Spectral Transformation Lanczos Algorithm for the Symmetric Definite Generalized Eigenvalue Problem[1]

Ayobami Adebesin

April 8th, 2025

[1] **[1]:** I think it is appropriate to use your thesis title.

Problem Statement

The Symmetric Definite Generalized Eigenvalue Problem

For $n \times n$ real matrices $A = A^{\mathrm{T}}$ and positive definite (or semidefinite) $B = B^{\mathrm{T}}$, find $\boldsymbol{v} \neq \mathbf{0}$ and λ such that

$$A\boldsymbol{v} = \lambda B\boldsymbol{v}$$

where $\mathcal{N}(A) \cap \mathcal{N}(B) = \{\mathbf{0}\}$. The value λ is a generalized eigenvalue and $\boldsymbol{v}$ is the corresponding generalized eigenvector. If B is invertible, the generalized eigenvalues are eigenvalues for $B^{-1}A\boldsymbol{v} = \lambda\boldsymbol{v}$. Otherwise, for $B\boldsymbol{v} = \mathbf{0}$, we say that $\lambda = \infty$ is an eigenvalue. If B is positive definite, there are n linearly independent eigenvectors.

We assume throughout that $A \neq 0$ and $B \neq 0$.

Applications and Algorithms

- Generalized eigenvalues are real for this problem.
- Vibration Analysis in Structural engineering e.g (Boeing). For example, equation of a vibrating system:

$$K\mathbf{x} = \lambda M\mathbf{x}$$

  where M is the mass matrix, K is the stiffness matrix, $\mathbf{x}$ is the displacement vector, and λ is the eigenfrequencies of the system.
- Existing algorithms for both dense and sparse problems typically reduce to a symmetric standard eigenvalue problem. ~~Existing factorization algorithms for the dense problem all have performance and/or stability issues.~~
- Reducing to a symmetric eigenvalue problem preserves symmetry and we get real eigenvalues.
- Stability has been a long-standing concern.
- Recent work on direct methods have proven residual bounds for dense problems. [Michael Stewart, 2024].

The Standard ~~ReductionAlgorithm~~ (J.H. Wilkinson, 1965)

If B is positive definite then it has a Cholesky factor

$$B = C_b C_b^{\mathrm{T}}.$$

Thus

$$A\boldsymbol{v} = \lambda B\boldsymbol{v} \Leftrightarrow A\boldsymbol{v} = \lambda C_b C_b^{\mathrm{T}}\boldsymbol{v} \Leftrightarrow C_b^{-1} A C_b^{-\mathrm{T}} C_b^{\mathrm{T}}\boldsymbol{v} = \lambda C_b^{\mathrm{T}}\boldsymbol{v}.$$

Standard Reduction to an Ordinary Eigenvalue Problem

Solve the symmetric eigenvalue problem

$$C_b^{-1} A C_b^{-\mathrm{T}}\boldsymbol{u} = \lambda\boldsymbol{u}, \qquad \boldsymbol{u} \neq \mathbf{0}$$

Then solve $C_b^{\mathrm{T}}\boldsymbol{v} = \boldsymbol{u}$.

This is a symmetric eigenvalue problem that can be solved using either a direct method or the Lanczos algorithm

The Standard Algorithm (cont'd)

- This shows that the symmetric definite generalized eigenvalue problem has real eigenvalues. Eigenvectors are orthogonal with respect to the inner product $(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{y}^{\mathrm{T}} B \boldsymbol{x}$.
- For dense problems~~It is fast and~~ it is the approach used by LAPACK. It is also used frequently, but not universally, with the Lanczos algorithm for sparse problems.
- For efficiency on sparse problems, a sparse Cholesky factorization is required to use it with Lanczos.
- It fails if B is semidefinite and is unstable when B is ill-conditioned (i.e. $\kappa_2(B) = \|B\|_2 \|B^{-1}\|_2$ is large.)
- If B is ill-conditioned, it usually delivers small residuals for large eigenvalues and large residuals for small eigenvalues.
- ~~There are alternatives, each with its own set of problems...~~[2]

