SuperDC: A stable superfast divide-and-conquer eigensolver

Xiaofeng Ou
Joint work with Jianlin Xia

Department of Mathematics, Purdue University

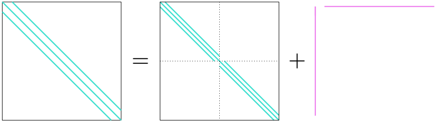
Fast Direct Solver 2021

October 26, 2021

Overview

- **Introduction and review**
- Stable and adaptive dividing strategy
- Stable and efficient conquering strategy
- Numerical experiments and performance

Divide and conquer eigensolutions for symmetric tridiagonal matrices



$= Q\Lambda Q^T$

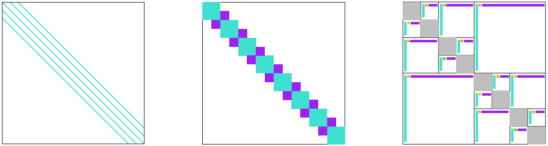
- Rank-one update to symmetric eigenvalue problems [Golub, 1973], [Bunch, Nielson, Sorenson, 1978], etc.
- Symmetric tridiagonal divide-and-conquer (DC) [since Cuppen, 1981]
- Arrow head eigenvalue problems [O'Leary, Stewart, 1990], [Arbenz, 1992], etc.
- Stability studies [Dongarra, Sorensen, 1987], [Sorensen, Tang, 1991], etc.
- Fully stable version [Gu, Eisenstat, 1995]

Costs

- $O(n^2)$ for eigenvalues, $O(n^3)$ for eigenvectors
- Possible acceleration to $O(n \log^p n)$ with FMM (not done) [Gu, Eisenstat, 1995], [Demmel, 1997], [Chandrasekaran, Gu, 2004]

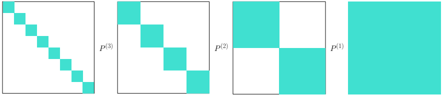
Superfast Divide and Conquer Eigensolver for rank-structured Hermitian matrix

$$A = Q\Lambda Q^*$$



Banded Block tridiagonal Dense HSS

- Stable, controllable accuracy, structured eigenmatrix Q



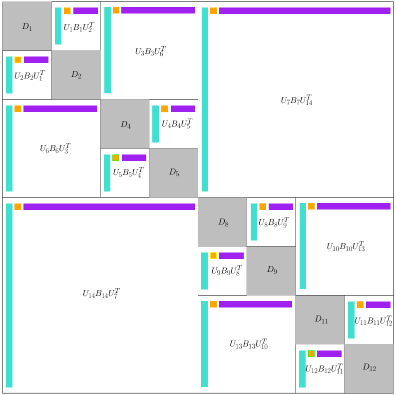
- No tridiagonal reduction needed.
- Almost linear complexity and storage (r = off-diagonal rank of A)

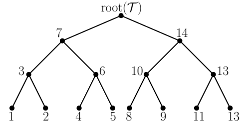
$$\xi_{\text{flops}} = O(r^2 n \log^2 n), \quad \xi_{\text{storage}} = O(rn \log n)$$

HSS Representation

Hierarchical semiseparable matrices (HSS)

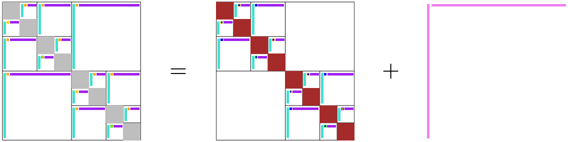
$$A = D_{\mathrm{root}(\mathcal{T})}, \quad D_i = \begin{pmatrix} D_{c_1} & U_{c_1} B_{c_1} U_{c_2}^T \\ U_{c_2} B_{c_2} U_{c_1}^T & D_{c_2} \end{pmatrix}$$





