EXPLORING THE BENEFITS OF SKETCHED KRYLOV METHODS IN PRIMME

Heather Switzer
Advisor: Dr. Andreas Stathopoulos
College of William & Mary

April 8, 2024

Abstract

Randomized subspace embedding methods, referred to as "sketching" have become increasingly popular for solving large-scale least-squares problems, reducing the problem size while maintaining solution accuracy. In 2022, Nakatsukasa and Tropp noted that the Rayleigh-Ritz method, used for extracting eigenpair estimates of a matrix A from a subspace, can be reformulated as a least-squares problem, enabling it to be used with sketching.

Krylov-based iterative methods generate a Krylov basis for a matrix A from which these eigenpair approximations can be extracted. However, to achieve accurate approximations, the basis must be orthonormal or close to orthonormal. When seeking many eigenpairs, the Krylov basis will lose orthogonality due to "ghost eigenvectors" appearing in the basis. While reorthogonalization techniques can remedy this, they can also lead to a bottleneck in computation. Sketched Rayleigh-Ritz offers an alleviation to this computational expense, but introduces expenses of its own.

In this work, we implement sketched Rayleigh-Ritz in the PRIMME software library and compare it against its nonsketched counterpart using the Lanczos and Generalized Davidson Krylov methods. This allows us to assess both computational benefits and performance advantages between the two methods.

Introduction

The eigenvalue problem is found across a diversity of scientific disciplines, including spacecraft control systems,[18] computational physics,[29] and machine learning.[4] The eigenproblem considers the equation

$$Ax = \lambda x, \tag{1}$$

where $A \in \mathbb{C}^{n\times n}$ is a square matrix, $x \in \mathbb{C}^n$ is a column vector, and $\lambda \in \mathbb{C}$ is a scalar value. Any (x, λ) pair that satisfies this equation is considered an *eigenpair* of A, with x being an *eigenvector*, and λ being its associated *eigenvalue*.

Using direct methods to compute all eigenpairs of matrix A has a computational complexity of $O(n^3)$, rendering these methods infeasible for large-scale matrices.[30] Most users do not require all eigenpairs of a matrix anyway, and instead only seek an approximate subset. This has led to the development of cost-effective Krylov-based iterative methods, which begin with an initial guess of the eigenvectors of a matrix before refining the approximations iteratively.

Krylov methods build a subspace of the matrix A using repeated applications of A to some initial normal vector(s).[15] This subspace, known as a *Krylov basis*, can then be used in conjunction with methods such as Rayleigh-Ritz (RR) to extract information for solving a system of linear equations or approximating the eigenpairs of a matrix.

Theoretically, this Krylov basis, denoted $V \in \mathbb{C}^{n\times d}$, should always maintain perfect orthogonality. However, in practical applications, this is not the case. Due to having finite precision, once eigenpairs begin converging in the basis, repeated directions, referred to as *ghost eigenvectors*, begin entering, causing degradation in the orthogonality of V and an escalation in the condition number $\kappa(V)$.[24] When $\kappa(V)$ is large, RR may yield inaccurate solutions that result in convergence slowdown or stagnation and heightened numerical instability.[30]

One way to avoid this degradation is by periodically reorthogonalizing the basis using methods such as the QR Decomposition,[10] Householder,[13] and Classical Gram-Schmidt,[5] each with their own accuracy and speed tradeoffs. However, reorthogonalizing a matrix is expensive and may result in a bottleneck in computation if done frequently. This motivated the idea of using randomized subspace embeddings, or "sketching" methods, as they allow extraction of data from a poorly conditioned Krylov basis with minimal loss of accuracy compared to their non-sketched counterparts.[21]

The remainder of this paper is as follows: Sec-

Algorithm 1: The Rayleigh-Ritz Method

Input:
$A \in \mathbb{C}^{n\times n}$ = A Hermitian matrix
$V \in \mathbb{C}^{n\times d}$ = The Krylov basis of A
$H \in \mathbb{C}^{d\times d}$ = $V^\dagger AV$

Output:
$X \in \mathbb{C}^{n\times d}$ = The Ritz vectors of A
$\Lambda \in \mathbb{C}^{d}$ = The Ritz values of A
resNorms $\in \mathbb{C}^{d}$ = The residual norms

`Rayleigh-Ritz(`A, V, H`)`
1 $[Y, \Lambda] =$ `eig(`H`)`;
2 $X = VY$;
`# Compute the residual norms`
3 **for** $i = 1{:}d$ **do**
4 $\quad$ resNorms$(i) = \|AX(:,i) - X(:,i)\Lambda(i)\|_2$
5 **return** $[X, \Lambda,$ resNorms$]$;

tion 2 will discuss background information pertinent to our work, including the RR method, an introduction to Krylov methods, and the intuition behind sketching. Following this, Sections 3 and 4 will discuss using sketched RR with the Lanczos method and provide experimental results. Similarly, Sections 5 and 6 will discuss sketched RR used in conjunction with Generalized Davidson before presenting numerical results. Future work will be outlined in Section 7, before concluding in Section 8.

Background

The Rayleigh-Ritz Method

The Rayleigh-Ritz (RR) method is a mathematical technique used for approximating the eigenpairs of a matrix $A \in \mathbb{C}^{n\times n}$ using a subspace $V \in \mathbb{C}^{n\times d}$. Instead of solving the $n\times n$ eigenproblem $Ax = \lambda x$, RR instead solves the smaller, more managable eigenproblem on the system $V^\dagger AV \in \mathbb{C}^{d\times d}$, yielding eigenpairs (x_i, λ_i) of $V^\dagger AV$. The approximate eigenpairs of A, referred to as the *Ritz pairs*, are then computed as (Vx_i, λ_i) for $i = 1, \cdots, d$. This method is also Galerkin, as it sets the residuals of the estimated eigenpairs orthogonal to the basis V.[8,25] The standard RR method is outlined in Algorithm 1.

