Superfast Divide-and-Conquer Eigensolvers

Jimmy Vogel\textsuperscript{1} Jianlin Xia\textsuperscript{1}

\textsuperscript{1}Department of Mathematics, Purdue University

August 29, 2015
Durham, NC
Multi-resolution Interactions Workshop
Strengths of Algorithms

// Speed and Complexity
- FLOP’s and runtime of eigendecomposition near-linear
- FLOP’s and runtime of eigenvector matvec near-linear
- storage for eigenvector matrix near-linear

// Accuracy and Stability
- can analytically prove stability
- can prove accuracy to $|\tau \log n|$ ($\tau$ HSS aprox. accuracy)
- even stronger results in several special cases

// Robustness
- eigensolver: eigenvalues need not be distinct or well-separated
- SVD: matrix need not by symmetric or even square
- “black box” code: do not a priori need rank structure
  (but there is some computational savings if you have a guess)
- options: full SVD, thin SVD, just spectrum, or just subset
  (computational savings if less information requested)
Overview

// Introduction

// Superfast divide-and-conquer (DC) [Vogel, Xia, et al., 2015]
- Algorithm
- Structured perturbation analysis

// Extensions of algorithm
- Case of clustered eigenvalues and high multiplicities
- SVD for non-symmetric and non-square matrices
- Some preliminary applications: PDEs and Optimization

// Conclusion
- Numerical tests
- Future directions
Previous work on rank-structured Eigensolvers

// QR iterations for semiseparable (off-diagonal rank 1) matrices
[Dewilde, Van Dooren, 1983], [Gemignani, Gohberg, Mastronardi, Vandebril, Van Barel, et al., 2002 - ]

// Companion (polynomial roots), semiseparable with
off-diagonal rank 1 \(O(n^2)\) complexity
[Pan, et al.]

// Toeplitz, Hankel \(O(n^2)\) complexity or higher
[Kailath, et al.]

// Symmetric tridiagonal, block diagonal plus semiseparable
\(\approx O(n)\) complexity [Chandrasekaran, Eisenstat, Gu]
Let \( T \in \mathbb{R}^{n \times n} \) with \( n \geq 2 \) be symmetric and tridiagonal.

Suppose the spectral decompositions
\[
\hat{T}_1 = Q_1 D_1 Q_1^T \quad \text{and} \quad \hat{T}_2 = Q_2 D_2 Q_2^T
\]
have been computed.

Then
\[
T = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \left( \begin{bmatrix} D_1 & \beta \\ \beta & D_2 \end{bmatrix} + \beta z z^T \right) \begin{bmatrix} Q_1^T \\ Q_2^T \end{bmatrix}
\]

where \( z^T = [q_1^T, q_2^T] \), \( q_1^T \) is the last row of \( Q_1 \), and \( q_2^T \) is the first row of \( Q_2 \).
Introduction

Superfast divide-and-conquer (DC) [Vogel, Xia, et al., 2015]
- Algorithm
- Structured perturbation analysis

Extensions of algorithm
- Case of clustered eigenvalues and high multiplicities
- SVD for non-symmetric and non-square matrices
- Some preliminary applications: PDEs and Optimization

Conclusion
- Numerical tests
- Future directions
HSS divide-and-conquer: Construction

// Explicit construction - banded or block-tridiagonal

\[ D_i = \begin{pmatrix}
  A_{j-1,j-1} & A_{j-1,j} & & \\
  A_{j,j-1} & A_{j,j} & & \\
  & A_{j+1,j} & A_{j+1,j+1} & \\
  & & A_{j+1,j+2} & 0
\end{pmatrix}, B_{c_1} = \begin{pmatrix}
  0 & 0 \\
  A_{j+1,j+2} & 0
\end{pmatrix} \]