- Rank-structured
- Nested basis
- Examples
  - discretized kernel function
  - Toeplitz matrix in Fourier space
  - banded matrix
  - schur complement of discretized matrix
  - block tridiagonal matrix

HSS matrices can be divided [Vogel, Xia, et al, SISC 2016]



$$D_p = \begin{pmatrix} D_i - U_i B_i B_i^T U_i^T & \\ & D_j - U_j U_j^T \end{pmatrix} + \begin{pmatrix} U_i B_i \\ U_j \end{pmatrix} \begin{pmatrix} B_i^T U_i^T & U_j^T \end{pmatrix}$$

$$\equiv \begin{pmatrix} \tilde{D}_i & \\ & \tilde{D}_j \end{pmatrix} + Z_p Z_p^T$$

Theorem (HSS Structure preserving) [Vogel, Xia, et al, SISC 2016]

$\tilde{D}_i \equiv D_i - U_i H U_i^T$ is HSS with generators ($k \to k_l \to \cdots \to k_1 \to i$: a path in $\mathcal{T}$ from k to i)

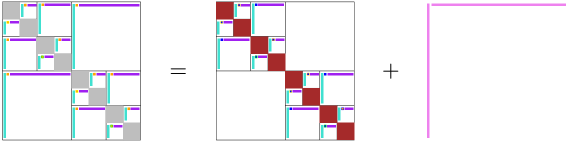
$$\tilde{U}_k = U_k, \quad \tilde{R}_k = R_k,$$

$$\tilde{B}_k = B_k - (R_k R_{k_l} \cdots R_{k_1}) H (R_{k_1}^T \cdots R_{k_l}^T R_{\mathrm{sib}(k)}^T),$$

$$\tilde{D}_k = D_k - U_k (R_k R_{k_l} \cdots R_{k_1}) H (R_{k_1}^T \cdots R_{k_l}^T R_k^T) U_k^T$$

Dividing and Conquer Eigensolver

$$D_p = \begin{pmatrix} \tilde{D}_i & \\ & \tilde{D}_j \end{pmatrix} + Z_p Z_p^T$$



- Solve recursively smaller problems $\tilde{D}_i = Q_i \Lambda_i Q_i^T$ and $\tilde{D}_j = Q_j \Lambda_j Q_j^T$

$$D_p = \begin{pmatrix} Q_i & \\ & Q_j \end{pmatrix} \left[\tilde{\Lambda}_p + \sum_{k=1}^{r} z_k z_k^T \right] \begin{pmatrix} Q_i^T & \\ & Q_j^T \end{pmatrix}$$

- Conquer by solving diagonal-plus-rank-r update problem

$$\tilde{\Lambda}_p + \sum_{k=1}^{r} z_k z_k^T = Q_p \Lambda_p Q_p^T$$

- Reduce to r rank-1 update problems.

Rank-1 update eigenproblem [Golub, 1973], [Bunch, Nielson, Sorenson, 1978]

$$\begin{pmatrix} \tilde{\lambda}_1 & & \\ & \ddots & \\ & & \tilde{\lambda}_n \end{pmatrix} + vv^T = Q \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} Q^T$$

λ_k's are roots of the secular function

$$f(\lambda) = 1 + \sum_{k=1}^{n} \frac{v_k^2}{\tilde{\lambda}_k - \lambda}.$$

and eigenmatrix

$$Q = \left(\frac{v_i s_j}{\tilde{\lambda}_i - \lambda_j} \right)_{1 \le i,j \le n}.$$

Rough idea:

f and Q can be accelerated by the **Fast Multipole method** [Greengard, Rokhlin, 1987].