Krylov-based Iterative Methods

Krylov methods were initially developed on the insight that a subspace of a matrix $A \in \mathbb{C}^{n\times n}$ can be formed by successive applications of A on some vector y. Information can then be extracted from this subspace, referred to as a Krylov basis.[15] This is written

$$V = \{y, Ay, A^2y, A^3y, \cdots, A^{d-1}y\}, \tag{2}$$

where $y \in \mathbb{C}^n$ represents an initial normal vector, and d denotes the number of columns in the basis, or the *basis size*. Once $V \in \mathbb{C}^{n\times d}$ has been established, RR can be performed by solving $V^\dagger AV\hat{x} = \lambda\hat{x}$, resulting in the Ritz pairs of A, $(V\hat{x}_i, \lambda_i)$ for $i = 1, 2, \cdots, d$.[24]

Several Krylov-based iterative methods have been developed for different problem and matrix types. Arnoldi,[2] Lanczos,[16] LOBPCG,[14] GMRES,[26] and Generalized Davidson[20] are just a few examples. These methods have a common goal of minimizing the residual norms of the sought eigenpairs.

The residual $r \in \mathbb{C}^n$ of an approximate eigenpair $(\hat{x}, \hat{\lambda})$ of A is a measure of how well the estimation satisfies Equation 1 and is computed as

$$r = A\hat{x} - \hat{\lambda}\hat{x}. \tag{3}$$

The norms of the residuals can then be found:

$$\|r\|_2 = \sqrt{r^\dagger r} = \sqrt{\sum_{i=1}^{n} r_i^2}. \tag{4}$$

A smaller residual norm indicates a better eigenpair approximation.

In this work, we restrict ourselves to the Lanczos and Generalized Davidson methods.

2.2.1 The Lanczos Method

The Lanczos algorithm employs a 3-term recurrence to establish a Krylov basis $V \in \mathbb{C}^{n\times d}$ spanning $A \in \mathbb{C}^{n\times n}$. It is particularly useful when dealing with symmetric matrices. While the basis V is being formed, the matrix $H \in \mathbb{C}^{d\times d} = V^\dagger AV$ is also being built. When A is symmetric, H becomes a symmetric, tridiagonal matrix of the form:

$$H = \begin{bmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & \cdots & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{d-1} & \alpha_{d-1} & \beta_d \\ & & & \beta_d & \alpha_d \end{bmatrix} = V^\dagger AV. \tag{5}$$

This form of H is well-suited for use by the RR method, as it circumvents the need for the orthogonalization of V. Lanczos targets the eigenpairs corresponding to the eigenvalues of A situated at the extreme ends of the eigenspectrum.

Algorithm 2 outlines the Lanczos process. On line 12, the new basis vector is constructed through a simple 3-term recurrence without additional orthogonalization. After establishing the Krylov basis, line 13 invokes RR to extract the approximate Ritz pairs of A.

V should always maintain orthogonality in theory. However, as eigenpairs converge, repeated directions of the converged eigenvectors reenter the

Algorithm 2: The Lanczos Algorithm with Rayleigh-Ritz

Input:

$A \in \mathbb{C}^{n \times n}$ = A Hermitian matrix
$y \in \mathbb{C}^{n}$ = Initial normal vector
d = Max Krylov basis size

Output:

$X \in \mathbb{C}^{n \times d}$ = The Ritz vectors of A
$\Lambda \in \mathbb{C}^{d}$ = The Ritz values of A
resNorms $\in \mathbb{C}^{d}$ = The residual norms

`Lanczos(`A, y, d`)`

1 n = `size(`A, 1`)`;
2 H = `zeros(`n, n`)`;
`# Initial Iteration`
3 $V(:,1) = y$;
4 $\hat{w} = AV(:,1)$;
5 $H(1,1) = \hat{w}^T V(:,1)$;
6 $w = \hat{w} - H(1,1)V(:,1)$;
7 **for** $i = 2 : d$ **do**
8 $H(i,i-1) = H(i-1,i) = \|w\|_2$;
9 $V(:,i) = w \ / \ H(i,i-1)$;
10 $\hat{w} = AV(:,i)$;
11 $H(i,i) = \hat{w}^T V(:,i)$;
12 w=$\hat{w}$-$H(i,i)V(:,i)$-$H(i,i\text{-}1)V(:,i\text{-}1)$;
13 [X,Λ,resNorms]=`Rayleigh-Ritz(`A, V, H`)`;
14 **return** [X, Λ, resNorms]

basis, causing convergence degradation or stagnation due to the loss of orthogonality and increased condition number of V. To mitigate this issue, a few strategies practitioners employ are *full-orthogonalization Lanczos* and *restarting*.

Full-orthogonalization Lanczos, as its name suggests, orthogonalizes all new basis vectors against the preceding vectors in the basis, which ensures V remains orthogonal at all times. Referencing Algorithm 2, this would add an additional line after line 12 that uses an orthogonalization technique such as the Classical Gram-Schmidt process.[5]

Another notable expansion of the classic Lanczos algorithm is *restarting*, which is used to overcome challenges related to memory constraints and orthogonalization issues and accelerate convergence in instances of stagnation. Once the number of columns in V reaches a user-defined threshold d and not all desired eigenpairs have converged, restarting is initiated.[25] The standard Lanczos restarting technique discards the entire Krylov basis before restarting Lanczos using a new initial starting vector. This approach may result in a slower convergence due to the loss of convergence information.[11]

Thick-Restarted Lanczos handles this limitation by explicitly restarting the basis with multiple Ritz vectors.[31] In this method, the k Ritz vectors corresponding to the smallest or largest Ritz values, depending on the sought eigenpairs, are preserved, allowing the most significant information in the basis to be retained. k is referred to as the *restart size*, and is defined by the user.

