An improved mixed-precision FEAST algorithm for solving symmetric eigenvalue problems

Yi Xie
College of Computer Science and Technology
National University of Defense Technology
Changsha, China
xieyi00@nudt.edu.cn

Shengguo Li*
College of Computer Science and Technology
National University of Defense Technology
Changsha, China
nudtlsg@nudt.edu.cn

Tiejun Li*
College of Computer Science and Technology
National University of Defense Technology
Changsha, China
tjli@nudt.edu.cn

Meiyue Shao
School of Data Science and MOE Key Laboratory for Computational Physical Sciences
Fudan University
Shanghai, China
myshao@fudan.edu.cn

Ruixuan Ren
College of Computer Science and Technology
National University of Defense Technology
Changsha, China
renruixuan18@nudt.edu.cn

***Abstract*—Solving symmetric eigenvalue problems is vital in many areas of scientific computing. FEAST is a well-known package designed for large-scale eigenvalue problems, incorporating mixed-precision techniques to accelerate linear equation solving. Unlike FEAST's approach, this work introduces a new mixed-precision method that approximates the original eigenvalue problem at a lower precision to quickly provide a good initial guess. These results are then used to accelerate the convergence in working precision. Extensive experiments on large sparse matrices from real applications and randomly generated banded matrices demonstrate the effectiveness of this approach. In appropriate circumstances, our improved mixed-precision FEAST algorithm achieves an average speedup of $1.58\times$ compared to the double-precision FEAST algorithm, with a maximum speedup of $1.79\times$. Additionally, compared to the original mixed-precision approach in the latest FEAST library, our method offers up to a $1.43\times$ performance improvement.**

***Index Terms*—FEAST, Mixed-Precision Algorithm, Symmetric Eigenvalue Problem, Banded Matrix**

I. Introduction

Given two real symmetric or Hermitian matrices A and B, where B is positive definite, it often requires to compute the eigenvalues that lie within a specific interval (a, b) along with their corresponding eigenvectors that satisfy

$$AX = BX\Lambda, \tag{1}$$

where $X = [x_1, \ldots, x_k]$ and $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_k)$. This represents a generalized eigenvalue problem, which plays a crucial role in many fields such as quantum physics [1], information retrieval [2] and structural mechanics [3].

*Corresponding author

The FEAST algorithm is a widely used eigenvalue solver, capable of addressing both standard and generalized eigenvalue problems [4], [5]. Its main computational tasks involve a Rayleigh–Ritz procedure and performing a numerical quadrature, which requires solving independent linear systems along a complex contour, each with multiple right-hand sides [5]. Solving these shifted linear equations during each iteration is the main cost of FEAST; see line 4 of Algorithm 1. Due to its outstanding performance, FEAST is now also applied in fields such as spectral clustering [6] and differential eigenvalue problems [7].

In recent years, mixed-precision techniques have received increasing attention due to recent hardware trends. Hardware vendors have designed many special-purpose units for low-precision arithmetic such as NVIDIA Tensor Cores in GPUs [8] and Intel Advanced Matrix Extensions in CPUs [9]. Lower precision computations result in lower power usage, reduced data movement, and increased computational performance. Many algorithms have achieved substantial benefits when combining with mixed-precision techniques [10], [11]. In the latest FEAST version (4.0), mixed-precision techniques are employed to solve the linear systems in single precision [12], while the Rayleigh–Ritz procedure is performed using double precision arithmetic. This approach aims to lower the computational cost in each iteration; however, it may lead to an increase in the number of iterations.

In this work, we propose a new mixed-precision approach to accelerate the FEAST algorithm. Since the effectiveness of FEAST heavily relies on a good trial subspace, we first perform all the operations in lower precision to quickly compute a good approximate solution, which is then used in

working precision computations, with the aim of accelerating convergence. In summary, our improved algorithm can be divided into two main stages—a pure single-precision stage and a mixed-precision stage. In the pure single-precision stage, the algorithm's two main tasks—Rayleigh–Ritz procedure and solving shifted linear systems—are executed in single precision. The eigenvectors obtained from the single-precision stage are nearly B-orthogonal in double precision, and using them as the basis to construct the initial subspace for the subsequent mixed-precision stage can significantly accelerate the convergence. The mixed-precision stage is performed just as done in FEAST version 4.0 but with a good initial guess from the first stage. Therefore, the number of iterations performed in working precision is reduced and so is the computational time. Our approach works for both general sparse matrices and banded sparse (or dense) matrices. Therefore, it can also be employed to enhance the performance of the hybrid direct eigensolvers proposed in [13] for dense matrices.