Overview

- Introduction and review
- **Stable and adaptive dividing strategy**
- Stable and efficient conquering strategy
- Numerical experiments and performance

Dividing Strategy

Original dividing strategy [Vogel, Xia, et al, SISC 2016]

$$D_p = \begin{pmatrix} D_i - \underbrace{U_i B_i B_i^T U_i^T}_{Quadratic} & \\ & D_j - U_j U_j^T \end{pmatrix} + \begin{pmatrix} U_i B_i \\ U_j \end{pmatrix} \begin{pmatrix} B_i^T U_i^T & U_j^T \end{pmatrix}$$

$$\equiv \begin{pmatrix} \tilde{D}_i & \\ & \tilde{D}_j \end{pmatrix} + Z_p Z_p^T$$

- HSS generators of $\tilde{D}_i$ and $\tilde{D}_j$ are updated from D_i and D_j
- Straightforward but unstable
- Th quadratic terms $B_i B_i^T \Longrightarrow$ generators of $\tilde{D}_i$ grow enormously

Proposition (Unstable dividing) [Ou, Xia, 2020]

Suppose $\beta = \sup_{j \in \mathcal{T}} \|B_j\|_2$, then the final updated generators $\tilde{B}_i$ and leaf-level diagonal blocks $\tilde{D}_i$ satisfy

$$\|\tilde{D}_i\|_2 = O(\beta^{\frac{n}{2}}),$$
$$\|\tilde{B}_i\|_2 = O(\beta^{\frac{n}{4}}).$$

Specifically, if $\beta \gg 1$, $\tilde{D}_i$ and $\tilde{B}_i$ overflow.

Stable Dividing Strategy

Stable dividing strategy [Ou, Xia, 2020]

$$
\begin{aligned}
D_p &= \begin{pmatrix} D_i - \underbrace{\frac{1}{\|B_i\|_2} U_i B_i B_i^T U_i^T}_{\text{Scaled quadratic}} & \\ & D_j - \|B_i\|_2 U_j U_j^T \end{pmatrix} \\
&\qquad + \begin{pmatrix} \frac{1}{\sqrt{\|B_i\|_2}} U_i B_i \\ \sqrt{\|B_i\|_2} U_j \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{\|B_i\|_2}} B_i^T U_i^T & \sqrt{\|B_i\|_2} U_j^T \end{pmatrix} \\
&\equiv \begin{pmatrix} \hat{D}_i & \\ & \hat{D}_j \end{pmatrix} + \hat{Z}_p \hat{Z}_p^T
\end{aligned}
$$

- HSS generators of $\hat{D}_i$ and $\hat{D}_j$ are updated from D_i and D_j
- Scaled quadratic terms $\frac{B_i B_i^T}{\|B_i\|_2}$, stable

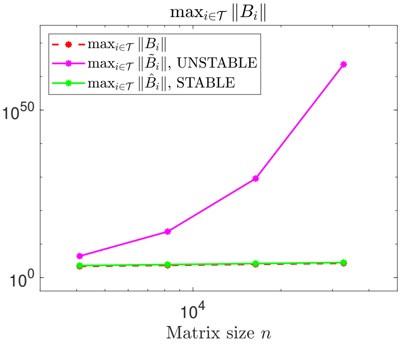
Proposition (Stable dividing) [Ou, Xia, 2020]

The updated generator $\hat{B}_i$ and leaf-level diagonal blocks $\hat{D}_i$ satisfy

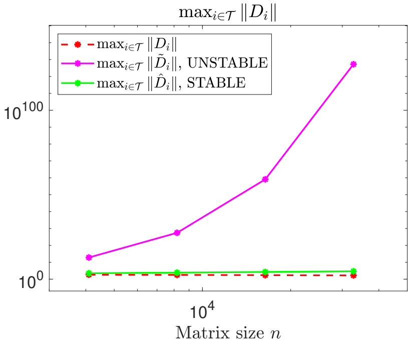
$$
\|\hat{D}_i\|_2 \leq \|D_i\|_2 + \frac{n}{2}\beta,
$$

$$
\|\hat{B}_i\|_2 \leq \frac{n}{4}\beta.
$$

Comparisons between two dividing strategy for $A = \left(\sqrt{|x_i - x_j|}\right)_{i,j}, x_i = \cos\frac{2\pi i+1}{2n}$.