// SVD or RRQR based direct construction

// Fast multipole method (FMM) [Greengard, Rokhlin] based

// Randomized, matrix-vector product based
- Stable, adaptive, parallel, and matrix-free [Xi, Xia, et al.]
- Thm. [Xi, Xia] with structured generators, HSS approx. error:
  \[ \|E\|_F = O(\tau r^{\log_2 n}) \|A\|_F \]  (\(r\): max off-diagonal rank)
HSS divide-and-conquer: Dividing

\[ D_i = \begin{pmatrix} D_{c1} & D_{c2} \\ D_{c2} & \end{pmatrix} + \begin{pmatrix} U_{c1} & U_{c2} \\ U_{c1} & U_{c2} \end{pmatrix} \begin{pmatrix} B_{c1}^T & B_{c1} \\ B_{c1} & \end{pmatrix} \begin{pmatrix} U_{c1}^T \\ U_{c2}^T \end{pmatrix} \]

\text{rank-2r update}

\[ = \begin{pmatrix} D_{c1} - U_{c1} U_{c1}^T & D_{c2} - U_{c2} B_{c1}^T B_{c1} U_{c2}^T \\ D_{c2} - U_{c2} B_{c1}^T B_{c1} U_{c2}^T & \end{pmatrix} + \begin{pmatrix} U_{c1} & U_{c2} B_{c1}^T \\ U_{c2} B_{c1}^T & \end{pmatrix} \begin{pmatrix} U_{c1} \\ U_{c2} B_{c1}^T \end{pmatrix} \]

\[ \overset{\text{rank-r update}}{\Rightarrow} \begin{pmatrix} \tilde{D}_{c1} \\ \tilde{D}_{c2} \end{pmatrix} = ZZ^T \]

Superfast Divide-and-Conquer
HSS divide-and-conquer: Dividing

// HSS form of $A - U_k H U_k^T$

\[ i \rightarrow k_l \rightarrow ... \rightarrow k_1 \rightarrow k : \text{path connecting } i \text{ to root} \]
\[ \hat{U}_i = U_i, \quad \hat{R}_i = R_i, \]
\[ D_i \leftarrow D_i - U_i R_i R_{k_1} ... R_{k_l} H R_{k_l}^T ... R_{k_1}^T R_i^T U_i^T \]
\[ B_i \leftarrow B_i - R_i R_{k_1} ... R_{k_l} H R_{k_l}^T ... R_{k_1}^T R_{\text{sib}(i)}^T \]

diagonal structure preserved during recursive dividing

// Lower level $D, B$ generators updated multiple times

computation reuse
**Stage 3:** Conquer, **Bottom-Up**, $O(r^2 n \log_2 n) + O(rn \log_2^2 n)$

for $i = \log n : -1 : 1$

for $j = 1 : r$

for $k = 1 : 2^k$

1) apply previous eigenvector matrix to rank-1 update
2) solve secular eq. to obtain eigenvalues of $D + ww^T$
3) solve diag. system to obtain $D + ww^T$ eigenvectors
4) normalize eigenvectors of $D + ww^T$
HSS divide-and-conquer: Conquering

We circumvent costly matvecs by noting the special structure of the local eigenvector matrices $Q_i$. 

$$Q_{i,j} = \frac{v_i n_j}{\lambda_j - \Lambda_{i,i}}, \text{ where } n_j \text{ is a normalization constant}$$ (1)

Thus we have an explicit formula for each component of $\hat{z} = Q^T z$. 

$$\hat{z}_i = n_i \sum_{j=1}^{n} \frac{z_j v_j}{\lambda_i - \Lambda_{j,j}}$$ (2)

This is of the form $\Phi(\xi) = \sum_{j=1}^{n} c_j \varphi(\xi - \xi_j)$, where $\varphi(x)$ is either $\log(x)$, $1/x$, or $1/x^2$. Thus it can be evaluated at $m$ points in $O(m + n)$ complexity by the fast multipole method (FMM).