The main contributions of this paper are as follows:

1) We propose a new mixed-precision strategy for FEAST, which quickly computes a set of eigenvectors used as initial subspace in a lower precision. The convergence at working precision is accelerated.
2) To evaluate the efficacy of the new algorithm, a series of tests are conducted, including some large sparse matrices from the SuiteSparse [14] and ELSES [15] matrix collections and some randomly generated banded matrices. Compared to the original mixed-precision version in the latest FEAST library, our method can offer up to $1.43\times$ speedup.

The rest of this paper is organized as follows. Section II introduces the fundamental concepts, the algorithmic process of the FEAST algorithm, and some related works. In Section III, we discuss implementation details of the existing mixed-precision FEAST algorithm and present our improved version. Section IV provides numerical experiments demonstrating the efficiency of our proposed approach.

II. Background

A. The FEAST algorithm

The FEAST algorithm utilizes spectral projection and subspace iteration to obtain the eigenpairs of interest [4]. FEAST projects the initial guess $X^{(0)}$ (which can be randomly generated) onto the desired invariant subspace. Subsequently, the Rayleigh–Ritz procedure is employed to extract approximations of eigenvalues and eigenvectors. Different from other projection methods, the projection of FEAST is constructed through a contour integral

$$Q = \rho(A,B)X^{(0)} = \frac{1}{2\pi \mathrm{i}} \oint_{\Gamma} (zB - A)^{-1} B X^{(0)} \, \mathrm{d}z, \quad (2)$$

where $\mathrm{span}(Q)$ is the newly projected subspace, and Γ is a contour enclosing the desired eigenvalues.

If the integration is computed exactly then the filter $\rho(A,B)$ becomes a spectral projector with $\rho(A,B) = XY^{\mathsf{H}}B$, where

Algorithm 1 The FEAST algorithm

Input: The matrices A, B, the search interval (a,b), the initial guess $X^{(0)}$, quadrature nodes and their corresponding weights $\{(z_j, w_j)\}_{j=1,2,\ldots,n_e}$, iteration limit `itmax`.
Output: Eigenpairs (Λ, X) of (A,B) within the interval (a,b).

for $k = 0$ to `itmax` **do**
  Check convergence
  for $j = 1$ to n_e **do**
    $Q_j = (z_j B - A)^{-1} B X^{(k)}$
    $Q = Q + \omega_j Q_j$
  end for
  $A_q = Q^{\mathsf{H}} A Q$, $B_q = Q^{\mathsf{H}} B Q$
  Solve $A_q W_q = B_q W_q \Lambda_q$
  $\Lambda = \Lambda_q$, $X^{(k+1)} = Q W_q$
end for

X and Y are the exact right and left eigenvector matrices associated with the eigenvalues located within the contour Γ. In practice, however, the integration (2) can only be calculated approximately using a quadrature rule. The basis of the subspace generated by the FEAST algorithm is

$$Q = \sum_{j=1}^{n_c} \omega_j (z_j B - A)^{-1} B X^{(0)} = \sum_{j=1}^{n_c} \omega_j Q_j, \quad (3)$$

where z_j's and ω_j's, respectively, are the quadrature nodes and weights, and Q_j's are obtained by solving shifted linear systems with multiple right-hand sides $(z_j B - A)Q_j = BX^{(0)}$. The convergence of FEAST can be enhanced by increasing the number n_c of quadrature nodes in (3). These linear systems are independent of each other and can be solved simultaneously in parallel. FEAST iteratively refines the eigenpairs (Λ and $X^{(k)}$) via the Rayleigh–Ritz procedure. The general framework of FEAST is summarized in Algorithm 1. The iterative process terminates when the residual of the computed eigenpairs satisfies

$$\begin{aligned} \|AX^{(k-1)} - BX^{(k-1)}\Lambda\| &\le \mathtt{tol}, \\ \mathtt{tol} = \epsilon \cdot (\|A\|_2 + \|B\|_2 \cdot \max_i |\lambda_i|) \|X^{(k-1)}\|_2, \end{aligned} \quad (4)$$

where ϵ is a small constant, or a user specified iteration limit is reached.