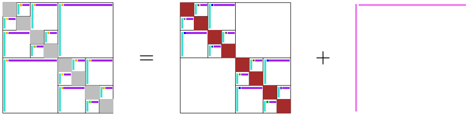
$\max_{i\in\mathcal{T}}\|B_i\|_2$



$\max_{i\in\mathcal{T}}\|D_i\|_2$

Adaptive Dividing Strategy

$$D_p = \begin{pmatrix} \hat{D}_i & \\ & \hat{D}_j \end{pmatrix} + \hat{Z}_p \hat{Z}_p^T$$



$$\# \text{ rank-1 update problems} = \text{colsize}(\hat{Z}_p)$$

Adaptivity to minimize colsize($\hat{Z}_p$)

$$\hat{Z}_p = \begin{cases} \begin{pmatrix} \frac{1}{\sqrt{\|B_i\|_2}} U_i B_i \\ \sqrt{\|B_i\|_2} U_j \end{pmatrix}, & \text{colsize}(B_i) \le \text{rowsize}(B_i) \\ \\ \begin{pmatrix} \sqrt{\|B_i\|_2} U_i \\ \frac{1}{\sqrt{\|B_i\|_2}} U_j B_i^T \end{pmatrix}, & \text{colsize}(B_i) > \text{rowsize}(B_i) \end{cases}$$

Overview

- Introduction and review
- Stable and adaptive dividing strategy
- **Stable and efficient conquering strategy**
- Numerical experiments and performance

Rank-1 update eigenproblem [Golub, 1973], [Bunch, Nielson, Sorenson, 1978]

$$\begin{pmatrix} \tilde{\lambda}_1 & & \\ & \ddots & \\ & & \tilde{\lambda}_n \end{pmatrix} + vv^T = Q \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix} Q^T$$

where $v = \begin{pmatrix} v_1 & \cdots & v_n \end{pmatrix}^T$.

λ_k's are roots of the secular function

$$f(\lambda) = 1 + \sum_{k=1}^{n} \frac{v_k^2}{\tilde{\lambda}_k - \lambda},$$

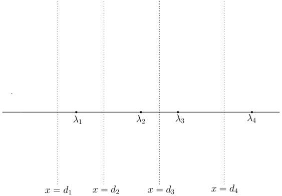
and eigenvector

$$q_k = \begin{pmatrix} \frac{v_1}{\tilde{\lambda}_1 - \lambda_k} & \cdots & \frac{v_n}{\tilde{\lambda}_n - \lambda_k} \end{pmatrix}^T.$$

Stable and Efficient Conquering Strategy

Classical conquering strategy [LAPACK, 1999], [Gu, Eisenstat, 1995], [Li, 1993]

$$f(x) \equiv 1 + \sum_{i=1}^{n} \frac{v_i^2}{d_i - x} = 0$$



- Solve each root $\lambda_k \in (d_k, d_{k+1})$ by modified Newton's method.
- To avoid cancellation,
  1. Separate positive and negative terms

$$f(x) = 1 + \underbrace{\sum_{i=1}^{k} \frac{v_i^2}{d_i - x}}_{<0} + \underbrace{\sum_{i=k+1}^{n} \frac{v_i^2}{d_i - x}}_{>0} \equiv 1 + \psi_k(x) + \phi_k(x),$$

  2. Shift the origin to d_k by letting $\delta_{ik} = d_i - d_k$, and solve the shifted equation

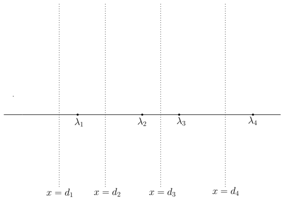
$$0 = g_k(\eta) = f(\eta + d_k) = 1 + \sum_{i=1}^{k} \frac{v_i^2}{\delta_{ik} - \eta} + \sum_{i=k+1}^{n} \frac{v_i^2}{\delta_{ik} - \eta}.$$

  Shifting is crucial for stability and accuracy (e.g., clustered eigenvalues).