Similarly, the FMM can be used to accelerate:

- rootfinding
- normalizations
- inverse eigenvalue problem
Eigenvector Matrix Structure

// Structures within local eigenmatrices

**Lemma.** $Q_i^{(1)}$ has max off-diag numerical rank $O(\log k)$. ($k$: number of distinct eigenvalues)

For distinct $\tilde{\lambda}_i$, $\lambda_j Q_i^{(1)} = \left( \frac{\hat{v}_i s_j}{\tilde{\lambda}_i - \lambda_j} \right)_{i,j}$ (Cauchy-like)

For repeated $\tilde{\lambda}_i$, $Q_i^{(1)}$ is Householder (off-diag rank $\leq 1$)
(Deflation as usual [Dongarra, Sorenson], [Gu, Eisenstat])

// Global eigenmatrix $Q$

**Theorem.** The off-diagonal numerical rank is bounded by $O(r \log^2 n)$. 
Algorithmic Complexity

// Banded matrix (finite bandwidth):

Eigendecomposition \( O(n \log^2 n) \)
Eigematrix-vector product \( O(n \log n) \)
Storage \( O(n \log n) \)

// Toeplitz matrix:

Eigendecomposition \( O(n \log^3 n) \)
Eigematrix-vector product \( O(n \log^2 n) \)
Storage \( O(n \log^2 n) \)

// General symmetric HSS matrix:

Eigendecomposition \( O(r^2 n \log n) + O(rn \log^2 n) \)
Eigematrix-vector product \( O(rn \log n) \)
Storage \( O(rn \log n) \)
Theorem. With $A_{ij} = U_i B_i V_j^T + E$, $\|E\|_2 \leq \tau$ and $l$ levels,

$$\|E\|_2 = \|A - \tilde{A} \text{ (HSS)}\|_2 \leq l\tau \quad \text{(attainable)}, \quad |\lambda_i - \tilde{\lambda}_i| \leq l\tau \quad (3)$$

To attain the error bound:

$$E^{(l)} \approx \sum_{\tilde{i}=1}^{l} \text{diag} \left( \begin{pmatrix} 0 & \tau \tilde{l} \\ \tau \tilde{l} & 0 \end{pmatrix} \right), \quad i : \text{all nodes at level } \tilde{i}$$

Corollary. (Based on [Ipsen, 2009], for well-separated eigenvalues)

If $\lambda_i - \lambda_{i+1} > 2l\tau$, $\lambda_{i-1} - \lambda_i > 2l\tau$, then:

$$|\lambda_i - \tilde{\lambda}_i| \leq \|E q_i\|_2$$
Structured Perturbation Analysis

Consider $A = (A_{ij})_{4 \times 4}$

\[
\|A_{i,i+1}\|_2 \leq \tau_1, \quad i = 1, 2, 3,
\]
\[
\|A_{i,i+2}\|_2 \leq \tau_2 \ll \tau_1, \quad i = 1, 2,
\]
\[
\|A_{i,i+3}\|_2 \leq \tau_3 \ll \tau_2, \quad i = 1.
\]

\[
\delta_1 = \min_{\tilde{\lambda} \in \lambda(A_{1:2,1:2})} |\lambda - \tilde{\lambda}|, \quad \delta_2 = \min_{\tilde{\lambda} \in \lambda(A_{2:4,2:4})} |\lambda - \tilde{\lambda}|, \quad \delta_3 = \min_{\tilde{\lambda} \in \lambda(A_{3:4,3:4})} |\lambda - \tilde{\lambda}|
\]

Error diminishing effect (similar to [Paige])

\[
\delta_1 \leq \frac{1}{\delta_3} \tau_1 \left( \tau_2 + \frac{1}{\delta_2} \tau_1^2 \right)
\]

Superfast Divide-and-Conquer
// Introduction

// Superfast divide-and-conquer (DC) [Vogel, Xia, et al., 2015]
- Algorithm
- Structured perturbation analysis

// Extensions of algorithm
- Case of clustered eigenvalues and high multiplicities
- SVD for non-symmetric and non-square matrices
- Some preliminary applications: PDEs and Optimization

// Conclusion
- Numerical tests
- Future directions
Clustered Eigenvalues (joint work with Xiaobai Sun)

Why do clusters pose a problem?

