Fast Eigenvalue/Eigenvector Computation for Dense Symmetric Matrices

Inderjit S. Dhillon
Department of Computer Sciences
University of Texas, Austin

University of Illinois, Urbana-Champaign
Feb 12, 2004

joint work with
Beresford N. Parlett, UC Berkeley
Paolo Bientinesi & Robert van de Geijn, University of Texas

Problem Definition

Given

$$T = \begin{bmatrix} a_1 & b_1 & & & & \\ b_1 & a_2 & b_2 & & & \\ & b_2 & . & . & & \\ & & . & . & . & \\ & & & . & a_{n-1} & b_{n-1} \\ & & & & b_{n-1} & a_n \end{bmatrix},$$

solve

$$Tv = \lambda v$$

for λ and $v \neq 0$.

This is a **central** problem in any symmetric eigenproblem.

Requirements in finite precision :

- $\|T\hat{v}_i - \hat{\lambda}_i \hat{v}_i\| \;=\; O(\varepsilon \|T\|), \quad i = 1, 2, \ldots, n$
- $(\hat{v}_i, \hat{v}_j) \;=\; O(\varepsilon), \quad i \neq j.$

Surprises

- Dense Symmetric positive definite eigenproblem, $n = 966$.
- Example from Computational Quantum Chemistry.
- Occurs in RIMP2 theory in the modeling of biphenyl.

- Three Phases:
  - Householder Reduction ($A = QTQ^T$) 11 s.
  - Tridiagonal Solution ($T = V\Lambda V^T$) 108 s.
  - Back-Transformation ($A = (QV)\Lambda(QV)^T$) 10 s.

History

- Jacobi [1846].
- Bargmann, Montgomery and von Neumann [1946].
- Lanczos [1950], Givens [1954], Householder [1958].
- Wielandt, Wilkinson [1940/1950s], perhaps earlier.
- Francis/Kublanovskaja [1961].
- Golub and Kahan [1965].
- Wilkinson [1965].
- Golub [1973], Cuppen [1981], Gu and Eisenstat [1995].
- Kahan [1966], Demmel [1991], Eisenstat and Ipsen [1995].
- Many other publications over the last 40-50 years.
- All algorithms cost $O(n^3)$ in the worst case.

"Holy Grail" Tridiagonal Eigensolver

DESIRED GOALS:

- **Minimum output complexity** — $O(n)$ per eigenvector.
- **Provably numerically accurate**.
  - Inverse Iteration can Fail (Dhillon[1998]).
- Each eigenpair is **independently computable**.

Difficulties

- All eigenvalues of T are easily computed in $O(n^2)$ time.
- Given $\hat{\lambda}$, inverse iteration computes the eigenvector:

$$(T - \hat{\lambda} I) x_{i+1} \;=\; x_i, \quad i = 0, 1, 2, \ldots .$$

  - Costs $O(n)$ per iteration.
  - Typically, 1-3 iterations are enough.

- **BUT**, inverse iteration only guarantees

$$\|T\hat{v} - \hat{\lambda}\hat{v}\| \;=\; O(\varepsilon \|T\|).$$

Fundamental Limitations

Gap Theorem :

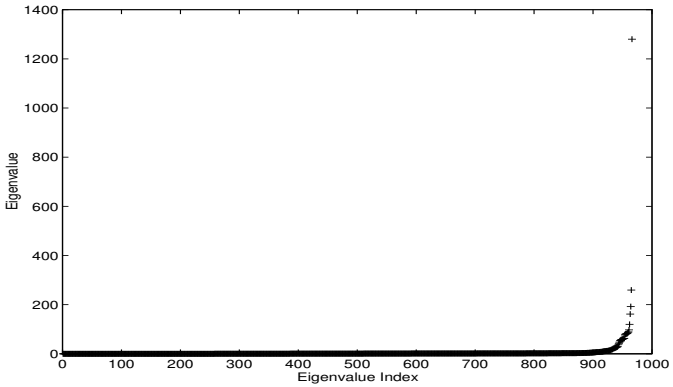
$$\sin \angle(v, \hat{v}) \;\leq\; \frac{\|T\hat{v} - \hat{\lambda}\hat{v}\|}{\mathsf{Gap}(\hat{\lambda})}.$$

$\mathsf{Gap}(\hat{\lambda})$ can be small :