2.2.2 The Generalized Davidson Method

The Davidson algorithm was designed to address large, sparse eigenproblems common in quantum chemistry,[6] and Generalized Davidson (GD) was later introduced to incorporate a basis restarting technique.[20] What sets GD apart from other Krylov methods is how it expands the basis and that it allows the use of preconditioning.

Unlike methods such as Lanczos, GD expands the basis using the residual of the first unconverged eigenpair to "target" it and accelerate its convergence. When introducing this new basis vector, GD orthogonalizes it against all previous vectors using Classical Gram-Schmidt (CGS).[24] If the basis fills up before all eigenpairs are converged, the method restarts with the first k unconverged Ritz vectors, similar to the restarted Lanczos algorithm. It then continues iterating until all estimated eigenvectors are marked as converged, or a user-specified number of the iterations have passed. A more detailed depiction of the GD algorithm is provided in Algorithm 3.

Although this algorithm may entail higher computational costs than methods such as Lanczos, several advantages often justify these expenses. Firstly, GD offers faster convergence, particularly when solving large, sparse eigenproblems.[20] Secondly, it can further enhance convergence rates due to the incorporation of preconditioning. Finally, it exhibits robustness across different problem types, making it applicable to various scientific domains.

Sketching

Randomized Numerical Linear Algebra (RNLA) has grown in popularity due to randomized algorithms offering advantages in speed, reliability, and cost-effectiveness.[17] These algorithms can be applied to various domains, including solving the least-squares problem,[23] preconditioning,[9] singular value decomposition (SVD),[7] and orthogonalization.[3]

One technique within RNLA is *random sketching*, also known simply as *sketching*, which operates as a dimensionality reduction method to lower the computational costs of matrix operations while still providing accurate estimations. Originally, sketching was utilized in solving $n \times d$ least-squares problems, formulated by the equation:

$$\text{minimize}_{y \in \mathbb{C}^d} \|Vy - f\|_2. \tag{6}$$

Algorithm 3: The Generalized Davidson Algorithm

Input:
- $A \in \mathbb{C}^{n \times n}$ = A Hermitian matrix
- $Y \in \mathbb{C}^{n \times \text{numEvals}}$ = Initial vector(s)
- r = Size of the restarted basis
- numEvals = # Evals searching for
- tol = Convergence tolerance

Output:
- $X \in \mathbb{C}^{n \times d}$ = The Ritz vectors of A
- $\Lambda \in \mathbb{C}^{d}$ = The Ritz values of A
- resNorms $\in \mathbb{C}^{d}$ = The residual norms

`Gen_Davidson(`A, Y, r, *numEvals*, *tol*`)`

$V(:, 1 : \text{numEvals}) = Y$;
$W(:, 1 : \text{numEvals}) = AY$;
for $m = 2, 3, \cdots$ **do**
  $W(:, m) = AV(:, m)$;
  $[X, \Lambda]$ = `Rayleigh-Ritz(`A, $V(:, 1 : m)$, $V(:, 1 : m)'W(:, 1 : m)$`)`;
  `# Compute residuals and norms`
  for $i = 1, 2, \cdots, m$ **do**
    resVecs $= WX(:, i) - VX(:, i)\Lambda(i)$
    resNorms$(i) = \|$ resVecs$(:, i)$ $\|_2$
  `# Target 1st unconverged vector`
  t = `find(`resNorms>tol, "first", 1`)`;
  if $t >$ *numEvals* **then**
    return $[X, \Lambda,$ resNorms$]$
  `# Restart basis`
  if $m > d$ **then**
    $V(:, 1 : r) = X(:, 1 : r)$;
    $W(:, 1 : r) = AV(:, 1 : r)$;
    $m = r$;
  `# Precondition new vector before orthogonalizing it against all previous basis vectors`
  `Precondition(`resVecs(:, t)`)`
  V(:,m+1)=`cgs(`V(:, 1:m), resVecs(:, t)`)`;

where $V \in \mathbb{C}^{n \times d}$ with $n \gg d$.[23]

Subsequently, the sketched least-squares problem reformulates Equation 6 into the reduced $s \times d$ problem:

$$\text{minimize}_{y \in \mathbb{C}^d} \|S(Vy - f)\|_2. \tag{7}$$

where $S \in \mathbb{C}^{s \times n}$ is the *sketching matrix* and $n \gg s$.

In a paper by Sarlos,[27] he states that the sketching matrix S is considered a subspace embedding for matrix $A \in \mathbb{C}^{n \times n}$ with distortion factor $\epsilon \in (0, 1)$ if

$$(1 - \epsilon) \cdot \|Ay\|_2 \leq \|SAy\|_2 \leq (1 + \epsilon) \cdot \|Ay\|_2, \tag{8}$$

noting that the optimal size for the embedding dimension is $s \approx \frac{d}{\epsilon^2}$. From this equation, we can also infer that if the original problem yields a small residual, the sketched problem will similarly result in a small residual.