// condition number of evaluating secular equation diverges

$$\kappa_2 = \left( \frac{2}{u_i - l_i} \right) \left( \sum_i \frac{2v_i^2}{(u_i - l_i)^2} \right) \left( 1 + \sum_i \frac{2v_i^2}{u_i - l_i} \right)^{-1}$$

// loss of orthogonality of eigenvectors [Dongarra, et al.]

// deflation becomes much more involved

Solution: find a nearby problem with nicer properties

// initial HSS approximation (i.e. find a matrix whose eigenvector matrix has nice rank properties)

// use inverse eigenvalue problem at each node so nearby problem has orthogonal eigenvectors [Gu, Eisenstat, 1995]

// use aggressive deflation by looking at individual eigenspaces to find nearby problem with higher multiplicity at each node

Superfast Divide-and-Conquer
Clustered Eigenvalues (joint work with Xiaobai Sun)

Standard bounds for eigenvalue perturbation:

\[ \lambda_i(A) \leq \lambda_i(A + vv^T) \leq \lambda_{i+1}(A) \]

Bounds with considerations made for individual eigenspaces [Ipsen, 2009]:

\[ \lambda_i(A) + l_i \leq l_i(A + vv^T) \leq \min\{\lambda_i(A) + u_i, \lambda_{i-1}\} \]

where \( u_i = \frac{1}{2} \left( \|y_i:n\|^2 - \text{gap}_{i+1} + \sqrt{(\text{gap}_{i+1} + \|y_i:n\|^2)^2 - 4\text{gap}_{i+1}\|y_{i+1:n}\|^2} \right) \)

and \( l_i = \frac{1}{2} \left( \text{gap}_i + \|y_{1:i}\|^2 - \sqrt{(\text{gap}_i + \|y_{1:i}\|^2)^2 - 4\text{gap}_i\|y_i\|^2} \right) \)
Important special cases:

// shift to within $\tau$ of next pole (most common behavior in clusters)
$$\frac{1}{2} \left( \text{gap}_i - \|y_{1:i}\|^2 + \sqrt{\left(\text{gap}_i + \|y_{i:n}\|^2\right)^2 - 4\text{gap}_i|y_i|^2} \right) \leq \tau \quad (4)$$

// negligible contribution in all eigenspaces to the left
$$I_i = \frac{1}{2} \left( \text{gap}_i + \|y_{1:i}\|^2 - \sqrt{\left(\text{gap}_i + \|y_{i:n}\|^2\right)^2 - 4\text{gap}_i|y_i|^2} \right) \leq \tau \quad (5)$$
Clustered Eigenvalues (joint work with Xiaobai Sun)

Several other inequities that allow us to aggressively deflate

Other situations:

Even when the eigenvalues are not quite clustered, examining individual eigenspaces still helps us:

// improves convergence of rootfinding subroutine
// very cheap to compute relevant coefficients to test
// prevents evaluation of secular equation near poles
Important observation:

HSS matrices can have off-diagonal blocks with ranks \( \geq 100 \). With this in mind, the structure of clusters can shift dramatically from level to level. **Clustered eigenvalues should be treated as a single eigenvalue of multiplicity whenever possible, but this should always be done local to each node; never globally.**

Example:

\[
A = \begin{pmatrix}
2.8909 & 0.9339 & 0.5835 & 0.9330 & 0.8795 & 0.7543 \\
0.9339 & 4.8198 & 0.6602 & 0.8195 & 1.0276 & 0.6892 \\
0.5835 & 0.6602 & 3.3969 & 0.6109 & 0.5735 & 0.4909 \\
0.9330 & 0.9195 & 0.6109 & 4.0137 & 0.8947 & 0.8111 \\
0.8795 & 1.0276 & 0.5735 & 0.8947 & 3.8878 & 0.7300 \\
0.7543 & 0.6892 & 0.4909 & 0.8111 & 0.7300 & 4.6515
\end{pmatrix}
\]

Eigenvalues after one and two levels of HSS divide-and-conquer respectively:

\[
eig(\tilde{A}) = \begin{pmatrix} 2 & 3 - (3 \cdot 10^{-8}) & 3 - (10^{-8}) & 3 + (10^{-8}) & 3 + (3 \cdot 10^{-8}) & 4 \end{pmatrix}^T,
\]

\[
eig(A) = \begin{pmatrix} 2.1865 & 3.0000 & 3.0208 & 3.4912 & 4.0647 & 7.8974 \end{pmatrix}^T.
\]
Let $A$ have a known SVD and $wv^T$ be a rank-one update.