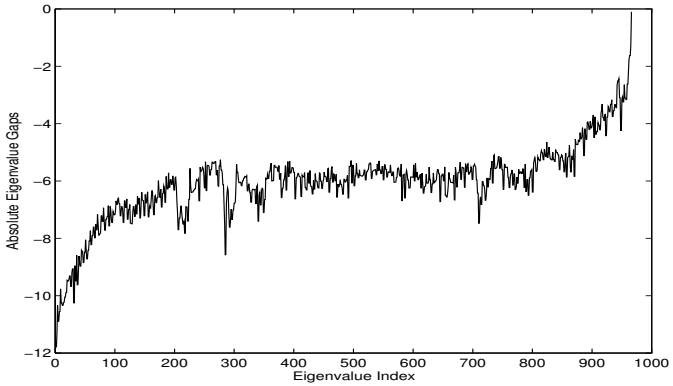
$$\begin{bmatrix} 1 & \varepsilon_1 & & \\ \varepsilon_1 & 1 & \varepsilon_2 & \\ & \varepsilon_2 & 1 & \varepsilon_3 \\ & & \varepsilon_3 & 1 \end{bmatrix}$$

When eigenvalues are close, independently computed eigenvectors **WILL NOT** be mutually orthogonal.

Eigenvalues of Biphenyl Matrix



- Plot Absgap$(i) = \log_{10}(\min(\lambda_{i+1} - \lambda_i, \lambda_i - \lambda_{i-1})/\|T\|)$:



- LAPACK — one big "cluster" $\lambda_1, \lambda_2, \ldots, \lambda_{939}$.
- Tridiagonal solution takes 80% of Total Time.

Proposed Solution

- Fundamental Limitation:

$$\sin \angle(v, \hat{v}) \ \leq \ \frac{\|T\hat{v} - \hat{\lambda}\hat{v}\|}{\mathsf{Gap}(\hat{\lambda})}.$$

- Get smallest possible residual norm.
  - Compute eigenvalue to greatest accuracy possible :

$$|\hat{\lambda} - \lambda| \ = \ O(\varepsilon|\lambda|).$$

  - Compute eigenvector to high accuracy :

$$\|T\hat{v} - \hat{\lambda}\hat{v}\| \ = \ O(\varepsilon|\lambda|).$$

- Gap Theorem implies :

$$\sin \angle(v, \hat{v}) \ = \ \frac{O(\varepsilon|\lambda|)}{\mathsf{Gap}(\hat{\lambda})} \ = \ \frac{O(\varepsilon)}{\mathsf{Relgap}(\hat{\lambda})}.$$

- **Can we achieve the above ?**

Factored Forms yield Better Representations

- Tridiagonals DO NOT determine their eigenvalues to high relative accuracy.
- Bidiagonals determine their singular values to high relative accuracy.

$$T + \mu I \quad = \quad L \quad * \quad L^T$$

$$\begin{bmatrix} x & x & & & \\ x & x & . & & \\ & . & . & . & \\ & & . & . & . \\ & & & . & x & x \\ & & & & x & x \end{bmatrix} = \begin{bmatrix} x & & & & \\ x & x & & & \\ & x & . & & \\ & & . & . & \\ & & & . & x & \\ & & & & x & x \end{bmatrix} * \begin{bmatrix} x & x & & & \\ & x & x & & \\ & & . & . & \\ & & & . & . & \\ & & & & x & x \\ & & & & & x \end{bmatrix}$$

- Bidiagonal Factors are "better" since they allow us to
  - compute eigenvalues to high accuracy,
  - compute eigenvectors to high accuracy.
- We call the bidiagonal factors: **Relatively Robust Representations (RRRs)**.
- High accuracy $\Rightarrow$ **Orthogonality**.

Algorithm Outline

1. Choose μ such that $T + \mu I$ is positive definite.

2. Compute the factorization :

$$T + \mu I \;=\; LDL^T.$$

3. Compute eigenvalues of LDL^T to high relative accuracy (by *dqds* or *bisection*).

4. Given eigenvalues, compute accurate eigenvectors of LDL^T.

   — **HOW?**

Differential Transformations

- Inverse iteration — Solve for z :

$$LDL^T - \hat{\lambda} I \;=\; L_+ D_+ L_+^T.$$
$$L_+ D_+ L_+^T \, z \;=\; \text{random vector.}$$

Simple qd :

$D_+(1) \;:=\; d_1 - \hat{\lambda}$
for $i = 1, n-1$
  $L_+(i) \;:=\; (d_i l_i)/D_+(i)$
  $D_+(i+1) \;:=\; d_i l_i^2 + d_{i+1} - L_+(i) d_i l_i - \hat{\lambda}$
end for

Differential qd :

$s_1 \;:=\; -\hat{\lambda}$
for $i = 1, n-1$
  $D_+(i) \;:=\; s_i + d_i$
  $L_+(i) \;:=\; (d_i l_i)/D_+(i)$
  $s_{i+1} \;:=\; L_+(i) l_i s_i - \hat{\lambda}$
end for
$D_+(n) \;:=\; s_n + d_n$

