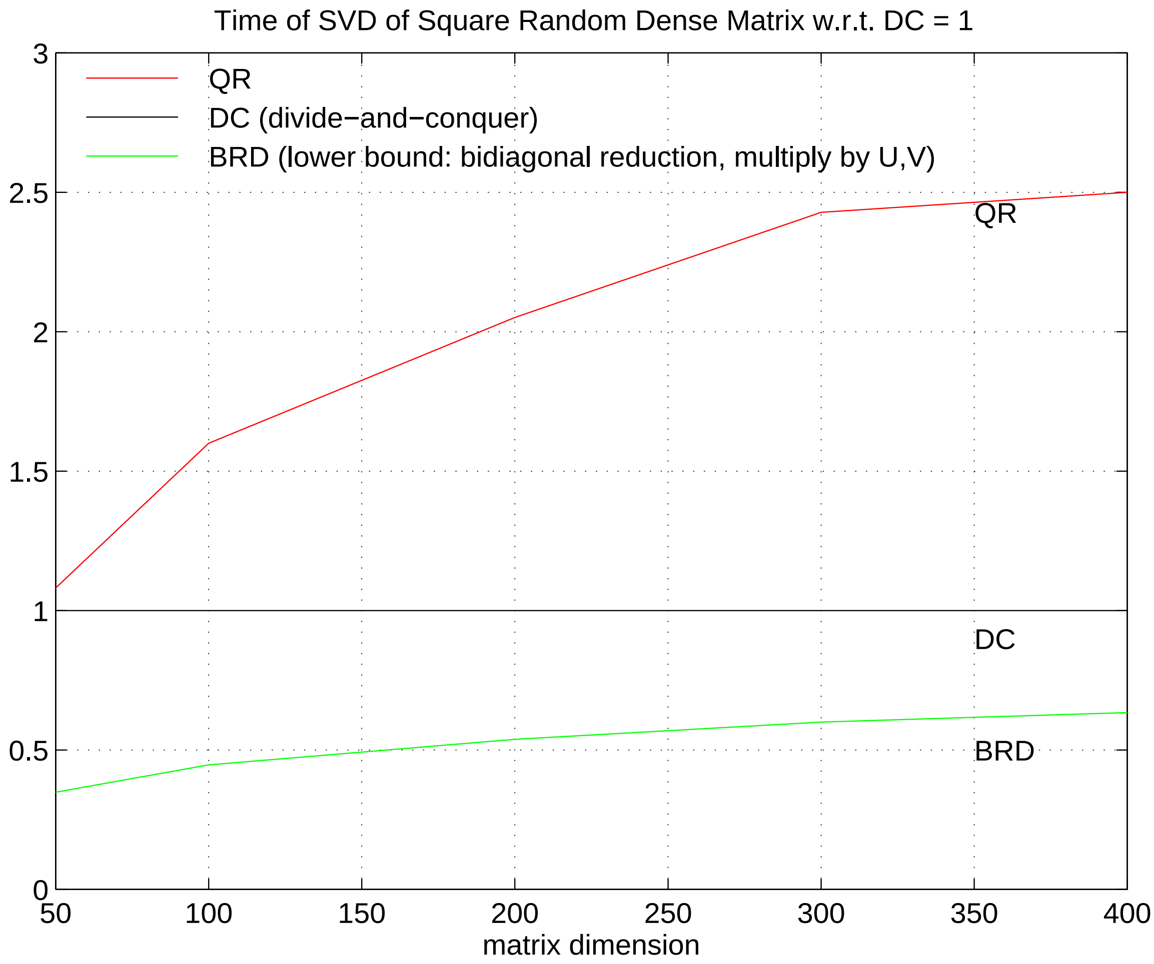


Speedups with Divide-and-conquer Dense Matrices on DEC Alpha





SVD Speedups using Divide and Conquer
Random Dense Matrices
on IBM RS6000/590



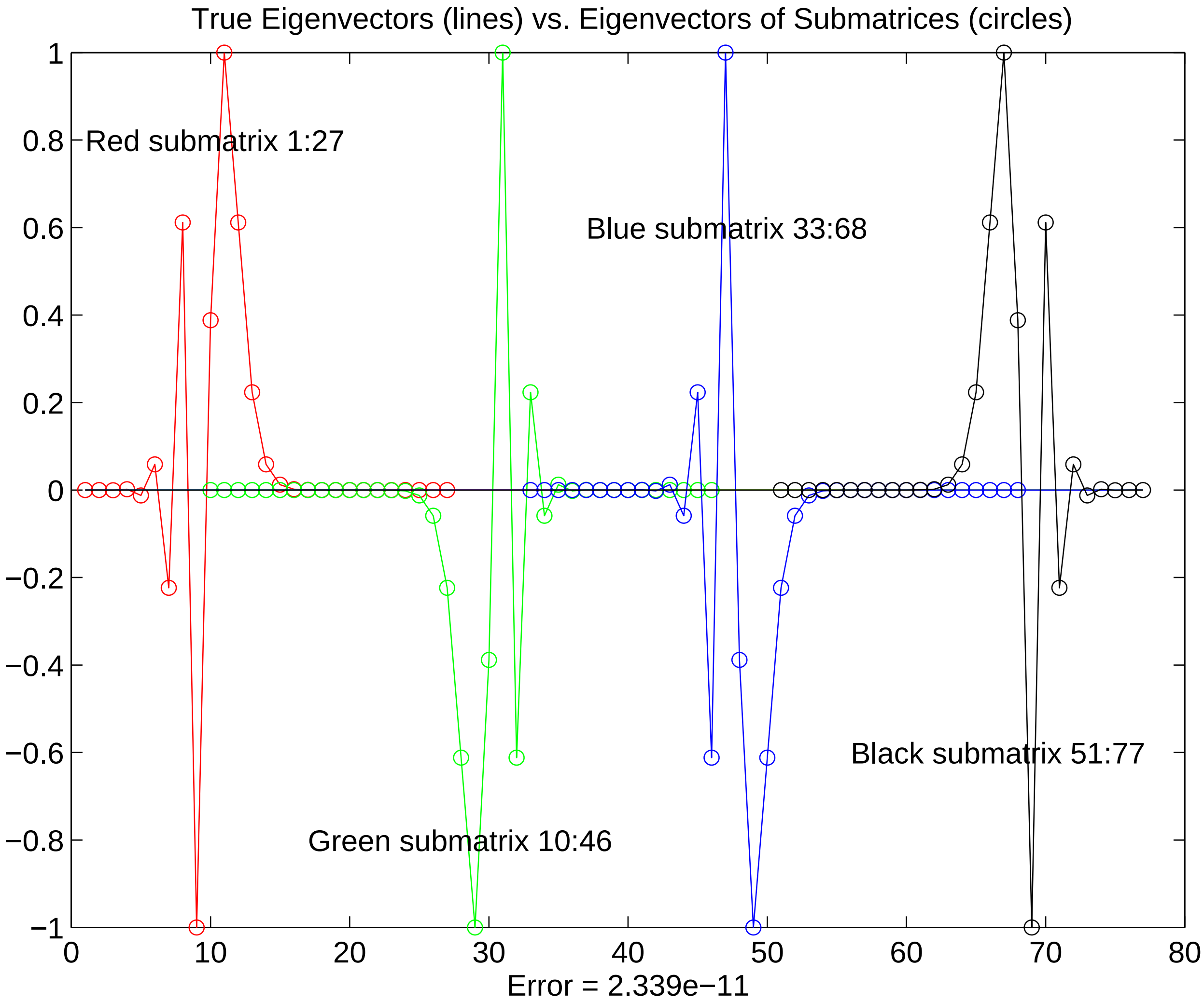
Fernando/Parlett/Dhillon

- **Holy Grail ?**
  ($O(n^2)$ and embarrassingly parallel)
- **Replacement for inverse iteration**
- **First new idea: avoid iterating**
  - **Solve $(T - \lambda I)x = e_i$ for right i**
  - **i chosen by cheap recurrence**
  - **Guarantees tiny residual if reasonable eigenvalue gap**
- **What about clustered eigenvalues?**
- **Current inverse iteration may cost $O(n^3)$**
- **Second new idea: avoid reorthogonalization**
  - **Clustered eigenvalue of T is isolated eigenvalue of submatrix $T' = T(i : j, i : j)$**
  - **Compute eigenvector of T', append zeros**
- **Also: (simulated) double precision**

Fernando/Parlett/Dhillon - Example

$$\begin{bmatrix} & 1 & & 1 & & & & & & & \cdots & & & & & & & 1 & & 1 & \\ 20 & & 19 & & \cdots & 2 & 1 & 2 & \cdots & 19 & 20 & 19 & \cdots & 2 & 1 & 2 & \cdots & & 19 & & 20 \\ & 1 & & 1 & & & & & & & \cdots & & & & & & & 1 & & 1 & \end{bmatrix}$$

- **4 eigenvalues at** 11 ± 10^{-11}



How to offer "Numerical Advice" on Eigenproblems

- **Users with different needs**
  - **Experts**
  - **Engineers and scientists generally**
  - **Teachers and Students**
- **Different kinds of resources:**