In summary, the FEAST algorithm is essentially a subspace iteration accelerated by approximate spectral projection [5]. Furthermore, by dividing the complex plane into a few subregions and independently solving for eigenpairs in each region, it achieves high parallelism. However, this work focuses on accelerating the solution process within each subregion by employing mixed-precision techniques. Note that the latest version of FEAST, released in February 2020 (v4.0), already introduces a mixed-precision approach, which is shown in Algorithm 2. The differences between our improved approach and the original method are discussed in detail in the following section.

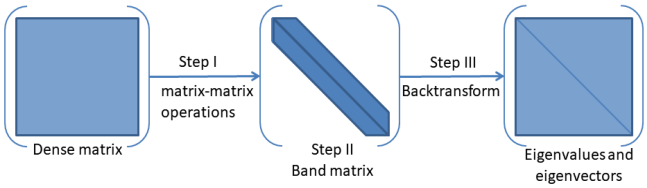


Fig. 1. The main steps of direct eigenvalue method.

B. Direct methods for eigenvalue problems

Dense eigenvalue problems are usually solved by using "direct methods" [16], [17], which consist of *three steps*, just as shown in Figure 1. First, a dense matrix is reduced to a tridiagonal form via orthogonal transformations. One efficient reduction algorithm is the two-stage approach [18], which first reduces a dense matrix to banded form and then further reduces it to a tridiagonal matrix. Then, the eigendecomposition of the tridiagonal matrix is computed by the classical methods such as QR, Divide-and-Conquer (DC) algorithms [19], [16]. Finally, the eigenvectors are obtained via matrix–matrix multiplications.

Although the FEAST algorithm is an iterative eigenvalue method primarily designed for sparse matrices, it has been demonstrated in [13] that FEAST can also accelerate direct eigenvalue algorithms. In this context, FEAST is employed to compute the eigendecomposition of the intermediate banded matrix. Therefore, our improved mixed-precision FEAST algorithm can further improve the performance of the hybrid direct method presented in [13]. We use some banded sparse matrices from SuiteSparse [14] and some randomly generated banded dense matrices in Section IV-B to show the case.

C. Mixed-precision accelerated eigensolvers

In the realm of eigenvalue problems, mixed-precision techniques have also been studied. For dense eigenvalue problems where all eigenpairs are required, existing mixed-precision algorithms are based on either Newton-like iterative refinement [20], [21], [22], [23], [24] or one-sided Jacobi algorithm [25]. In cases where only a few eigenvalues and eigenvectors are required, mixed-precision algorithms have been devised that are based on classical iterative refinement [26] and low-precision preconditioning [27]. Compared to solving linear equations, there are relatively few mixed-precision eigenvalue algorithms, and many of these apply mixed-precision techniques only to the intermediate linear systems involved in the process [28], [10], [12].

III. A New Mixed-Precision FEAST Algorithm

In order to conduct a comprehensive analysis of the concept and algorithmic flow of mixed-precision algorithms, we consider two types of precision. The *working precision* is set to IEEE double precision, which is used to store all input and output data. During the intermediate calculation, certain steps are performed with IEEE single precision, which is referred to as the *lower precision*. The functions `lower`$(\cdot)$ and `working`$(\cdot)$ are used to convert data from working precision to lower precision and from lower precision to working precision, respectively.

Algorithm 2 Original mixed-precision FEAST algorithm

Input: The matrices A, B, the search interval (a, b), the initial guess $X^{(0)}$, quadrature nodes and their corresponding weights $\{(z_j, w_j)\}_{j=1,2,\ldots,n_e}$, iteration limit `itmax`.

Output: Eigenpairs (Λ, X) of (A, B) within the interval (a, b).

1: **for** $k = 0$ to `itmax` **do**
2: Check convergence
3: **for** $j = 1$ to n_e **do**
4: $Q_j = (z_j B - A)^{-1} B X^{(k)}$ (*in single precision*)
5: $Q_{\text{low}} = Q_{\text{low}} + \omega_j Q_j$ (*in single precision*)
6: **end for**
7: $Q = \texttt{working}(Q_{\text{low}})$
8: $A_q = Q^{\mathsf{H}} A Q$, $B_q = Q^{\mathsf{H}} B Q$
9: Solve $A_q W_q = B_q W_q \Lambda_q$
10: $\Lambda = \Lambda_q$, $X^{(k+1)} = Q W_q$
11: **end for**