Stable and Efficient Conquering Strategy

Classical conquering strategy [LAPACK, 1999], [Gu, Eisenstat, 1995], [Li, 1993]

$$f(x) \equiv 1 + \sum_{i=1}^{n} \frac{v_i^2}{d_i - x} = 0$$



- For stability
  3. Compute Lowner's formula

$$\hat{v}_i = \sqrt{\frac{\prod_j (\lambda_j - d_i)}{\prod_{j \neq i} (d_j - d_i)}},$$

  and (stable) dense eigenvector

$$\hat{q}_k = \begin{pmatrix} \frac{\hat{v}_1}{\tilde{\lambda}_1 - \lambda_k} & \cdots & \frac{\hat{v}_n}{\tilde{\lambda}_n - \lambda_k} \end{pmatrix}^T.$$

- About $O(n)$ to find one root λ_k, hence roughly $O(n^2)$ to find all pairs $\{(\lambda_k, \hat{q}_k)\}_{k=1}^n$

Stable and Efficient Conquering Strategy

New stable and efficient conquering strategy

$$f(x) \equiv 1 + \sum_{i=1}^{n} \frac{v_i^2}{d_i - x} = 0$$

- Deflation with user-supplied accuracy τ
- Assemble f as mat-vec with $C = \left(\frac{1}{d_i - x_k}\right)$

$$f(\vec{x}) = \vec{1} + C\vec{v}$$

- Solve all roots $\{\lambda_k\}_{k=1}^{n}$ simultaneously with FMM acceleration
- Our previous work [Vogel, Xia, SISC 2016] uses direct FMM acceleration but lack of stability.

$$f(\vec{x}) = \vec{1} + C\vec{v} = 0$$

- For stability
  1. lower and upper triangular split (corresponding to positive-negative split)

$$f(\vec{x}) = \vec{1} + C\vec{v} = \vec{1} + C_{\text{lower}}\vec{v} + C_{\text{upper}}\vec{v},$$

Triangular FMM $\xrightarrow{\text{accelerate}}$ triangular matrix-vector products $C_{\text{lower}}\vec{v}$, $C_{\text{upper}}\vec{v}$

  2. Shifting $\delta_{ik} = d_i - d_k$ is NOT directly applicable in triangular FMM.

e.g. consider the shifted matrix $C_{\text{shift}} = \left(\frac{1}{\delta_{ik} - (\lambda_k - d_k)}\right)$

possible remedy:

no shift in FMM, but store extra dense eigenvectors (not efficient).

- For stability
  3. To incorporate shifting into triangular FMM:
     - *near-field*: shift locally $\delta_{ik} = d_i - d_k$
     - *far-field*: compute as usual

Importance of near-field local shifting

- Stability: avoid division by zero, ensure eigenmatrix orthogonality
- Efficiency: speed up convergence of root-finding

n	4,096	8,192	16,384	32,768	65,536
With local shifting	98.97%	99.25%	99.09%	99.11%	99.16%
Without local shifting	30.9%	34.8%	35.0%	39.9%	53.7%

Eigenvalues converged within 5 iterations.

Stable and Efficient Conquering Strategy

- For stability

4. Lowner's formula with FMM

$$\log \hat{v}_i = \frac{1}{2}\left(\sum_k \log|\lambda_k - d_i| - \sum_{k\neq i}|d_k - d_i|\right)$$

5. Normalized eigenvector with FMM

$$s_k^{-1} = \sum_i \frac{\hat{v}_i^2}{(d_i - \lambda_k)^2}$$

6. Cauchy-like structured eigenmatrix

$$Q = \left(\frac{\hat{v}_i s_k}{d_i - \lambda_k}\right).$$

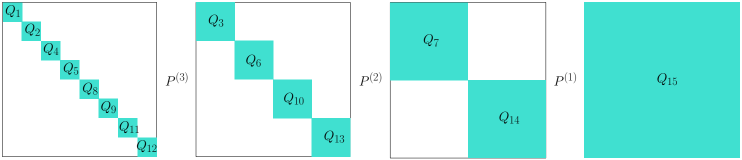
- Roughly $O(n)$ to find all roots $\{\lambda_k\}_{k=1}^n$.