In their 2022 manuscript, Nakastukasa and Tropp noted that RR could also be cast as a least-squares problem:

$$\text{minimize}_{\lambda, x} \|Ax - \lambda x\|_2, \tag{9}$$

presenting an opportunity to apply sketching techniques to Krylov-based iterative methods.[21]

2.3.1 Sparse Maps

The sparse dimension reduction map,[19,22] also known as *Sparse Maps*, is one method used to construct a sketching matrix. Defined as

$$S = \frac{1}{\sqrt{s}}[s_1, s_2, \cdots, s_n] \in \mathbb{C}^{s \times n}, \tag{10}$$

the sketching matrix $S \in \mathbb{C}^{s \times n}$ consists of statistically independent columns, with each column s_i containing exactly z nonzero element, implying that $\texttt{nnz}(S) = z \cdot n$. Each nonzero element of S is drawn from the Steinhaus distribution, which is uniform on the complex unit circle, for a complex matrix, or chosen as ± 1 with 0.5 probability for a real matrix.

The computational cost of applying S to some matrix $A \in C^{n \times n}$ is $O(z * \texttt{nnz}(A))$ if leveraging a software library capable of sparse matrix operations. When constructing a Krylov basis $V \in \mathbb{C}^{n \times d}$, with d being the maximum basis size, we adhere to the conventions outlined in Nakatsukasa and Tropp's manuscript by setting $z = \lceil 2\log(1 + d) \rceil$.[21]

Lanczos with Sketched Rayleigh-Ritz

As mentioned in Section 2, one notable limitation of the Lanczos method is that when searching for

many eigenpairs, the Krylov basis being built will eventually lose orthogonality due to repeated directions appearing in the basis V. Reorthogonalization methods can prevent this but may lead to a bottleneck in computation. This is the motivation behind using Lanczos in conjunction with sketched RR.

Sketched RR (sRR) solves the eigenvalue problem on the sketched system $C^\dagger Dy = \lambda y$, and does not require a fully orthogonal basis to extract information as long as $\kappa(V) < \epsilon_{\texttt{mach}}^{-1}$. Here, $C \in \mathbb{C}^{s\times d}$ is the sketched Krylov basis, SV, and $D \in \mathbb{C}^{s\times d}$ is the sketched projected basis SAV. $\epsilon_{\texttt{mach}}$ refers to the *machine epsilon* of a system, which in most modern systems is $\approx 1E-16$.

Once V begins to lose orthogonality, $\kappa(V)$ will inevitably grow to surpass $\epsilon_{\texttt{mach}}^{-1}$, rendering sRR pointless. There have been two proposed ways of handling this situation. The first method is *whitening*,[23] which computes the QR decomposition of the sketched basis,

$$[U, T] = \texttt{qr}(C). \tag{11}$$

Here, $U \in \mathbb{C}^{s\times d}$ is an orthonormal matrix, and $T \in \mathbb{C}^{d\times d}$ is an upper triangular matrix and $C = UT$. Not only is $\kappa(T) \approx \kappa(C) \approx \kappa(V)$, T can be used to "pseudo-orthogonalize" V, bringing the $\kappa(A)$ back down to ~ 1. The whitened basis $\hat{V}$ is computed

$$\hat{V} = VT^{-1} \tag{12}$$

before proceeding with sRR. Similarly, C must be recomputed as $S\hat{V}$ and D as $SA\hat{V}$.

The second method, known as *stabilization*,[21] computes the truncated SVD[12] of the sketched basis

$$[U_{\texttt{SVD}}, \Sigma_{\texttt{SVD}}, V_{\texttt{SVD}}^\dagger] = \texttt{svd}(C), \tag{13}$$

throwing away all singular triplets (u_i, σ_i, v_i) where $\frac{\sigma_{\max}}{\sigma_i} > \epsilon_{\texttt{mach}}^{-1}$. The eigenvalue problem is then solved on the truncated system

$$U_{\texttt{SVD}}^\dagger D V_{\texttt{SVD}} \hat{y} = \lambda \Sigma_{\texttt{SVD}} \hat{y}. \tag{14}$$

This results in the Ritz pairs of A, $(VV_{\texttt{SVD}}\hat{y}, \Lambda)$. The computational cost between whitening and stabilization is approximately the same.

Preliminary experiments were run to compare the effectiveness of whitening compared to stabilization. Results, shown in Figure 1, indicate that while whitening reduces the condition number of the sketched basis more than stabilization, it also results in convergence stagnation when applied at every instance of sRR while yielding worse Ritz pair approximations than when stabilization is used every time sRR is called.

For this reason, we do not use whitening at all and solely rely on stabilization when computing sRR with $\kappa(V) > \epsilon_{\texttt{mach}}^{-1}$. This decision leads to our version of sRR algorithm, delineated in Algorithm 4.

We implemented Algorithm 4 [1] into the C/C++ high-performance eigensolver library, PRIMME.[28] The Lanczos method, previously not in PRIMME, was implemented per Algorithm 2, and adapted to be run with RR and sRR, depending on the user's input parameters[2]. The only difference between Lanczos with RR and with sRR is that we no longer need to explicitly store the matrix $H \in \mathbb{C}^{d\times d}$, and on line 12, the Rayleigh-Ritz function call is changed to its sketched counterpart.

Lanczos Experiments

Initial tests were conducted in MATLAB to evaluate the efficiency of Thick-Restarted Lanczos with sRR compared to Thick-Restarted Lanczos without sketching. The parameters utilized for this experiment were as follows:

- $A = \texttt{diag}(\sqrt{1:5000})$
- Maximum basis size = 100
- Restart size = 20
- Seeking 10 eigenpairs of largest magnitude
- Convergence tolerance = $\epsilon_{\texttt{mach}}$
- $\epsilon_{\texttt{mach}}^{-1} = 4.5036E+15$

The results of this experiment are shown in Figure 2.