Computing an Eigenvector

- Compute the appropriate Twisted Factorization :

$$T - \hat{\lambda} I \;=\; N_r D_r N_r^T,$$

where D_r is diagonal, and

$$N_r \;=\; \begin{bmatrix} \mathsf{x} & & & & & & \\ \mathsf{x} & \mathsf{x} & & & & & \\ & \cdot & \cdot & & & & \\ & & \mathsf{x} & \textcircled{\gamma_r} & \mathsf{x} & & \\ & & & & \cdot & \cdot & \\ & & & & & \mathsf{x} & \mathsf{x} \\ & & & & & & \mathsf{x} \end{bmatrix}$$

and r is the index of the twist.

- Solve for z, $N_r D_r N_r^T z = \gamma_r e_r$ ($\Rightarrow N_r^T z = e_r$) :

$$z(i) \;=\; \begin{cases} \;\;1, & i = r, \\ -L_+(i) \cdot z(i+1), & i = r-1, \ldots, 1, \\ -U_-(i-1) \cdot z(i-1), & i = r+1, \ldots, n. \end{cases}$$

- Solves an open problem posed by Wilkinson (1965).

Main Theorem

THEOREM. [Dhillon & Parlett, 2003] Eigenvectors computed by Algorithm **Getvec** are numerically orthogonal if eigenvalues of LDL^T have large relative gaps. In particular,

$$(\hat{v}_i, \hat{v}_j) \;=\; \frac{O(\varepsilon)}{\mathsf{Relsep}(\lambda_i, \lambda_j)},$$

where

$$\mathsf{Relsep}(\lambda_i, \lambda_j) \;=\; \frac{|\lambda_i - \lambda_j|}{\max(|\lambda_i|, |\lambda_j|)}.$$

- Example of Large Relsep :

$$\lambda_1 = 10^{-16},\ \lambda_2 = 10^{-15} \quad \Rightarrow \quad \mathsf{Relsep}(\lambda_1, \lambda_2) \approx 1$$

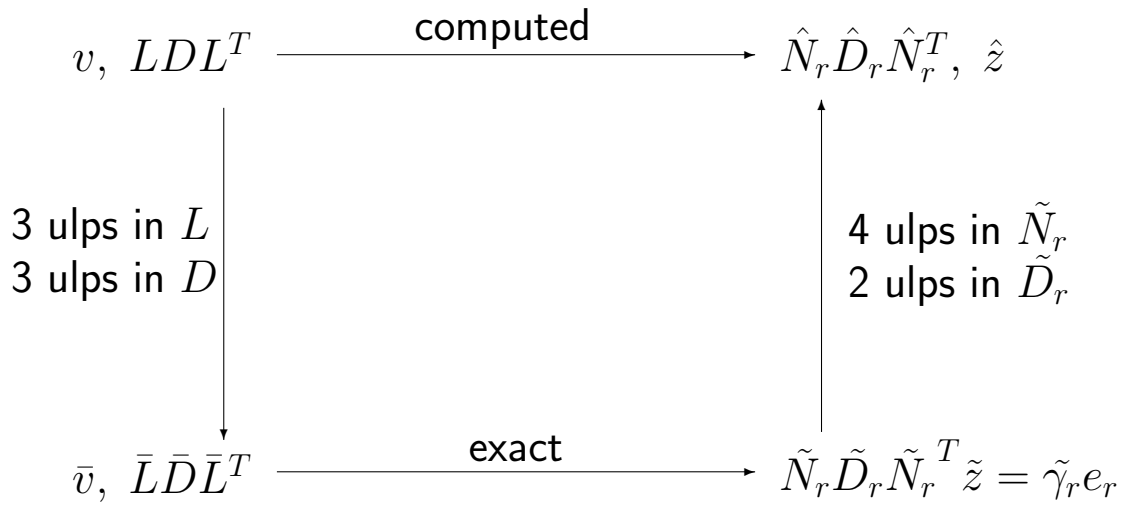
Above Theorem $\Rightarrow$ Automatic Orthogonality.

- Example of Small Relsep :

$$\begin{aligned} \lambda_1 &= 1.000000000000000\mathbf{1}, \\ \lambda_2 &= 1.000000000000000\mathbf{2}. \end{aligned}$$

Proof of Correctness

- Desired Relationship: $LDL^T - \hat{\lambda} I = N_r D_r N_r^T$, and $N_r D_r N_r^T z = \gamma_r e_r$.



- Exact Mathematical relationship holds : $\bar{L}\bar{D}\bar{L}^T - \hat{\lambda} I \ = \ \tilde{N}_r \tilde{D}_r \tilde{N}_r{}^T$.