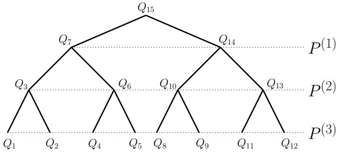
Stable and Efficient Conquering Strategy

Conquer stage

Classical	New
Solve each root λ_k separately	Solve all roots $\{\lambda_k\}$ simultaneously
Separate positive and negative terms $$f(x) = 1 + \psi_k(x) + \phi_k(x)$$	Split lower and upper triangular parts $$f(\vec{x}) = \vec{1} + C_{\text{lower}}\vec{v} + C_{\text{upper}}\vec{v}$$
Evaluate $f(x)$ directly	Evaluate $f(\vec{x})$ with triangular FMM
Global shifting $\delta_{ik} = d_i - d_k$ for $1 \le i \le n$	Local shifting $\delta_{ik} = d_i - d_k$ in *near field*
Lowner's formula	Lowner's formula with FMM
Dense eigenmatrix $$Q = \left(\frac{\hat{v}_i s_k}{d_i - \lambda_k}\right) = \left(\frac{\hat{v}_i s_k}{\delta_{ik} - \eta_k}\right)$$	Cauchy-like structured eigenmatrix $$Q = \left(\frac{\hat{v}_i s_k}{d_i - \lambda_k}\right)$$
$O(n^2)$ operations + $O(n^2)$ storage	About $O(n)$ operations + $O(n)$ storage

Structured Eigenmatrix





- $Q_i = \begin{cases} \text{dense}, & \text{leaf node}, \\ \prod_{l=1}^{r} \text{Cauchy-like}, & \text{non-leaf node}, \end{cases}$
- $P^{(l)}$ = permutation matrices.
- Interlacing product of HSS and permutaion matrices

$$Q = Q^{(L)} \prod_{l=L-1}^{0} \left(P^{(l)} Q^{(l)} \right),$$

where $Q^{(l)} = \mathrm{diag}(Q_i)_{\mathrm{lvl}(i)=l}$ is HSS.

Performance

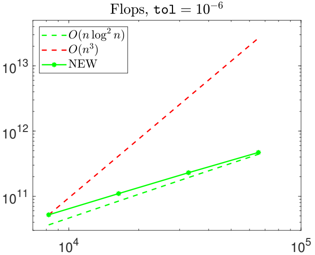
Performance compared to MATLAB's **eig**

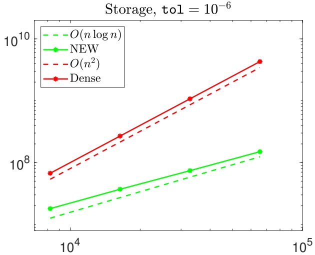
1. Tridiagonal matrix
   - Breakeven point ≈ 4096
   - At $n = 32764$, 6**x** faster and 6% memory
2. Banded matrix with corner points
   - At $n = 32764$, 11**x** faster and 7% memory
3. Prolate Toeplitz matrix
   - Off-diagonal rank $r = O(\log n)$
   - Intensive deflations happen
   - At $n = 32764$, 136**x** faster and 6% memory

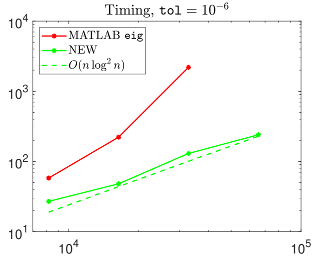
Memory efficient, applicable to much larger n.