The computational tasks of FEAST can be roughly divided into two stages: 1) solving internally independent linear systems with multiple right-hand sides, and 2) iteratively updating using the Rayleigh–Ritz procedure. Solving the internally independent linear systems is computationally expensive. Therefore, low-precision operations are implemented in the FEAST library v4.0 for this task (lines 4–5 of Algorithm 2), and then the precision is switched back to double precision for subsequent Rayleigh–Ritz process. This approach ensures significant computation cost reduction per iteration, achieving a speedup of over $1.30\times$ for some general sparse matrices; see Figure 2. Since these linear systems are solved in low-precision, it may lead to an increase of the total number of iteration. For example, the `PENTF20400` matrix in Figure 2 requires two more iterations when using the mixed-precision approach.

The FEAST algorithm is fundamentally a subspace iteration method, and its convergence heavily depends on the choice of the initial subspace $\text{span}(X^{(0)})$. Typically, $X^{(0)}$ consists of randomly generated vectors, which may be far from the desired eigenvectors. Consequently, high accuracy is not required for the linear systems and Rayleigh–Ritz procedures during the first few iterations. By using low-precision arithmetic, we can quickly compute a reasonably good initial subspace at a lower computational cost, with the aim of accelerating the convergence in working precision.

Thus, our new mixed-precision FEAST algorithm consists of two stages: the pure single-precision stage (stage 1) and the mixed-precision stage (stage 2). The pure single-precision stage can be seen as a pre-processing step for the mixed-precision stage. This pre-processing step employs the pure single-precision FEAST algorithm, thereby capitalising on the benefits of low-precision computation and enabling the

Algorithm 3 New mixed-precision FEAST algorithm

Input: The matrices A, B, the search interval (a, b), the initial guess $X^{(0)}$, quadrature nodes and their corresponding weights $\{(z_j, w_j)\}_{j=1,2,\ldots,n_e}$, iteration limit `itmax`.

Output: Eigenpairs (Λ, X) of (A, B) within the interval (a, b).

// pure single-precision stage (stage 1):

1: Generate the initial random subspace $X^{(0)}$
2: Run Algorithm 1 in single precision with a low convergence threshold
3: Construct the trial subspace $\widetilde{X}^{(0)}$

// mixed-precision stage (stage 2):

4: Run Algorithm 2 using the output of **stage 1** ($\widetilde{X}^{(0)}$) as the initial guess

decomposition of matrices when solving linear systems to be saved for reuse in subsequent mixed-precision stages. Most notably, the computational results in this stage are B-orthogonal at low precision, facilitating the rapid capture of eigenpairs by the constructed trial subspace.

The overall structure is illustrated in Algorithm 3. In the pure single-precision stage, all computations, including the linear system solve and Rayleigh–Ritz procedure, are performed using single precision. Additionally, a looser convergence criterion is applied, similar to equation (4), but with ϵ placed with a larger constant $\epsilon_{\text{low}} > \epsilon$, yielding an approximate solution in single precision. These approximate eigenvectors are then used to construct a new trial subspace $\widetilde{X}^{(0)}$, which serves as the initial guess for the mixed-precision stage.

As mentioned earlier, it is crucial to estimate the number of eigenvalues within the search interval beforehand. While there are estimation methods that avoid the costly LDL^{H} decompositions [29], [30], they may be inaccurate, particularly when eigenvalues are clustered near the boundary of the interval or when the number of eigenvalues within the interval is small. The inital choice of M_0, the trial subspace dimension, can often be inaccurate. One potential advantage of our mixed-precision approach is that we can use the the number of eigenpairs obtained from the pure single-precision stage to generate a highly accurate estimate of the subspace dimension for the mixed-precision stage.

As the number of eigenvalues within the search interval increases, the FEAST algorithm usually requires more iterations. In these cases, our approach tends to perform more iterations during the pure single-precision stage, leading to more substantial performance gains. This makes our approach particularly advantageous for matrices which are difficult for the original FEAST algorithm.

In summary, the performance gains of our improved mixed-precision FEAST algorithm arise from two main factors. First, the pure single-precision stage benefits from reduced computational costs due to the use of cheaper operations and a more relaxed convergence criterion. This phase rapidly provides approximate eigenvalues and eigenvectors. The more iterations the original FEAST solver requires, the greater the performance gains achieved by our new mixed-precision FEAST algorithm. Secondly, the output from the pure single-precision stage offers a reasonably accurate approximation of the solution. With this good initial guess, the mixed-precision stage achieves very rapid convergence.