[2] **[2]:** remove

An Alternate Formulation

The generalized eigenvalue problem can be formulated in another way as follows:

The Generalized Eigenvalue Problem Version II

The eigenvalue problem can be written

$$\beta A\boldsymbol{v} = \alpha B\boldsymbol{v}$$

where $\boldsymbol{v} \neq \mathbf{0}$ and β and α are not both zero. The original formulation eigenvalues are given by $\lambda = \alpha/\beta$. Each eigenvalue is a nonunique pair (α, β) that can be scaled by $c \neq 0$. It can be identified with a subspace of $\mathbb{R}^2$ (or $\mathbb{C}^2$):

$$\mathcal{E} = \{c \cdot (\alpha, \beta) : c \in \mathbb{R}\}$$

The QZ Algorithm I (C. B. Moler and G. W. Stewart, 1972)[3]

Generalized Schur Form

For A and B not necessarily symmetric, there exist unitary Q and Z such that

$$Q^H AZ = T_a, \qquad Q^H BZ = T_b.$$

where T_a and T_b are upper triangular with diagonal elements α_i and β_i. The eigenvalues for $A\boldsymbol{v} = \lambda B\boldsymbol{v}$ are given by $\lambda_i = \alpha_i/\beta_i$. Eigenvectors can be obtained from Z with additional computation.

- With rounding, the QZ algorithm for computing this is backward stable: There exist exactly unitary $\tilde{Q}$ and $\tilde{Z}$ close to the computed Q and Z for which the computed T_a and T_b satisfy

$$\tilde{Q}^H (A+E)\tilde{Z} = T_a, \qquad \tilde{Q}^H (B+F)\tilde{Z} = T_b.$$

[3] **[3]:** I would remove this slide, but maybe mention elsewhere that the QZ algorithm is standard for the dense non-symmetric problem, but is not suitable

More on the QZ Algorithm[4]

- The errors satisfy $\|E\| = O(u)\|A\|$ and $\|F\| = O(u)\|B\|$, where u is the unit roundoff. ($u \approx 10^{-16}$ for double precision.)
- The pairs (α_i, β_i) are exact generalized eigenvalues of matrices close to A and B.
- The algorithm is much slower than the standard algorithm.
- Unfortunately E and F are not guaranteed to be symmetric even when A and B are. The computed eigenvalues can even have imaginary parts that are not small. Simply truncating the imaginary part does not give satisfactory results.

[4] **[4]:** also remove

Diagonalization Using Congruences
S. Chandrasekaran 2000[5]

Diagonalization

For the symmetric definite problem there exists nonsingular Z such that

$$A = ZD_aZ^{\mathrm{T}}, \qquad B = ZD_bZ^{\mathrm{T}}.$$

If α_i and β_i are the diagonal elements of D_a and D_b, then the generalized eigenvalues are (α_i, β_i) or $\lambda_i = \alpha_i/\beta_i$. The eigenvectors are the columns of $V = Z^{-\mathrm{T}}$.

- It can be shown that $V = Z^{-\mathrm{T}}$ is a good eigenvector matrix.
- It is as close to ideal numerically as any current algorithm.
- It involves solving multiple ordinary eigenvalue problems and its complexity is not proven to be $O(n^3)$.

[5] **[5]:** remove

Spectral Transformation Lanczos [T. Ericsson and A. Ruhe, 1980][6]