Observation of Figure 2 reveals an immediate stagnation in the convergence of Thick-Restarted Lanczos with sRR. Subsequent analysis revealed that the descent direction of the sketched Ritz vectors used to restart the Krylov basis no longer corresponded to the direction of the actual eigenvectors of A. These results were consistent across various input parameters and matrix types. Consequently, it became apparent that sRR, when paired with the Lanczos algorithm, can not be used when restarting. As a result, all subsequent experiments were conducted using unrestarted Lanczos.

Next, we turned our attention to comparing the time it took sRR with unrestarted Lanczos to run compared to RR with unrestarted Lanczos, as well as the difference in the number of iterations until full convergence was reached. All tests were conducted using 32 MPI processes on one node of the Femto subcluster at William & Mary, where each compute node is a 32-core 960 Xeon Skylake with a clock speed of 2.1GHz.[1] While experiments are still being conducted, some initial results can be seen in Figure 3.

[1] https://tinyurl.com/SketchedRR
[2] https://tinyurl.com/PrimmeLanczos

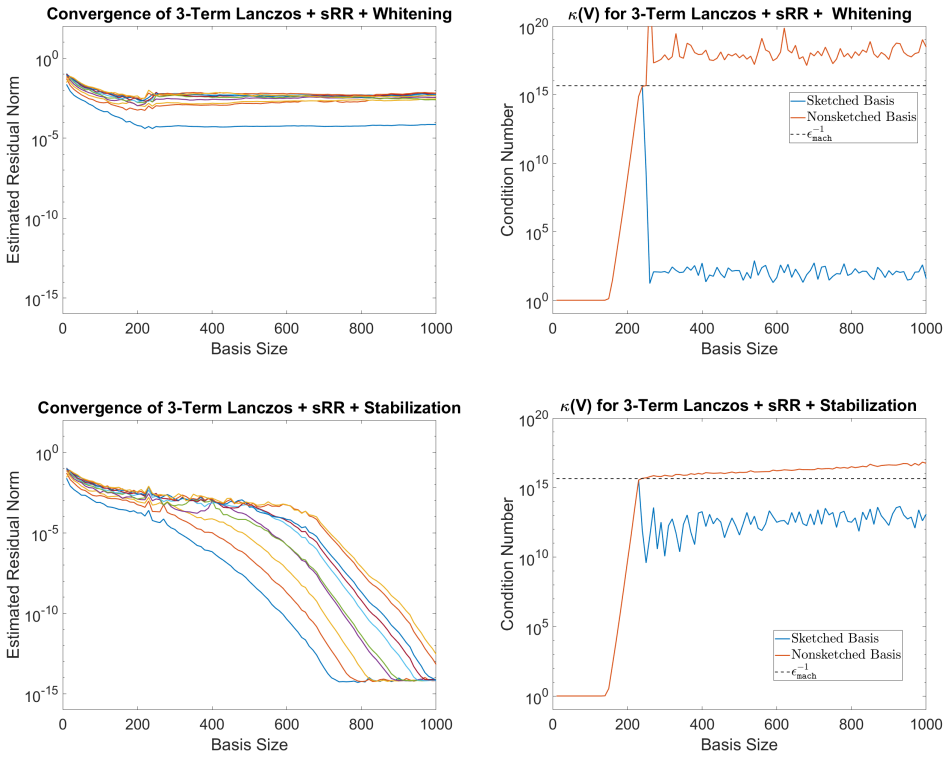

Figure 1: Comparisons of convergence between 3-term Lanczos with sRR utilizing whitening and stabilization to manage the condition number of the basis. **Parameters**: $A = \texttt{diag}(\sqrt{1:5000})$, basis size = 1000, eigenpairs = 10 LM, convergence tolerance = $\epsilon_{\texttt{mach}}$, $\epsilon_{\texttt{mach}}^{-1} = 4.5036\text{E+}15$.

Figure 3 shows the 3-term Lanczos method run with the following parameters:

- $A = diag(1.00001^{[1:1E+6]})$
- Maximum basis size = 3000
- Seeking 100 eigenpairs of largest magnitude
- Convergence tolerance = $1E-6$

While we present results for a specific matrix run, our observations generalize across all matrices and inputs tested. Initially, we note that the convergence rates of unrestarted Lanczos for sketched and nonsketched RR are nearly identical. This consistency was observed across various matrices, sough eigenpairs, and basis sizes.

Although the convergence rates for sketched and nonsketched RR with Lanczos are similar, there is a notable disparity in the runtime. For the specified matrix and basis size, the total runtime was approximately 61 seconds for the nonsketched run, with about 58 seconds spent within the RR function. In contrast, the sketched run took approximately 1,500 seconds, with 439 seconds being spent in the sRR function, while the remaining time was spent maintaining the sketched basis via sparse matrix-vector multiplications.

These observations yield a few conclusions:

- Our method may not be implemented optimally, warranting further investigation into the code
- The performance advantages of sRR might only become apparent when using larger matrices or basis sizes. A comprehensive analysis of the computational complexities of the two methods is necessary.

Generalized Davidson with Sketched Rayleigh-Ritz

Upon realizing the incompatibility of Thick-Restarted Lanczos with sRR, using Generalized Davidson emerged as the next logical step. Given that GD expands the basis with residuals of the unconverged Ritz vectors, maintaining the descent direction and preserving orthogonality is inherent.