IV. Numerical Experiments

This section presents numerical results of the new mixed-precision algorithm applied to eigenvalue problems involving general sparse and banded matrices.

A. Experiment settings

We use matrices from ELSES [15] to formulate the sparse eigenvalue problem, and MKL-PARDISO to solve the corresponding sparse linear systems. For the symmetric banded matrix experiments, banded matrices from SuiteSparse [14] are employed to construct the eigenvalue problem, with the SPIKE library [31] utilized to solve the linear systems. The following variants of the FEAST algorithm are tested:

1) **FEAST-Double:** FEAST algorithm performed entirely in double precision (Algorithm 1).
2) **FEAST-Mixed:** The original mixed-precision FEAST algorithm (Algorithm 2) which uses single precision for solving the linear systems, and double precision for the rest (mainly the Rayleigh–Ritz procedure).
3) **FEAST-New:** The new mixed-precision FEAST algorithm (Algorithm 3) consists of two stages: a pure single-precision stage and a mixed-precision stage. The results from the pure single-precision stage are used as the initial guess for the mixed-precision stage.

Unless otherwise stated, all experimental calculations are performed on a search interval containing the smallest 200 eigenvalues, and the dimension of the trial subspace M_0 is chosen to be 1.5 times the number of desired eigenvalues, i.e., $M_0 = 300$. The number of quadrature nodes on the upper semicircle is 8, and the number of processes is also 8. The convergence threshold ϵ is set to 10^{-13}. In our mixed-precision algorithm, the stopping threshold ϵ_{low} for the pure single-precision stage is set to 10^{-6}. All tests are performed on a server equipped with two 12-core Intel Xeon E5-2692v2 3.0 GHz CPUs and 128 GB of RAM.

B. General sparse matrices

This subsection investigates the effectiveness of the new mixed-precision FEAST algorithm in solving the eigenvalue problem with sparse matrices. We utilize sparse matrices from ELSES with dimensions spanning from 10,000 to 100,000 for the formulation of both generalized and standard eigenvalue problems, the matrix information is shown in Table I.

Figure 2 shows the relative runtime of the two mixed-precision algorithms. The left figure illustrates the generalized eigenvalue problem, and the right figure depicts the standard eigenvalue problem. Data labels on FEAST-Mixed and FEAST-New indicate the speedup achieved by these two methods compared to FEAST-Double. The values in each bar chart

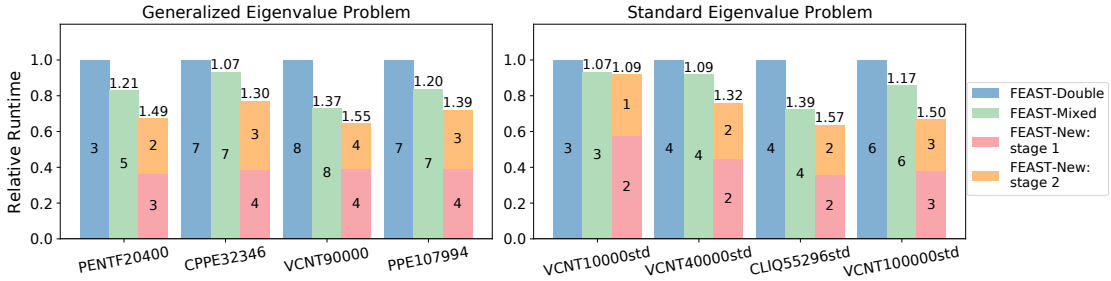


Fig. 2. Relative runtime of two mixed-precision algorithms compared to the double-precision one. Data labels on FEAST-Mixed and FEAST-New indicate the speedup achieved by these methods relative to FEAST-Double. The values shown in each bar chart correspond to the number of iterations for the specified stage.

TABLE I
INPUT GENERAL SPARSE MATRICES FROM ELSES [15].

Matrix	Rows	#nnz	#nnz/rows
PENTF20400	20 K	1.84 M	90
CPPE32346	32 K	0.86 M	27
VCNT90000	90 K	17.53 M	194
PPE107994	11 K	2.71 M	25
VCNT10000std	10 K	0.21 M	21
VCNT40000std	40 K	0.82 M	21
CLIQ55296std	55 K	0.85 M	15
VCNT100000std	100 K	2.05 M	21

represent the number of iterations for the corresponding stage. In the FEAST-New bar chart, the red and orange segments reflect the relative runtime of the pure single-precision stage and the mixed-precision stage, respectively.