- **Key step in proof** is to relate $\hat{z}$ to v in 3 steps :
  1. $\hat{z}$ is close to $\tilde{z}$, (only multiplications),
  2. $\sin \angle(\bar{v}, \tilde{z}) = O(\varepsilon|\bar{\lambda}|)/\mathsf{gap}(\hat{\lambda})$, $(|\tilde{\gamma}_r| = O(\varepsilon|\bar{\lambda}|))$,
  3. $\sin \angle(\bar{v}, v) = O(\varepsilon)/\mathsf{relgap}(\hat{\lambda})$ (relative perturbation theory).

$$\Rightarrow \quad \sin \angle(\hat{z}, v) \ = \ \frac{O(\varepsilon)}{\mathsf{Relgap}(\hat{\lambda})}.$$

Algorithm MR^3 (Multiple RRRs)

1. Choose μ such that $T + \mu I$ is positive definite.

2. Compute the factorization :
$$T + \mu I \;=\; LDL^T.$$

3. Compute eigenvalues of LDL^T to high relative accuracy (by *dqds* or *bisection*).

4. Group eigenvalues according to their **Relative Gaps** :

   a) isolated (agree in $<$ 3 digits). Compute eigenvector using a twisted factorization.

   b) clustered (agree in $>$ 3 digits).

   - Pick μ near cluster to form $LDL^T - \mu I = L_1 D_1 L_1^T$.
   - "Refine" eigenvalues in cluster to high relative accuracy.
   - Set $L \leftarrow L_1$, $D \leftarrow D_1$. Repeat step 4 for eigenvalues in cluster.

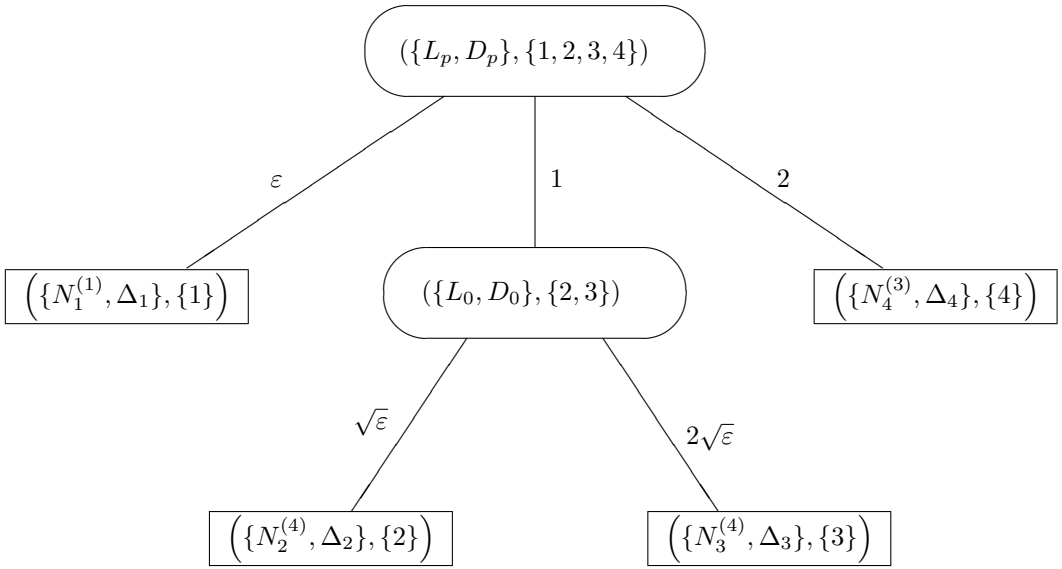
Example 1

- Eigenvalues: ε, $1+\sqrt{\varepsilon}$, $1+2\sqrt{\varepsilon}$, 2.

- Extra representation needed at $\sigma = 1$:

$$L_p D_p L_p^T - I \;=\; L_0 D_0 L_0^T.$$

- The following **Representation Tree** captures the steps of the algorithm:



Wilkinson's Matrix

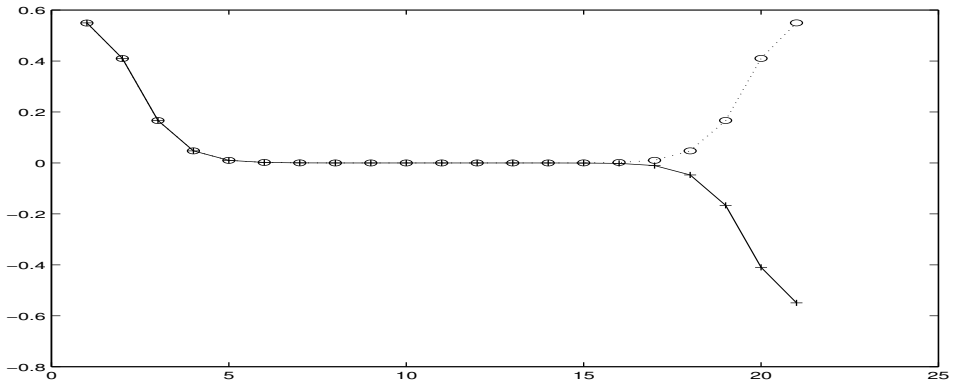
- W_{21}^{+} : 21×21 Wilkinson's matrix.
- λ_{20} and λ_{21} are identical to working precision.
- What happens in this case?