Several adjustments were necessary in modifying the base GD code[3] within PRIMME, with specific implementation details to be noted:

1. By default, PRIMME's GD implementation operates as a hybrid method, switching to the JDQMR method once convergence slows. For

[3] https://tinyurl.com/PrimmeDavidson

Algorithm 4: The Sketched Rayleigh-Ritz Method

Input:
$A \in \mathbb{C}^{n \times n}$ = A Hermitian matrix
$V \in \mathbb{C}^{n \times d}$ = The Krylov basis of A
$S \in \mathbb{C}^{s \times n}$ = The sketching matrix

Output:
$X \in \mathbb{C}^{n \times d}$ = The Sketched Ritz vectors
$\Lambda \in \mathbb{C}^{d}$ = The Sketched Ritz values
$\text{resNorms} \in \mathbb{C}^{d}$ = The Sketched residual norms

`Rayleigh-Ritz(`A, V, S`)`

`# Find the QR factors of the sketched basis`

$C = SV$
$D = SAV$
$[U, T] = \texttt{qr}(C, 0)$
if $\texttt{cond}(T) > \epsilon_{\text{mach}}^{-1}$ **then**
  `# Stabilization`
  $[U_{\text{SVD}}, \Sigma_{\text{SVD}}, V_{\text{SVD}}] = \texttt{truncated\_SVD}(C)$;
  $[\hat{Y}, \Lambda] = \texttt{eig}(U_{\text{SVD}}^{\dagger} D V_{\text{SVD}}, \Sigma_{\text{SVD}})$
  $Y = V_{\text{SVD}} \hat{Y}$
else
  $[Y, \Lambda] = \texttt{eig}(U^{\dagger} SAV, T)$;
$\hat{X} = VY$;

`# Normalize the sketched Ritz vectors`

for $i = 1 : d$ **do**
  $X(:, i) = \hat{X}(:, i) / \|\hat{X}(:, i)\|_2$

`# Compute the sketched residuals`

for $i = 1 : d$ **do**
  $\text{resNorms}(i) = \|DY(:, i) - CY(:, i)\Lambda(i)\|_2 \; / \; \|CY(:, i)\|_2$
return $[X, \Lambda, \text{resNorms}]$;

our purposes, this dynamic method was disabled.

2. When sRR is turned on, locking within GD is disabled.

3. Any orthogonalization done within GD is disabled when sRR is turned on, except during the verification of estimated residual norms marked as converged right before returning. During restart, a pseudo-orthogonalization is performed on the restarted basis.

Davidson Experiments

All experiments utilizing GD in PRIMME were again conducted using 32 MPI processes on one node of the Femto subcluster at William & Mary, as detailed in Section 4. Tests were conducted with the following parameters:

- $A = diag(1.00001^{[1:1E+6]})$
- Maximum basis size = 200, 1200, or 2200
- Seeking 1, 100, 500, or 1000 eigenpairs of largest magnitude
- Convergence tolerance = $1E-6$
- Restart size = # of eigenpairs sought

If an experiment exceeded 48 hours, it was forcibly terminated.

Figure 4 illustrates a single run of GD seeking 100 eigenpairs corresponding to the eigenvalues of largest magnitude with and without sketching. It is important to note that both methods were terminated prematurely as their runtimes reached the 48-hour threshold without achieving convergence for all eigenpairs.

Observations derived from this figure remain consistent across all matrices and input parameters tested. Notably, GD with sRR exhibits a slower convergence rate than its nonsketched counterpart. While a higher number of iterations to reach convergence is acceptable if each iteration requires less time, this isn't the case.

To further understand the runtime discrepancy, the average time per iteration is broken down in Figure 5.

Even though GD with sRR consistently required more time per iteration than RR regardless of the number of eigenpairs sought (1, 100, 500, or 1000), the ratio between the two methods remains relatively consistent. Specifically, sRR takes approximately 7.5x more time, most of which is dedicated specifically to the sketched Rayleigh-Ritz function. This suggests that the issue may not lie within the implementation itself, but rather the inherent computational costs of the two methods. Further investigation into this matter is required.

Future Work

While this work remains ongoing, we are actively pursuing various avenues for refinement and exploration. These include:

1. Investigation of the PRIMME code to further optimize the performance of sketching.

2. Conducting a comprehensive computational cost analysis of the Lanczos algorithm with and without sketching, as well as the Generalized Davidson algorithm with and without sketching in PRIMME.

3. Expanding experiments to encompass a wider range of matrices and input parameters to gain a more comprehensive understanding of the methods' behavior.