Lemma

Let $\lambda = \alpha/\beta \neq \infty$ *and* $\boldsymbol{v} \neq 0$ *satisfy* $A\boldsymbol{v} = \lambda B\boldsymbol{v}$. *Assume that* $A - \sigma B$ *is nonsingular and* $B = C_b C_b^{\mathrm{T}}$, $C_b \in \mathbb{R}^{n \times r}$ *with linearly independent columns. Then* $\theta = 1/(\lambda - \sigma)$ *is an eigenvalue of the shifted and inverted problem*

$$C_b^{\mathrm{T}}(A - \sigma B)^{-1} C_b \boldsymbol{u} = \theta \boldsymbol{u}, \qquad \boldsymbol{u} \neq \mathbf{0}.$$

with eigenvector $\boldsymbol{u} = C_b^{\mathrm{T}} \boldsymbol{v} \neq \mathbf{0}$.
Conversely, assume that $\boldsymbol{u} \neq \mathbf{0}$ *is an eigenvector for the shifted and inverted problem with eigenvalue* θ. *Then the vector* $\boldsymbol{v} = (A - \sigma B)^{-1} C_b \boldsymbol{u} \neq 0$ *is an eigenvector for the eigenvalue* $(1 + \sigma\theta, \theta)$ *of the original problem.*

[6] **[6]:** add a slide before this one describing the spectral transformation more generally for the standard eigenvalue problem and the nonsymmetric problem;

Spectral Transformation for Dense Problems [Michael Stewart, 2024]

This direct method employs the spectral transformation described by [T. Ericsson and A. Ruhe, 1980], and symmetric decompositions of $A - \sigma B$ and B such that

$$A - \sigma B = C_a D C_a^{\mathrm{T}}, \qquad \text{and} \qquad B = C_b C_b^{\mathrm{T}},$$

to transform the problem into a symmetric standard eigenvalue problem given by

$$C_b^{\mathrm{T}} C_a^{-T} D_a C_a^{-1} C_b \boldsymbol{u} = \theta \boldsymbol{u}, \qquad \boldsymbol{v} = C_a^{-\mathrm{T}} D_a C_a^{-1} C_b \boldsymbol{u}$$

with $(\alpha, \beta) = (1 + \sigma\theta, \theta)$ or $\lambda = (1 + \sigma\theta)/\theta$.

- B can factored using pivoted Cholesky and A using LDL^{T} factorization with rook pivoting, both available in LAPACK.
- We cannot expect a shift to result in well conditioned $A - \sigma B$ or C_a, **but ill conditioning is not what matters!**

Interesting Questions[7]

- Do the residual bounds proven for a direct method apply ~~applies~~ when an iterative method is used for the spectral transformed problem?
- Does the spectral transformed problem respects symmetry in the decomposition of $A - \sigma B$?

[7] **[7]:** you refer to residual bounds for direct methods. I would also point out here, before mentioning residual bounds, that such bounds were proven in my paper for the direct method.

What is our approach?

- Apply the Lanczos algorithm to the spectral problem
- Investigate if the residuals for the computed eigenvalues follows the bounds for the direct methods in terms of the choice of shift.~~?~~
- Explore the effect of preserving symmetry in the~~symmetric~~ decomposition of $A - \sigma B$ on residuals

Spectral Transformation Lanczos Algorithm[8] I

```
function SPECTRAL_LANCZOS(A, B, m, n, σ, tol)
    Choose an arbitrary vector b and set an initial vector q_1 = b/||b||_2
    Set β_0 = 0 and q_0 = 0
    Set Q = zeros(m, n + 1)
    Precompute the LU factorization of A − σB: LU = (A − σB)
    Factor: B = CC^T
    for j = 1, 2, ..., n do
        Q[:, j] = q_j
        u = Cq_j
        Solve: (LU)v = u for v
```

Spectral Transformation Lanczos Algorithm[9] II

$\mathbf{v} = \mathbf{v} - \beta_{j-1}\mathbf{q}_{j-1} - \alpha_j \mathbf{q}_j$

Full reorthogonalization: $\mathbf{v} = \mathbf{v} - \sum_{i \le j} (\mathbf{q}_i^T \mathbf{v})\mathbf{q}_i$

$\beta_j = \|\mathbf{v}\|_2$

if $\beta_j < tol$ **then**

  restart or **exit**

end if

$\mathbf{q}_{j+1} := \mathbf{v}/\beta_j$

end if

end for

$Q = Q[:, : n]$[a]

[a] **[8]:** I would use $Q[:, 1 : n]$. The notation $: n$ is very specific to Python and not widely used in pseudocode.