Key observation:

*This is equivalent to a rank-one update and downdate to a symmetric eigenproblem and hierarchical structure allows us to do this implicitly, and thus condition $\kappa(A)$ is unaffected.*

\[
\hat{A}^T \hat{A} = (V\Sigma U^T + wv^T)(U\Sigma V^T + vw^T),
\]

\[
= A^T A + V\Sigma U^T vw^T + vw^T U\Sigma V^T + (v^Tv)ww^T.
\]

Denote $\hat{v} = V\Sigma U^T v$ and $\alpha = v^Tv$.

So $\hat{A}^T \hat{A} = A^T A + \hat{v}w^T + w\hat{v}^T + \alpha ww^T$. 
Superfast SVD

Thus

\[ \hat{A}^T \hat{A} = A^T A + \begin{bmatrix} w & \hat{v} \end{bmatrix} \begin{bmatrix} \alpha & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} w^T \\ \hat{v}^T \end{bmatrix} \].

The Matrix \( \begin{bmatrix} \alpha & 1 \\ 1 & 0 \end{bmatrix} \) has a Schur decomposition \( Q_2 U_2 Q_2^T \).

Now let \( \sigma_2 = \frac{\alpha + \sqrt{\alpha^2 + 4}}{2} \), \( \sigma_2 = -\frac{1}{\sigma_2} \).

So we have that

\[ \hat{A}^T \hat{A} = A^T A + wQ_2 \sigma_1 Q_2^T w^T + \hat{v} Q_2 w \sigma_2 Q_2^T \hat{v}^T \]

\[ = AA^T + \rho_1 v_1^T + \rho_2 v_2^T, \]

where \( v_3 = \sqrt{-\sigma_1} wQ_2 \) and \( v_3 = \sqrt{\sigma_2} \hat{v} Q_2 \).
Many of the results from [Vogel, Xia, et al., 2015] can be proven in the case of the SVD in a straightforward way.

As a simple example, consider a block $2 \times 2$ HSS form:

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} \approx \tilde{A} = \begin{pmatrix} D_1 & U_1 B_1 U_2^T \\ U_2 B_2 V_1^T & D_2 \end{pmatrix}.$$  

Applying the Jordan-Wielandt theorem, it is easy to show that

$$\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \sim \begin{pmatrix} 0 & D_1 & 0 & \hat{U}_1 \hat{B}_1 \hat{V}_2^T \\ D_1^T & 0 & \hat{V}_1 \hat{B}_2^T \hat{U}_2^T & 0 \\ 0 & \hat{U}_2 \hat{B}_2 \hat{V}_1^T & 0 & D_2 \\ \hat{V}_2 \hat{B}_1^T \hat{U}_1^T & 0 & D_2^T & 0 \end{pmatrix}$$  

and thus (applying Lemma 4.1 from [Vogel, Xia, et al., 2015]) it follows that $|\sigma_i - \tilde{\sigma}_i| \leq \tau$, for all computed $\tilde{\sigma}_i$.  

Useful observation:

Suppose that we are computing an SVD for an HSS matrix $A$ but only want to
the top $k$ singular values/vectors. Then significant information can be discarded
throughout “Conquer” with only controlled (or often no) loss of accuracy

Example: suppose we have a block $2 \times 2$ matrix of only size $6 \times 6$, and that we only
wish to find the largest singular value. After the ‘Divide’ portion of the algorithm and
Conquer at the leaf level, we have that:

$$A = \begin{pmatrix} U_1 \Sigma_1 V_1^T \\ U_2 \Sigma_2 V_2^T \end{pmatrix} + wz^T = \begin{pmatrix} U_1 \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{pmatrix} V_1^T \\ \Sigma_2 V_2^T \end{pmatrix} + wz^T$$

where $\sigma_1 > \sigma_2 > \sigma_3$. We can set $\alpha_3 = 0$.