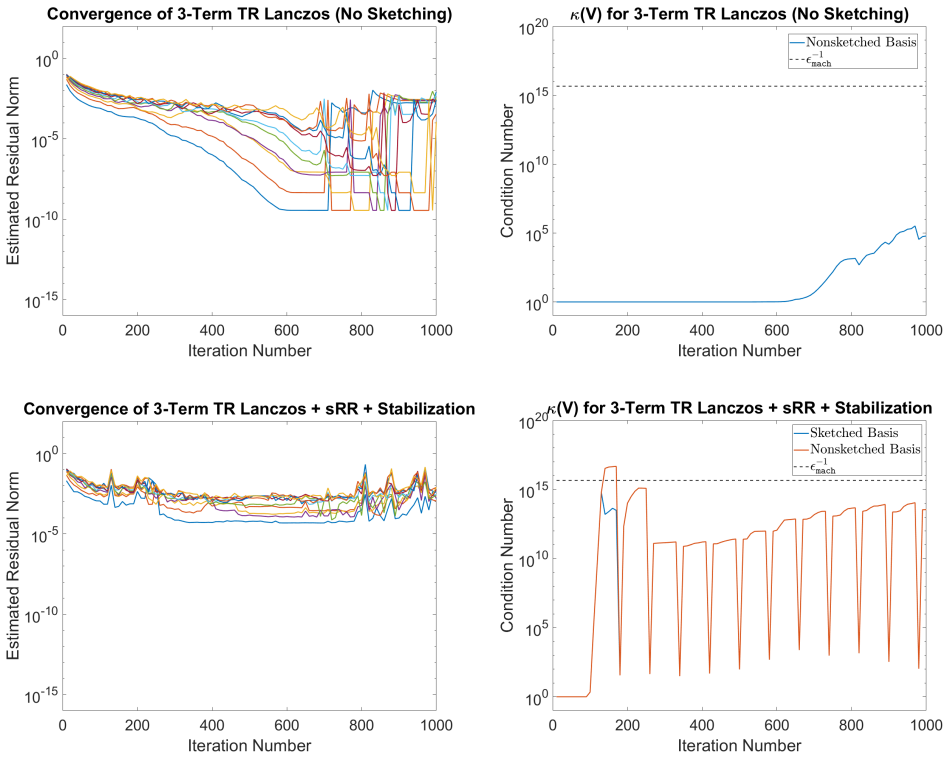

Figure 2: Comparisons of convergence between Thick-restarted 3-term Lanczos with classical RR and sRR.

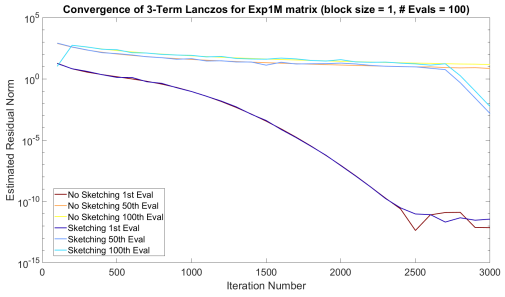

Figure 3: Convergence rates of the 1st, 50th, and 100th eigenpairs

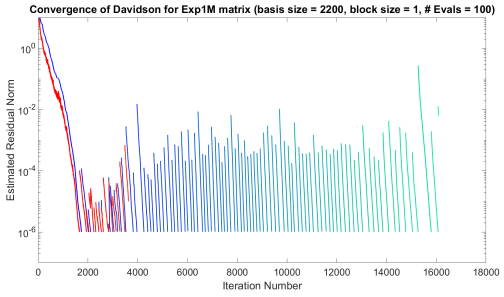

Figure 4: Convergence of *targeted* eigenpairs of GD with and without sketched. Results with sketching are shown in res, while results without sketching are shown in blue.

4. Potential integration sRR with other Krylov-based iterative methods to explore their combined efficiency.

5. Exploration of the use of block methods in conjunction with sketching to potentially increase efficiency and scalability.

Several of these points have already been explored or are currently being analyzed by us, but are not yet ready to be shared.

Conclusion

This work incorporates the sketched Rayleigh-Ritz (sRR) algorithm into the high-performance C/C++ software library PRIMME. PRIMME is designed to utilize iterative techniques for approximating the eigenpairs of a matrix A. Subsequently, we evaluated the efficiency of the sRR method in conjunction with two Krylov methods: Lanczos and Generalized Davidson. The outcomes of these tests were compared against their non-sketched counterparts.

Though this study remains ongoing, several insights have already be emerged. First, when the condition number of the Krylov basis exceeds $\epsilon_{\texttt{mach}}^{-1}$, utilizing sRR with stabilization proved more effective than using sRR with whitening. Sec-

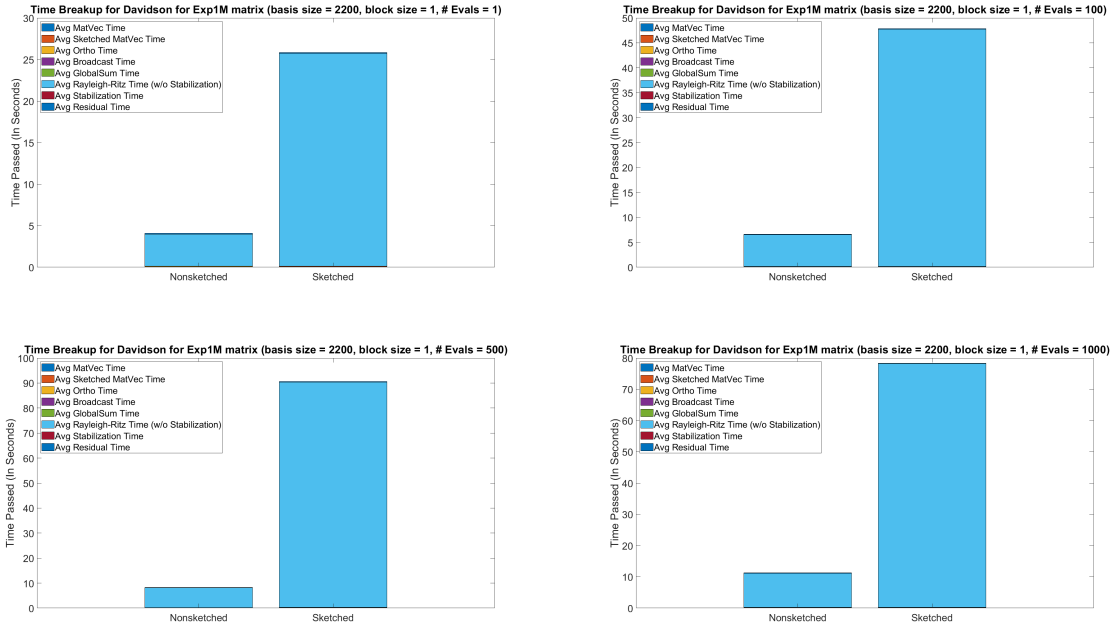

Figure 5: Break-up of where time is spent per iteration on average when seeking 1, 100, 500, and 1000 eigenpairs using GD with and without sketching

ondly, Thick-Restarted Lanczos is incompatible with sRR due to the sketched Ritz vectors no longer corresponding with the descent direction of the actual eigenspace. While unrestarted Lanczos is compatible with sRR, the lack of orthogonalization resulted in repeated directions continuously entering the basis, causing stagnation as stabilization discards these vectors without altering the basis itself.