$\mathbf{q} = Q[:, n]$

return $(Q, T, \mathbf{q})$

Spectral Transformation Lanczos Algorithm[10] III

end function

Some Definitions from [Michael Stewart, 2024]

$$X = C_a^{-1} C_b, \qquad W = X^{\mathrm{T}} D_a X, \qquad \mu = \frac{\|X\|_2^2}{\|W\|_2} \geq 1.$$

$$\eta = \frac{\|A - \sigma B\|_2^{1/2}}{\|B\|_2^{1/2}}, \qquad \sigma_0 = \sigma \frac{\|B\|_2}{\|A\|_2}, \qquad \text{and} \qquad \gamma = \frac{\|A\|_2}{\|A - \sigma B\|_2}.$$

- The shifted and inverted problem is $W\boldsymbol{u} = \theta \boldsymbol{u}$.
- The only "inversion" is in solving $C_a X = C_b$.
- The values of μ, $\eta\|X\|_2$, σ_0, and γ can potentially impact stability.
- $\eta\|X\|_2$ is the most interesting and important of these.

Bounds for $\eta\|X\|_2$

We have

$$\eta^2\|X\|_2^2 \leq \mu\kappa_2(A - \sigma B)$$

and even better

Lemma

Assume that $\sigma \neq 0$ and $A - \sigma B$ is invertible. Then

$$\begin{aligned}\eta^2\|X\|_2^2 &\leq \left(1 + \frac{1}{|\sigma_0|}\right)\frac{\mu}{\min_i\left|1 - \frac{\lambda_i}{\sigma}\right|}\\ &= (1 + |\sigma_0|)\frac{\mu}{\min_i\left|\frac{\|B\|_2}{\|A\|_2}\lambda_i - \sigma_0\right|}.\end{aligned}$$

Forward Errors

- **The size of $\eta^2\|X\|_2^2$ determines stability and it is usually much smaller than $\kappa_2(A - \sigma B)$.**
- It is surprisingly easy to avoid large $\eta\|X\|_2$. In practice, if $|\sigma_0|$ is not small, $\eta\|X\|_2$ is large only if σ is chosen to match an eigenvalue λ to several digits. A random shift in a reasonable interval almost always works.
- If A and B are both positive definite, all generalized eigenvalues are positive, $\mu = 1$, and simply choosing $\sigma_0 = -1$ gives

$$\eta^2\|X\|_2^2 \leq \left(1 + \frac{1}{1}\right) \frac{1}{\min_i \left|1 + \frac{|\lambda_i|}{|\sigma|}\right|} \leq 2$$

This is a very common special case in structural engineering.

Error Bounds: Moderate Shifts and Eigenvalue Stability

Eigenvalue Backward Errors

For the computed θ_i, there exist symmetric E and F and a vector $\tilde{\boldsymbol{v}}_i \neq \mathbf{0}$ such that

$$\theta_i(A+E)\tilde{\boldsymbol{v}}_i = (1+\sigma\theta_i)(B+F)\tilde{\boldsymbol{v}}_i$$

and

$$\max\left(\frac{\|E\|_2}{\|A\|_2}, \frac{\|F\|_2}{\|B\|_2}\right) \leq O(u)(1+|\sigma_0|)\eta^2\|X\|_2^2 + O(u^2)$$

If $|\sigma_0|$ is not large and $\eta^2\|X\|_2^2$ is not large, each $(1+\sigma\theta_i, \theta_i)$ is an eigenvalue of a pair close to (A, B).