Principal components of proof:

// Weyl’s theorem (treat truncations as structured perturbations)
// Cauchy’s interlacing theorem of eigenvalues
// “shielding effect” (similar to [Paige])
// “rank-one updates are biased upwards”
In the Thin SVD case, the eigenvector storage $\sigma$ is reduced to:

$$\sigma = \sum_{i=1}^{\log_2(n/(2r))} (5m)(r)(2^{i-1}) \leq (5rm)\frac{n}{2r}(2) = O(mn)$$

// The scaling with $n$ is better than general case by factor of $\log n$
// Unlike the general case, there is no scaling with HSS rank $r$

The asymptotic complexity does not change in the Thin SVD case. But the bottleneck step (applications of eigenvector matrices) is reduced dramatically, and this can in practice reduce the total runtime by a factor $\geq 100$.

Instead of $\tau_1 = O(rn \log^2 n)$, we have that:

$$\tau_1 = \log nO(r \log n) \sum_{i=1}^{j} O(k) = O(r \log^2 n j k) = O(r k \log^3 n)$$

which is sublinear for many HSS matrices
Applications (PDEs)

// linear or near-linear complexity linear solvers for some discretized problems:

\[(I \otimes T + T \otimes I)x = b\]

\[TX + XT = B\]

\[T = USU^T\]

// Examples

- 2d Poisson equation
- 2d Helmholtz equation
- Legendre collocation

1) get \(\Lambda_i,\ z_i\)

2) \[g = (Z_1^T w \otimes Z_2^T w)f\]

3) solve \((\Lambda_1 \otimes I + I \otimes \Lambda_2)v = g\)

4) \[u = (Z_1 \otimes Z_2)v\]
"STEP 0" Compute a thin SVD of the constraint matrix $A$

\[
A \approx U\Sigma V^T
\]

$U \in \mathbb{R}^{m \times k}$, $\Sigma \in \mathbb{R}^{k \times k, \text{diag}}$, $V \in \mathbb{R}^{n \times k}$, $k$ small

for $i = 1, 2, 3, \ldots 10$ do

$\mu = 10^{2-i}$

Solve for minimum of unconstrained problem:

$\beta_\mu(x) = f(x) + \mu \varphi(x)$

while $\|\nabla(\beta_\mu)\| > \text{tol}$ do

STEP 1: Construct HSS approx. to Hessian $\tilde{H}$ (FMM, randomized sampling, use $U\Sigma V^T$)

STEP 2: Compute eigendecomposition $\tilde{H} = Q\Lambda Q^T$

STEP 3: Modify $\Lambda$ as necessary to preserve positive definiteness

STEP 4: $p = Q\Lambda^{-1}Q^T(\nabla(\beta_\mu))$

STEP 5: Compute step length $\alpha$ with Wolfe Line Search

STEP 6: $x \leftarrow x + \alpha p$

end while

end for
// Introduction

// Superfast divide-and-conquer (DC) [Vogel, Xia, et al., 2015]
  - Algorithm
  - Structured perturbation analysis

// Extensions of algorithm
  - Case of clustered eigenvalues and high multiplicities
  - SVD for non-symmetric and non-square matrices
  - Some preliminary applications: PDEs and Optimization

// Conclusion
  - Numerical tests
  - Future directions
Banded eigenvalue solution (full bandwidth = 10)

**Eigenvalue solution cost $\xi$**

$\xi$ (flops)

$n = 4000$, 70 times faster than XXC14, an $O(n^2)$ solver in [Xia, et al.]