$$L_p D_p L_p^T - \hat{\lambda}_{21} I \;=\; L_0 D_0 L_0^T.$$

- Roundoff comes to the rescue.

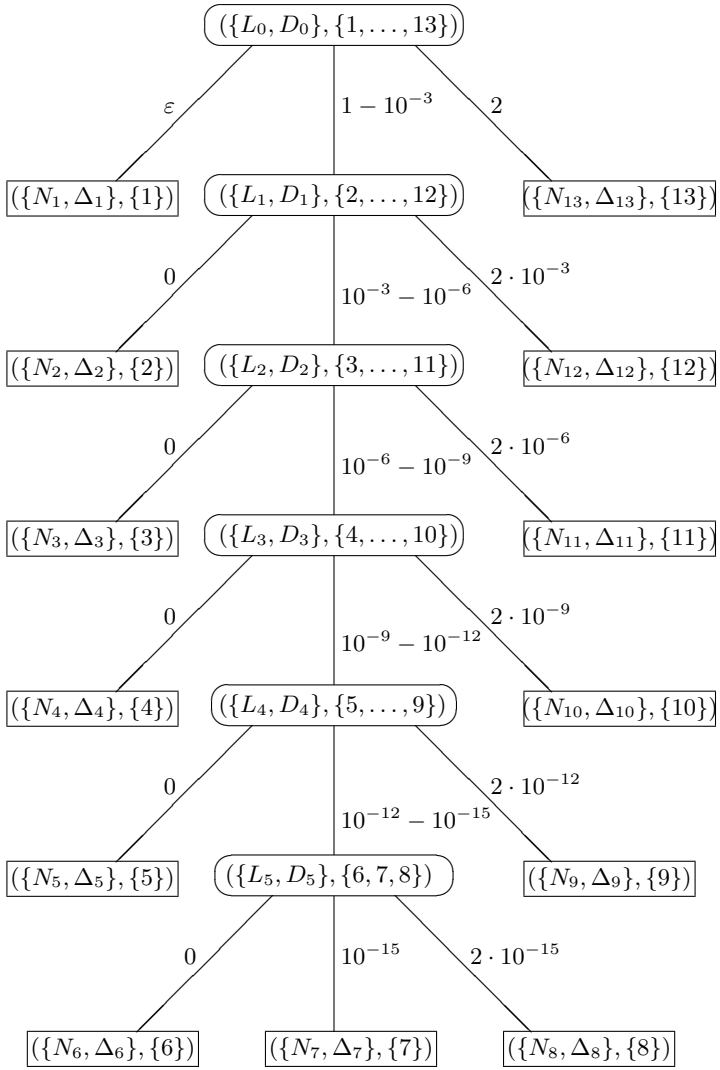
$$\begin{aligned} \lambda_{20}(L_0 D_0 L_0^T) \;\;\&\;\; \lambda_{21}(L_0 D_0 L_0^T) \;\; &\text{— no digits in common!} \\ -7.28 \times 10^{-14} \;\;\&\;\; -1.22 \times 10^{-15} & \\ (\hat{v}_{20}, \hat{v}_{21}) \;=\; 1.0 \times 10^{-16} & \end{aligned}$$

- Computed Eigenvectors $\hat{v}_{20}$ and $\hat{v}_{21}$:



Large Depth

13×13 matrix with eigenvalues: $0, 1, 1 \pm 10^{-15}, 1 \pm 10^{-12}, 1 \pm 10^{-9}, 1 \pm 10^{-6}, 1 \pm 10^{-3}, 2$.



Timing Results of Latest Code

Some Timings :

On a 1687×1687 $\mathrm{SiOSi_6}$ quantum chemistry matrix,

- Time (Algorithm $\mathbf{MR}^3$) $= 5.5$ s.
- Time (LAPACK bisection + inverse iteration) $= 310$ s.
- Time (EISPACK bisection + inverse iteration) $= 126$ s.
- Time (LAPACK QR) $= 1428$ s.
- Time (LAPACK Divide & Conquer) $= 81$ s.