Error Bounds: Moderate Shifts with Computed Eigenvectors

Computed Eigenvector Bounds

There exist symmetric E and F such that the computed θ_i and the computed eigenvector $\boldsymbol{v}_i$ satisfy

$$\theta_i(A+E)\boldsymbol{v}_i = (1+\sigma\theta_i)(B+F)\boldsymbol{v}_i$$

with

$$\max\left(\frac{\|E\|_2}{\|A\|_2}, \frac{\|F\|_2}{\|B\|_2}\right) \leq$$
$$O(u)|1-\lambda_i/\sigma||\sigma_0|\left(1+\max(\gamma,1)\left(1+|1-\lambda_i/\sigma|\right)\eta^2\|X\|_2^2\right)+O(u^2)$$

If $|\sigma_0|$ and $\eta^2\|X\|_2^2$ are not large, $A-\sigma B$ does not cancel, and $\lambda_i = (1+\sigma\theta_i)/\theta_i$ is not much larger than σ, then each $(1+\sigma\theta_i, \theta_i)$ and $\boldsymbol{v}_i$ is an eigenvalue/eigenvector of a pair close to (A, B).

Error Bounds: Large Shifts with Computed Eigenvectors

Computed Eigenvector Bounds

There exist symmetric E and F such that the computed θ_i and the computed eigenvector $\boldsymbol{v}_i$ satisfy

$$\theta_i(A+E)\boldsymbol{v}_i = (1+\sigma\theta_i)(B+F)\boldsymbol{v}_i$$

with

$$\max\left(\frac{\|E\|_2}{\|A\|_2}, \frac{\|F\|_2}{\|B\|_2}\right) \leq$$
$$O(u)|1-\sigma/\lambda_i|\left(1+\max(\gamma,1)\left(1+|1-\lambda_i/\sigma|\right)\eta^2\|X\|_2^2\right)+O(u^2)$$

If $\eta^2\|X\|_2^2$ is not large, $A-\sigma B$ does not cancel, and $\lambda_i = (1+\sigma\theta_i)/\theta_i$ is not much larger or smaller than σ, then each $(1+\sigma\theta_i, \theta_i)$ and $\boldsymbol{v}_i$ is an eigenvalue/eigenvector of a pair close to (A, B).

Setting up a Generalized Eigenvalue Problem I[11]

Given a diagonal matrix $D \in \mathbb{R}^{m \times m}$ with predefined eigenvalues, and regularization hyperparameter δ, the following algorithm sets up a generalized eigenvalue problem

function $\textsc{Generate\_Matrix}(D, \delta)$
    Set $m = \texttt{size}(D)$
    Q, $\_\_ = \texttt{qr}(\text{random.randn}(m, m))$
    $C = QDQ^T$
    $L_0 = \texttt{tril}(\texttt{random.randn}(m, m))$
    $B = (L_0 L_0^T) + \delta I$
    $L = \texttt{cholesky}(B)$
    $A = LCL^T$
    return (A, B)
end function

[11] **[12]:** You could add a slide before this one reminding the audience that we want to see to what extent the residual bounds from the previous slides might transfer over to the ST Lanczos algorithm.

Setting up a Generalized Eigenvalue Problem II[13]

- Generate a diagonal matrix $D \in \mathbb{R}^{3000\times 3000}$ of eigenvalues in the range $(10^{-3}, 10^{7})$
- Set regularization hyperparameter $\delta = 10^{1}$
- Generate matrices A and B with `GENERATE_MATRIX` function so that

$$\kappa_2(A) = 5.96 \times 10^{11}, \qquad \|A\|_2 = 1.18 \times 10^{11}$$

$$\kappa_2(B) = 8.09 \times 10^{2}, \qquad \text{and} \qquad \|B\|_2 = 1.34 \times 10^{5}.$$

12

- Both matrices are positive definite with their eigenvalues $\Lambda(A, B)$ equal to the diagonal elements of D.

[12] **[13]:** What is the condition of $A - \sigma B$? That is more important than any of the other condition numbers.

[13] **[14]:** I think it is important to include the subranges. The ones that are farther away are lead to a lot of nonconverged Ritz pairs in the graphs, so you might want to point that out.