  - **Netlib, GAMS, NHSE ...**
  - **Templates ...**
  - **Libraries ...**
  - **Textbooks ...**
  - **Journal articles ...**
  - **Matlab, Maple, ...**
- **Can we satisfy all users with one "medium?"**

Yes, we can please (almost) everyone!

- All resources on last slide should be one seamless whole
- All part of a distributed E-book
- Navigated via WWW browser
- Multiple access modes:
  - Table of contents, index
  - Decision tree to choose right algorithm
  - Pointers to other trees (Netlib)
- Multiple levels of detail available
  - "Template" is pseudo-code
  - One-page mathematical description
  - Matlab, Fortan, C++ implementations
  - Detailed text
  - Annotated bibliography
- Open architecture for future growth
- Hard-nosed editing for most reliable parts
- Under construction for Eigenproblems
- Will release design soon, want input!

Future Dense Eigencodes

- **Future LAPACK releases**
  - **Divide and conquer SVD**
  - **New Least Squares Solvers**
  - **Fernando/Parlett/Dhillon**
  - **Condition estimation for** $Ax = \lambda Bx$
- **Future ScaLAPACK releases**
  - **Parallel QR (PDSYEV)**
  - **SVD based on bisection, inverse iteration; QR; Fernando/Parlett/Dhillon**
  - **More sparse solvers**
- **Eigentemplates**
  - **Draft outline nearly done (html)**
  - **Seeking community input**
  - **Templates for** $Ax = b$**: http://www.netlib.org/templates/templates.ps**