On a 2000×2000 [1,2,1] matrix,

- Time (Algorithm $\mathbf{MR}^3$) $= 4.1$ s.
- Time (LAPACK bisection + inverse iteration) $= 808$ s.
- Time (EISPACK bisection + inverse iteration) $= 126$ s.
- Time (LAPACK QR) $= 1642$ s.
- Time (LAPACK Divide & Conquer) $= 106$ s.

A Parallel Eigensolver for Dense Symmetric Matrices using Multiple Relatively Robust Representations [PMR^3]

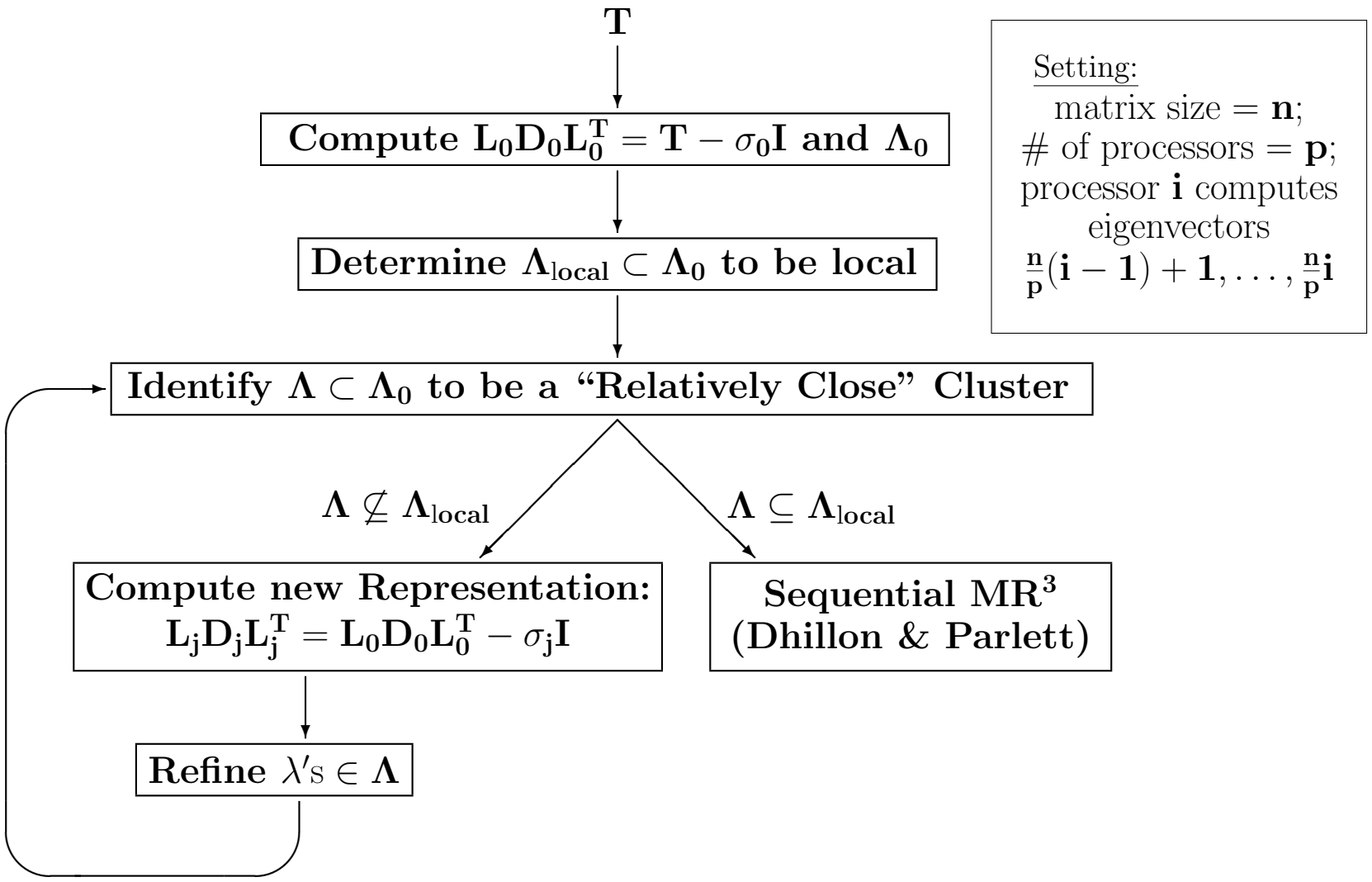
Paolo Bientinesi
Inderjit S. Dhillon
Robert A. van de Geijn

Department of Computer Sciences (CS)
Institute for Computational Engineering and Sciences (ICES)
The University of Texas at Austin