Relative Residuals I

- Relative Decomposition Residual

$$\frac{\|C_b^T(A-\sigma B)^{-1}C_bQ_n - Q_nT_n - \delta_n \mathbf{q}_{n+1}\mathbf{e}_n^T\|}{\|C_b^T(A-\sigma B)^{-1}C_b\|}$$

- Generalized Relative Residuals

$$\|\tilde{\mathbf{r}}_i\| = \frac{\|(\beta_i A - \alpha_i B)\mathbf{v}_i\|}{(|\beta_i|\|A\| + |\alpha_i|\|B\|)\|\mathbf{v}_i\|}$$

- Spectral Transform Residuals

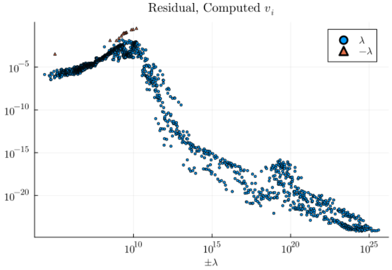
$$\frac{\|C_b^T(A-\sigma B)^{-1}C_b\mathbf{u}_i - \theta_i\mathbf{u}_i\|}{(\|C_b^T(A-\sigma B)^{-1}C_b\| + |\theta_i|)\|\mathbf{u}_i\|}$$

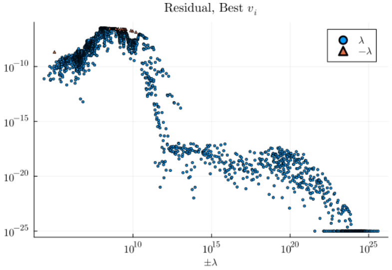
- Best Residuals

$$\frac{\|C_b^T(A-\sigma B)^{-1}C_b\mathbf{u}_i - \theta_i\mathbf{u}_i\|}{(\|C_b^T(A-\sigma B)^{-1}C_b\| + |\theta_i|)\|\mathbf{u}_i\|}$$

Standard Algorithm[14]

Figure: Relative Residual vs. $\pm\lambda$, Standard Algorithm

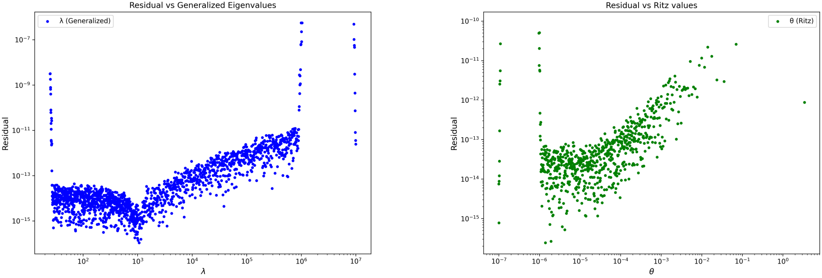




[14] **[15]:** This is a bit awkward, since you didn't run the standard reduction with the Lanczos algorithm and B is actually kind of well conditioned here, so if you ran it for the same problem, you would get excellent results. I'm actually worried about the condition of B. At some point I thought it was closer to 10^5. It needs to be larger than 800 for the experiments to be meaningful.

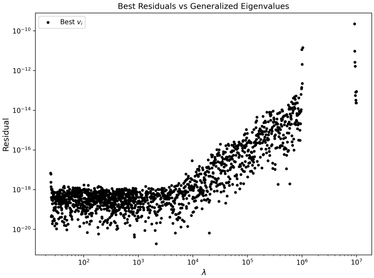
Spectral Transformation Lanczos(LU Decomposition) I

Figure: Residuals plot with moderate shift $\sigma = 1.5 \times 10^3$



(a) (b)

Spectral Transformation Lanczos(LU Decomposition) II



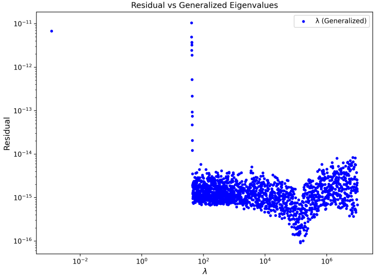
(c)

The computation gave the decomposition residual as 6.63×10^{-11}. Plot (a) is the generalized relative residual with the curve given by $10^{-14}|1 - \lambda_i/\sigma|$. [15] Plot (b) is the relative Ritz residuals. Plot (c) is the best achievable residual for an idealized eigenvector.