Numerical Experiments

Dense kernel matrix $A = \left(\sqrt{|x_i - x_j|}\right)_{i,j}$, $x_i = \cos\frac{2\pi i+1}{2n}$.





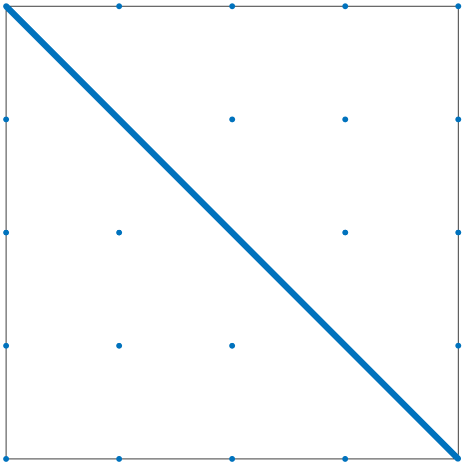


n	8192	16384	32768	65536
γ	8.0e-15	4.7e-15	3.3e-15	1.9e-15
e	2.4e-11	2.0e-11	1.1e-11	N/A
θ	3.2e-11	1.5e-11	7.9e-12	4.4e-12

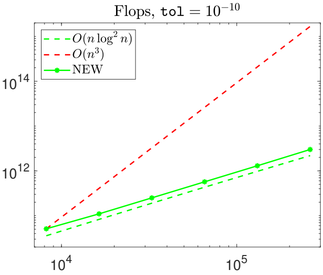
- $\gamma = \max \frac{\|Q^T q_i - e_i\|}{n}$, $e = \frac{\|\Lambda - \Lambda^{true}\|}{n\|\Lambda^{true}\|}$
- $\theta = \frac{\|AQ - \Lambda Q\|}{n\|A\|}$

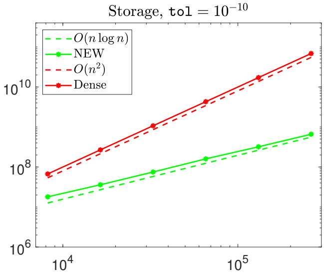
Numerical Experiments

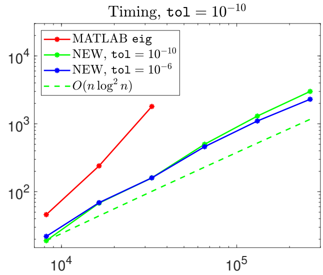
5-banded matrix with corner points, tolerance $\tau = 10^{-10}$



Numerical Experiments







n	8192	16384	32768	65536	131072
γ	1.8e-15	2.7e-15	2.3e-15	4.8e-15	2.3e-15
e	1.4e-17	6.4e-18	8.1e-17	N/A	N/A
θ	6.4e-15	6.8e-15	1.1e-14	1.1e-14	1.8e-14

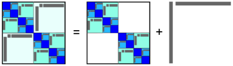
- $\gamma = \max \frac{\|Q^T q_i - e_i\|}{n}$, $e = \frac{\|\Lambda - \Lambda^{true}\|}{n\|\Lambda^{true}\|}$
- $\theta = \frac{\|AQ - \Lambda Q\|}{n\|A\|}$

SuperDC Package

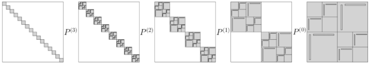
- https://www.math.purdue.edu/~xiaj/

SuperDC

Stable Superfast Divide-and-Conquer Eigenvalue Decomposition (in Matlab)