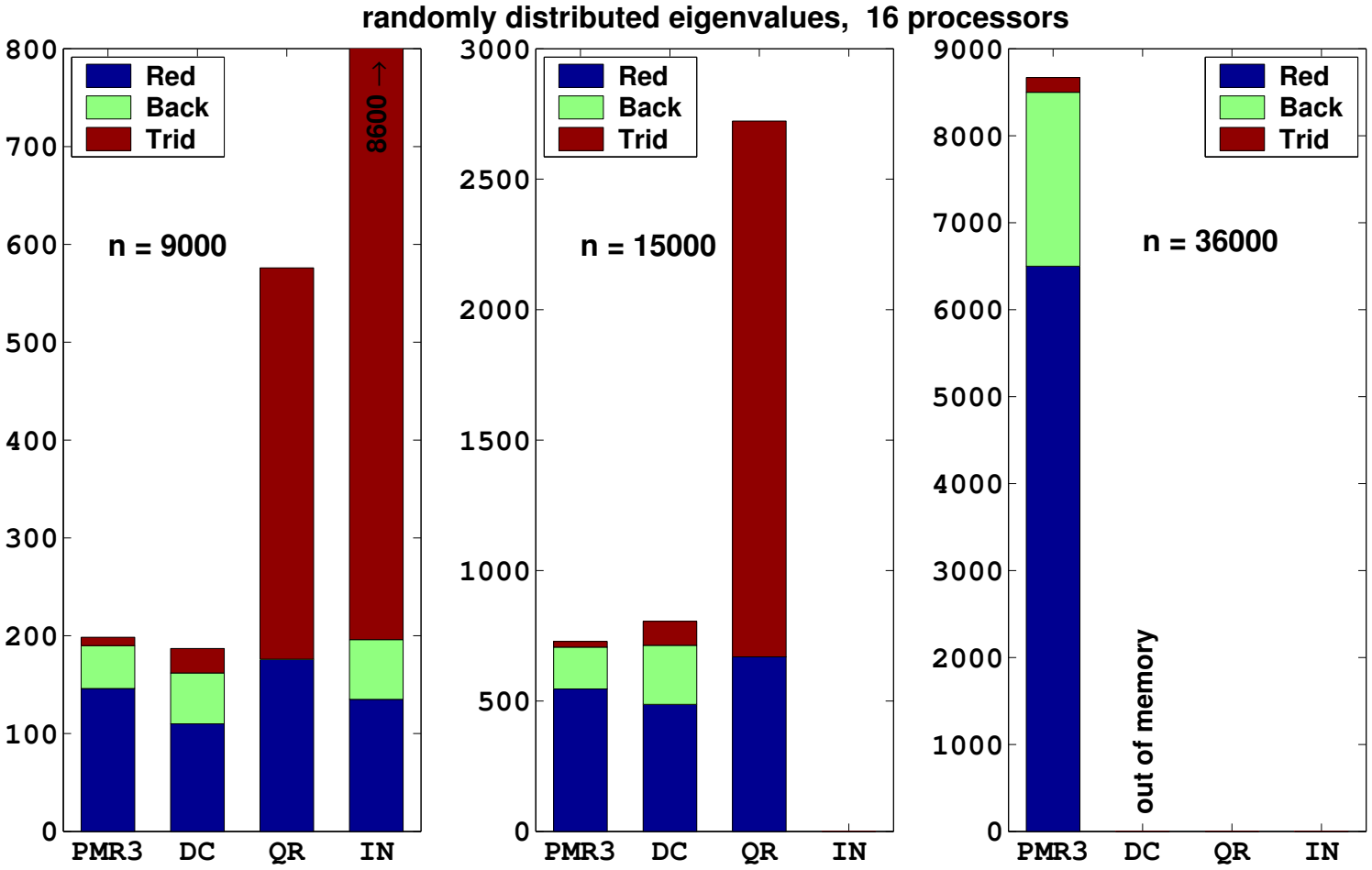
{pauldj,inderjit,rvdg}@cs.utexas.edu

More Information soon at: www.cs.utexas.edu/users/plapack/

Flowchart for PMR^3 on each processor

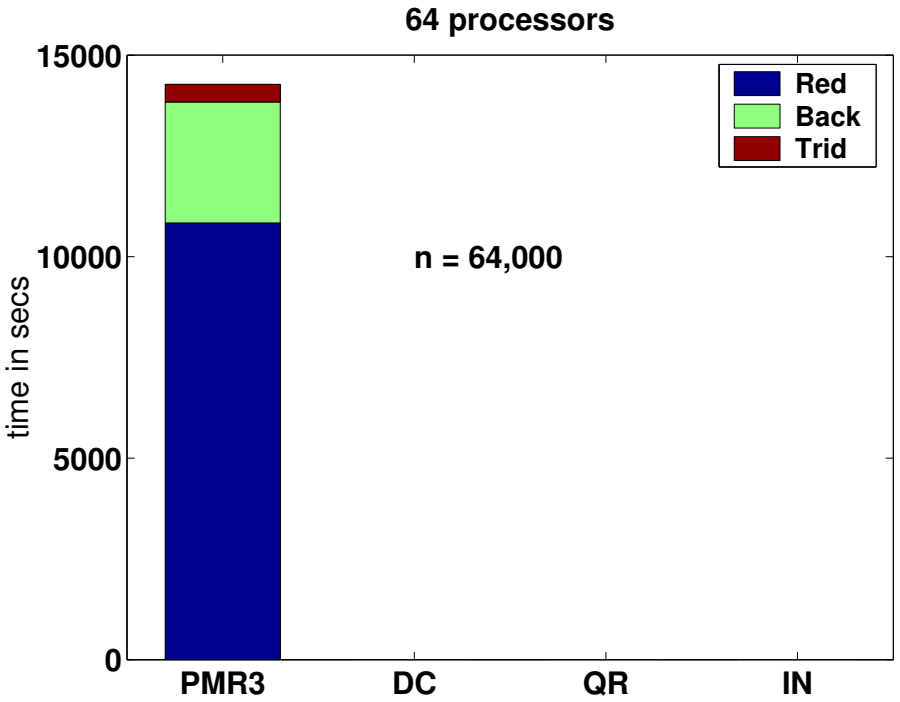


Running Time Comparisons (in secs) for dense symmetric matrices.



- PMR^3 and QR use PLAPACK for Reduction and Backtransformation stages.
- DC is ScaLAPACK's PDSYEVD, and IN is ScaLAPACK's PDSYEVX.
- Leftmost plot eliminates Inverse Iteration(IN), middle plot eliminates QR, rightmost plot eliminates DC for large matrices.

Largest Problem Solved on 64 processors



Reduction time	=	**11,000 secs**
Tridiagonal Eigsolver	=	**440 secs**
(Eigenvalue computation	=	**390 secs)**
Backtransformation	=	**3,000 secs**

Experiments run on the Buffalo CCR linux cluster: 300 nodes each equipped with 2Gb of memory and two Intel Pentium 4 processors, connected via Myricom high performance network (2Gigabit/s).

Access to parallel systems was provided by TACC, NPACI, and the Center for Computational Research of the University at Buffalo, SUNY.

CONCLUSIONS:

- **PMR^3 scales up well both in time and in space ($O(n)$ workspace).**
- **n=64,000 on 64 processors:**

$$\frac{\text{Extrapolated Time(QR Trid)}}{\text{Time(PMR}^3\text{ Trid)}} = \frac{\text{10 hours}}{\text{8 minutes}} = \mathbf{75}$$