Generalized Davidson (GD) maintains the basis's low condition number by solely expanding with the residuals of unconverged Ritz pairs, which are inherently orthogonal to the basis. Sketched Ritz pairs can also be used to restart the GD method. However, the number of iterations required for all eigenpairs to converge increases when using sRR without any reduction in runtime.

In future work, we aim to optimize our sRR algorithm within PRIMME and conduct an extensive computational analysis to compare our timing results with the accrued theoretical complexities.

Acknowledgements

This work was funded by the Virginia Space Grant Consortium (VSGC). I am grateful to VSGC for this opportunity and to my advisor, collaborators, and friends for supporting me.

References

[1] Femto.

[2] Arnoldi, W. E. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of Applied Mathematics 9* (1951), 17–29.

[3] Balabanov, O., and Grigori, L. Randomized gram–schmidt process with application to gmres. *SIAM Journal on Scientific Computing 44*, 3 (2022), A1450–A1474.

[4] Belabbas, M.-A., and Wolfe, P. J. Spectral methods in machine learning and new strategies for very large datasets. *Proceedings of the National Academy of Sciences 106*, 2 (2009), 369–374.

[5] Björck, Numerics of gram-schmidt orthogonalization. *Linear Algebra and its Applications 197-198* (1994), 297–316.

[6] Davidson, E. R. The iterative calculation of a few of the lowest eigenvalues and corresponding eigenvectors of large real-symmetric matrices. *Journal of Computational Physics 17*, 1 (1975), 87–94.

[7] Feng, X., Yu, W., and Li, Y. Faster matrix completion using randomized svd. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)* (2018), pp. 608–615.

[8] Fletcher, C. *Computational Galerkin Methods*. Scientific Computation. Springer Berlin Heidelberg, 2012.

[9] Frangella, Z., Tropp, J. A., and Udell, M. Randomized nyström preconditioning. *SIAM Journal on Matrix Analysis and Applications 44*, 2 (2023), 718–752.

[10] Gander, W. Algorithms for the qr decomposition. *Res. Rep 80*, 02 (1980), 1251–1268.

[11] Golub, G. H., and Van Loan, C. F. *Matrix Computations - 4th Edition*. Johns Hopkins University Press, Philadelphia, PA, 2013.

[12] Hansen, P. C. The truncated svd as a method for regularization. *BIT Numerical Mathematics 27* (1987), 534–553.

[13] Householder, A. *Principles of Numerical Analysis*. Dover books on mathematics. McGraw-Hill, 1953.

[14] Knyazev, A. V. Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method. *SIAM Journal on Scientific Computing 23*, 2 (2001), 517–541.

[15] Krylov, A. N. On the numerical solution of the equation by which in technical questions frequencies of small oscillations of material systems are determined. *Izvestija AN SSSR (News of Academy of Sciences of the USSR), Otdel. mat. i estest. nauk 7*, 4 (1931), 491–539.

[16] Lanczos, C. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators.

[17] Martinsson, P.-G., and Tropp, J. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numerica 29* (05 2020).

[18] Meirovitch, L. A new method of solution of the eigenvalue problem for gyroscopic systems. *AiAA Journal 12*, 10 (1974), 1337–1342.

[19] Meng, X., and Mahoney, M. W. Low-distortion subspace embeddings in input-sparsity time and applications to robust linear regression. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing* (2013), pp. 91–100.

[20] Morgan, R. B., and Scott, D. S. Generalizations of davidson's method for computing eigenvalues of sparse symmetric matrices. *Siam Journal on Scientific and Statistical Computing 7* (1986), 817–825.

[21] Nakatsukasa, Y., and Tropp, J. A. Fast accurate randomized algorithms for linear systems and eigenvalue problems, 2022.

[22] Nelson, J., and Nguyên, H. L. Osnap: Faster numerical linear algebra algorithms via sparser subspace embeddings. In *2013 ieee 54th annual symposium on foundations of computer science* (2013), IEEE, pp. 117–126.

[23] Rokhlin, V., and Tygert, M. A fast randomized algorithm for overdetermined linear least-squares regression. *Proceedings of the National Academy of Sciences 105*, 36 (2008), 13212–13217.

[24] Saad, Y. *Iterative Methods for Sparse Linear Systems*, second ed. Society for Industrial and Applied Mathematics, 2003.

[25] Saad, Y. Numerical methods for large eigenvalue problems.

[26] Saad, Y., and Schultz, M. H. Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing 7*, 3 (1986), 856–869.

[27] Sarlos, T. Improved approximation algorithms for large matrices via random projections. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)* (2006), pp. 143–152.

[28] Stathopoulos, A., and McCombs, J. R. Primme: Preconditioned iterative multimethod eigensolver—methods and software description. *ACM Trans. Math. Softw. 37*, 2 (apr 2010).

[29] Thijssen, J. *Computational physics*. Cambridge University Press, 2007.

[30] Trefethen, L. N., and Bau, III, D. *Numerical Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.

[31] Wu, K., and Simon, H. Thick-restart lanczos method for large symmetric eigenvalue problems. *SIAM Journal on Matrix Analysis and Applications 22*, 2 (2000), 602–616.