<table>
<thead>
<tr>
<th>$n$</th>
<th>250</th>
<th>500</th>
<th>1000</th>
<th>2000</th>
<th>4000</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXC14</td>
<td>$e$</td>
<td>$2.28e - 10$</td>
<td>$1.43e - 10$</td>
<td>$5.59e - 11$</td>
<td>$3.31e - 11$</td>
</tr>
<tr>
<td></td>
<td>$\gamma$</td>
<td>$2.96e - 11$</td>
<td>$3.15e - 11$</td>
<td>$9.02e - 10$</td>
<td>$8.31e - 11$</td>
</tr>
<tr>
<td>NEW</td>
<td>$\gamma$</td>
<td>$4.39e - 11$</td>
<td>$3.23e - 11$</td>
<td>$1.42e - 10$</td>
<td>$8.47e - 11$</td>
</tr>
<tr>
<td></td>
<td>$\theta$</td>
<td>$4.29e - 16$</td>
<td>$7.39e - 16$</td>
<td>$1.53e - 15$</td>
<td>$3.99e - 15$</td>
</tr>
</tbody>
</table>

**Structured eigenmatrix storage $\sigma$**

$n = 4000$, 70 times faster than XXC14, an $O(n^2)$ solver in [Xia, et al.]

$\sigma$ (number of nonzeros)

Superfast Divide-and-Conquer

\[
\varepsilon = \frac{\|\Lambda - \hat{\Lambda}\|_2}{n\|\Lambda\|_2}, \quad \gamma = \frac{\max_i \|A\hat{q}_i - \hat{\lambda}_i \hat{q}_i\|_2}{n\|A\|_2}, \quad \theta = \frac{\max_i \|\hat{Q}^T \hat{q}_i - e_i\|_2}{n}
\]
Toeplitz eigenvalue solution

**Eigenvalue solution cost $\xi$**

<table>
<thead>
<tr>
<th>$n$</th>
<th>160</th>
<th>320</th>
<th>640</th>
<th>1280</th>
<th>2560</th>
</tr>
</thead>
<tbody>
<tr>
<td>XXC14</td>
<td>$e$ 2.40e−10</td>
<td>1.02e−10</td>
<td>5.80e−11</td>
<td>4.39e−11</td>
<td>3.84e−11</td>
</tr>
<tr>
<td>NEW</td>
<td>$e$ 1.00e−09</td>
<td>1.07e−10</td>
<td>1.47e−10</td>
<td>9.32e−11</td>
<td>8.45e−11</td>
</tr>
<tr>
<td></td>
<td>$\gamma$ 3.49e−09</td>
<td>1.49e−09</td>
<td>7.38e−10</td>
<td>2.53e−10</td>
<td>9.99e−11</td>
</tr>
<tr>
<td></td>
<td>$\theta$ 1.79e−16</td>
<td>3.69e−16</td>
<td>7.94e−16</td>
<td>6.56e−16</td>
<td>8.53e−16</td>
</tr>
<tr>
<td>NEW</td>
<td>$e$ 9.64e−16</td>
<td>1.01e−15</td>
<td>1.27e−15</td>
<td>1.07e−15</td>
<td>1.31e−15</td>
</tr>
<tr>
<td></td>
<td>$\gamma$ 4.14e−15</td>
<td>4.40e−15</td>
<td>6.69e−15</td>
<td>7.62e−15</td>
<td>6.26e−15</td>
</tr>
<tr>
<td></td>
<td>$\theta$ 4.25e−16</td>
<td>5.33e−16</td>
<td>7.24e−16</td>
<td>9.37e−16</td>
<td>7.18e−16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$n$</th>
<th>10^2</th>
<th>10^3</th>
<th>10^4</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\xi$ (flops)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$n$</td>
<td>10^2</td>
<td>10^3</td>
<td>10^4</td>
</tr>
<tr>
<td>$\sigma$ (number of nonzeros)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tau \approx 10^{-10}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\tau \approx 10^{-15}$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Structured eigenmatrix storage $\sigma$**

Superfast Divide-and-Conquer
Superfast divide-and-conquer for SVD

\[(A: \text{KMS Toeplitz})\]

\[(A_{i,j} = \sqrt{|x_i^{(n)} - x_j^{(n)}|})\]