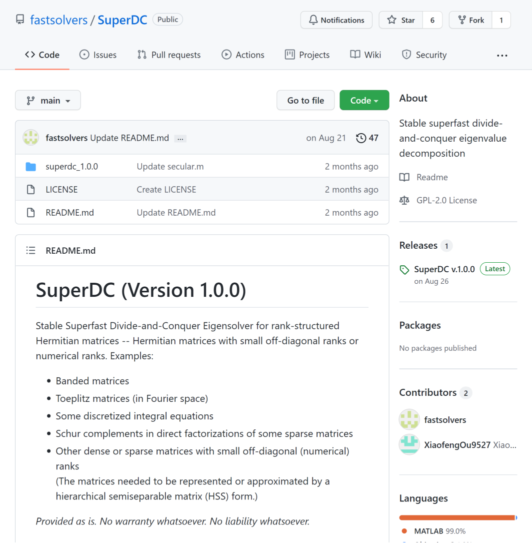
$A = Q\Lambda Q^*, \quad Q =$



- **Developers**
  Xiaofeng Ou (ou17 -at- purdue.edu), Jianlin Xia, Purdue University. Feedback and suggestions are welcome.
  Provided as is. No warranty whatsoever. No liability whatsoever. (GNU General Public License v2.0)
- **Applicability**
  Rank-structured Hermitian matrices (represented/approximated by HSS forms). Examples:
  - Banded matrices
  - Toeplitz matrices (in Fourier space)
  - Some discretized integral equations
  - Schur complements in direct factorizations of some sparse matrices
  - Other dense or sparse matrices with small off-diagonal (numerical) ranks
- **Features**
  - *No tridiagonal reduction needed*
  - *Nearly O(n) complexity for full eigenvalue decomposition*
  - *Nearly O(n) storage for all eigenvectors*
  - *Many stability features (stabilization like in tridiagonal divide-and-conquer, safeguards for clustered eigenvalues, stable FMM, etc.)*
  - *User controllable accuracy*

SuperDC Package

- https://github.com/fastsolvers/SuperDC



SuperDC Package

Currently support:

- HSS matrix
- Banded matrix
- Block tridiagonal matrix

Ongoing future work:

- SVD decomposition
- Generalized eigenvalue solution
- General graph sparse eigenvalue solution

Applications:

- Matrix functions computations
- Separable PDEs solution

E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, SIAM, Philadelphia, PA, third ed., 1999.

J. R. Bunch, C. P. Nielsen, and D. C. Sorensen, *Rank-one modification of the symmetric eigenproblem*, Numer. Math., 31 (1978), pp. 31–48.

D. Cai and J. Xia, *A stable and efficient matrix version of the fast multipole method*, preprint, (2016).

S. Chandrasekaran and M. Gu, *A divide-and-conquer algorithm for the eigendecomposition of symmetric block diagonal plus semiseparable matrices*, Numer. Math., 96 (2004), pp. 723–731.

J. W. Demmel, *Applied numerical linear algebra*, SIAM, 1997.

M. Gu and S. C. Eisenstat, *A divide-and-conquer algorithm for the symmetric tridiagonal eigenproblem*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 79–92.

R. C. Li, *Solving secular equations stably and efficiently*, tech. report, CALIFORNIA UNIV BERKELEY DEPT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCES, 1993.

X. OU AND J. XIA, *SuperDC: A stable superfast divide-and-conquer eigensolver*, preprint (2020)

J. Vogel, J. Xia, S. Cauley, and V. Balakrishnan, *Superfast divide-and-conquer method and perturbation analysis for structured eigenvalue solutions*, SIAM J. Sci. Comput., 38 (2016), pp. A1358-A1382.

J. Xia, *On the complexity of some hierarchical structured matrix algorithms*, SIAM Journal on Matrix Analysis and Applications, 33 (2012), pp. 388–410.

J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li, *Fast algorithms for hierarchically semiseparable matrices*, Numer. Linear Algebra Appl., 17 (2010), pp. 953–976.

J. J. M.Cuppen, *A divide and conquer method for the symmetric tridiagonal eigenproblem*, Numer. Math., 36 (1981), pp. 177–195.

L. Greengard and V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.