$$\frac{\text{Extrapolated Time(Dense Eigensolve by QR)}}{\text{Time(Dense Eigensolve by PMR}^3\text{)}} = \frac{\text{12.5 hours}}{\text{4 hours}} = \mathbf{3.1}$$

- **Problems of very large size can now be tackled in a reasonable amount of time.**

Summary

- New Algorithm $\mathbf{MR}^3$ for the Real Symmetric Tridiagonal Eigenproblem
- **High ACCURACY results in SPEED.**
- Faster $O(n^2)$ serial solution.
- Preliminary version of software is available as part of LAPACK.
- Software update will be available in next LAPACK release.
  - Send email to **inderjit@cs.utexas.edu** for latest version of the software.
- Good parallel scaling.
- Parallel implementation in PLAPACK.
- Papers available at : **http://www.cs.utexas.edu/users/inderjit**

Largest Problem Solved on 1 Processor

- Symmetric Tridiagonal with $n = 13,786$ from Jeff Bennighof (Aerospace Engg, UT Austin).
- Arises from Finite Element Model of an automobile body.
- Timing Results on a Sun Enterprise 450 with 4 400 MHz processors and 4GB memory:

Time (Algorithm $\mathbf{MR}^3$)	4 min 12 s
Time (LAPACK Divide & Conquer)	1 hr 26 min
Time (EISPACK bisection & invit)	22 hrs 2 min
Time (LAPACK bisection & invit)	67 hrs 32 min
Time (LAPACK QR)	84 hrs 42 min
Time spent in new algorithm	141s (eigs), 107s (evecs)
Maximum Residual Norm	10^{-17}
Maximum Dot Product	10^{-11}
Percentage of Zeros in Eigenvector Matrix	31%