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(n)</td>
<td>640</td>
<td>1280</td>
<td>2560</td>
</tr>
<tr>
<td>(\frac{|\Sigma - \Sigma^0|_2}{|\Sigma^0|_2})</td>
<td>1.55e – 07</td>
<td>2.56e – 07</td>
<td>3.62e – 07</td>
</tr>
<tr>
<td>(\frac{|A - U\Lambda V^T|_2}{|A|_2})</td>
<td>3.93e – 06</td>
<td>4.49e – 06</td>
<td>5.97e – 06</td>
</tr>
<tr>
<td>(|I - UU^T|_2)</td>
<td>2.35e – 15</td>
<td>4.25e – 15</td>
<td>5.62e – 15</td>
</tr>
</tbody>
</table>

\((A_{i,j} = \sqrt{|x_i^{(n)} - x_j^{(n)}|}, x_i^{(n)}: \text{Chebyshev roots})\)
Superfast divide-and-conquer for SVD

**Table:** Example (random banded matrix): Run time $t_{10}$ of NEW for finding the first 10 singular values of $A$, complexity $\tilde{\xi}_{10}$ of NEW for applying the singular vector matrices, and storage $\sigma_{10}$ of NEW for the eigenmatrix.

<table>
<thead>
<tr>
<th></th>
<th>$n$</th>
<th>250</th>
<th>500</th>
<th>1000</th>
<th>2000</th>
<th>4000</th>
<th>8000</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EIGS</strong></td>
<td>$t_{10}$</td>
<td>6.70e−02</td>
<td>1.12e−01</td>
<td>2.12e−01</td>
<td>4.52e−01</td>
<td>1.46e00</td>
<td>4.96e00</td>
</tr>
<tr>
<td><strong>NEW</strong></td>
<td>$t_{10}$</td>
<td>3.12e−02</td>
<td>7.05e−02</td>
<td>1.56e−01</td>
<td>3.48e−01</td>
<td>9.48e−01</td>
<td>2.54e00</td>
</tr>
<tr>
<td></td>
<td>$\tilde{\xi}$</td>
<td>4.50e06</td>
<td>7.86e06</td>
<td>2.80e07</td>
<td>5.64e07</td>
<td>2.01e08</td>
<td>3.99e09</td>
</tr>
<tr>
<td></td>
<td>$\sigma$</td>
<td>1.22e04</td>
<td>2.45e04</td>
<td>4.90e04</td>
<td>9.80e04</td>
<td>1.96e05</td>
<td>3.92e05</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$n$</th>
<th>250</th>
<th>500</th>
<th>1000</th>
<th>2000</th>
<th>4000</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NEW</strong></td>
<td>$e$</td>
<td>3.39e−7</td>
<td>4.38e−7</td>
<td>8.38e−7</td>
<td>9.39e−7</td>
<td>9.88e−7</td>
</tr>
<tr>
<td></td>
<td>$\gamma$</td>
<td>5.39e−7</td>
<td>6.39e−7</td>
<td>7.48e−7</td>
<td>8.39e−7</td>
<td>9.89e−7</td>
</tr>
<tr>
<td></td>
<td>$\theta$</td>
<td>2.53e−15</td>
<td>4.38e−15</td>
<td>1.93e−14</td>
<td>3.28e−14</td>
<td>3.59e−14</td>
</tr>
</tbody>
</table>
Future Directions

// Much more analysis!

// More general problems
  - generalized eigenvalue problems
  - MHS Matrices

// many more applications

// high performance codes
  - More adaptive
  - Parallel implementations
Thank you!!!

For more on structured eigenvalues:
math.purdue.edu/~xiaj
jimmyvogel.com

Structured Matrices and Applications
co-organized by Jimmy Vogel and Haizhao Yang (Duke)
speakers include Victor Pan (CUNY) and David Bindel (Cornell)
Friday October 30th, 2015
SIAM ALA, Atlanta, GA

Selected references:

// J. Vogel J. Xia, et al., Superfast divide-and-conquer method and
perturbation analysis for structured eigenvalue solutions. SIAM J. Sci.


// Y. Xi and J. Xia, On the stability of some hierarchical rank structured