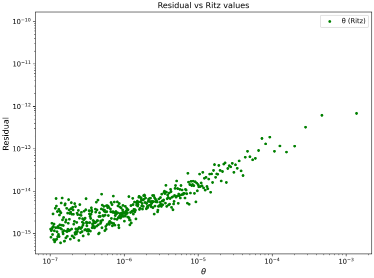
[15] **[16]:** I don't see a curve here. Just the points representing eigenvalue residuals. The curves would be nice to see both here and in the thesis. You can change the factor 10^{-14} to place the curve near the points but not covered by them.

Spectral Transformation Lanczos(LU Decomposition) I

Figure: Residuals plot with large shift $\sigma = 1.5 \times 10^5$

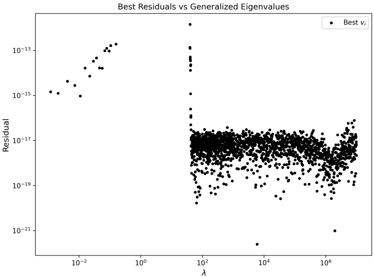


(a)



(b)

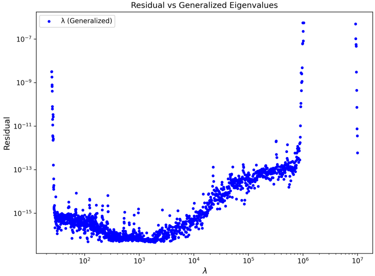
Spectral Transformation Lanczos(LU Decomposition) II



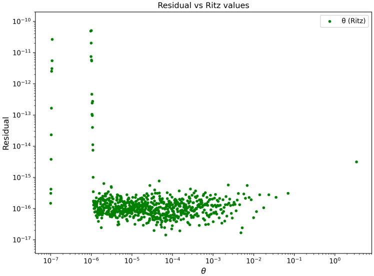
(c)

The computation gave the decomposition residual as 5.42×10^{-12}. Plot (a) is the generalized relative residual with the curve given by $10^{-15}|(1-\lambda_i/\sigma)(1-\sigma/\lambda_i)|$. Plot (b) is the relative Ritz residuals. Plot (c) is the best achievable residual for an idealized eigenvector.

ST Lanczos with Eigenvalue Decomposition I



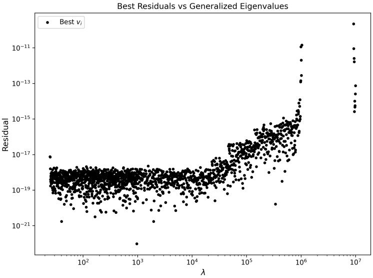
(a)



(b)

Figure: Residuals plot with small shift $\sigma = 1.5 \times 10^3$

ST Lanczos with Eigenvalue Decomposition II



(c)

The computation gave the decomposition residual to the order of 10^{-29}. Plot (a) is the generalized relative residual which shows lower residual to the order of unit round off $u \approx 10^{-16}$ for eigenvalues close the shift as compared to an LU decomposition. Plot (b) is the relative Ritz residuals. Plot (c) is the best achievable residual for an idealized eigenvector.

Pros and Cons[19]

Pros:

- The algorithm is fast for sparse matrices since it uses Lanczos algorithm which is $O(nm^2 + n^2m)$.
- It is efficient in computing a subset of eigenvalues, making it memory efficient.
- ~~Since all the work is done in matrix decompositions that are implemented in LAPACK, the algorithm is almost as fast as the standard method and is easy to implement efficiently, even in a slow language.~~[16]
- With a little effort, it can be designed to handles the case of semidefinite B effectively.
- Delivers really small residuals for symmetric decompositions.

Cons:

- The eigenvector computation is not unconditionally stable.
- ~~Choosing the shift annoying, even if it is relatively easy to choose, especially if one wants a black box algorithm.~~[17][18]

[16] **[17]:** remove