As illustrated in Figure 2, the new mixed-precision algorithm exhibits superior performance compared to the double-precision algorithm in both generalized and standard eigenvalue problems. It demonstrates improved efficiency compared to the original mixed-precision algorithm in a majority of scenarios, when solving the generalized eigenvalue problem consisting of PENTF20400, the maximum speedup is $1.23\times$ (the ratio of the two speedups, $1.49/1.21 = 1.23$). Similarly, when solving the standard eigenvalue problem involving VCNT100000std, the maximum speedup is $1.28\times$.

The advantage of the new mixed-precision algorithm comes from iteration cycles performed by the pure single-precision stage. Consequently, when the number of iterations is limited, the advantages of the new mixed-precision FEAST algorithm may not be fully realized. When solving the standard eigenvalue problem with VCNT10000std, the iteration count of the original mixed-precision algorithm is only 3. In such a circumstance, the performance improvement offered by the new mixed-precision algorithm is not particularly pronounced.

The original mixed-precision FEAST method uses low precision to solve linear systems, which improves speed per iteration, but may result in slower convergence (i.e., more iterations). When solving the generalized eigenvalue problem with PENTF20400, the double-precision FEAST algorithm requires only 3 iterations to reach convergence accuracy, while the original mixed-precision FEAST algorithm requires 5

TABLE II
INPUT BANDED SPARSE MATRICES FROM SUITESPARSE [14].

Matrix	Rows	#nnz	Semi-bandwidth
cz5108	5.10 K	51 K	267
fv3	9.80 K	87 K	100
cz10228	10 K	102 K	369
cz20468	20 K	206 K	514
chem_master1	40 K	201 K	201
cz40948	41 K	412 K	721

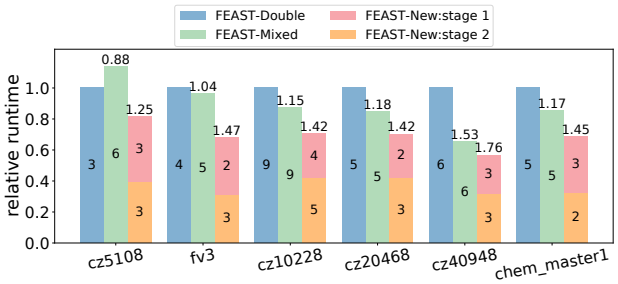


Fig. 3. Relative runtime of two mixed-precision algorithms compared to the double-precision algorithms. The eigenvalue problem is posed by banded matrices in SuiteSparse, with dimensions ranging from 5108 to 40948.

iterations. The new mixed-precision FEAST algorithm also benefits from low-precision computations without incurring additional iteration cycles, achieving a performance improvement of $1.49\times$.

1) Performance Comparison: In addition to the extensive range of applications directly involving banded matrices, dense symmetric eigenvalue problems can also be transformed into banded ones. Therefore, we also test banded matrices. A series of matrices derived from practical problems within SuiteSparse are selected for testing, with dimensions ranging from 5108 to 40948. Information of the test matrices is shown in Table II.

As depicted in Figure 3, our new approach demonstrates superior performance compared to the original mixed-precision FEAST algorithm across all test matrices. When solving the eigenvalue problem with fv3, the maximum speedup of the new mixed-precision algorithm in comparison to the original mixed-precision algorithm is attained, reaching $1.42\times$.

It is noteworthy that a unique circumstance arose during the

TABLE III
RANDOMLY GENERATED SYMMETRIC BANDED DENSE MATRICES USED IN OUR EVALUATION.

Matrix	Rows	#nnz	Semi-bandwidth
N5000_B32	5000	0.32 M	32
N5000_B64	5000	0.64 M	64
N5000_B128	5000	1.27 M	128
N5000_B160	5000	1.58 M	160
N5000_B224	5000	2.19 M	224
N5000_B256	5000	2.50 M	256

resolution of the eigenvalue problem represented by matrix cz5108. The original mixed-precision algorithm is found to be slower than the double-precision FEAST algorithm, with a speedup of only $0.88\times$. This phenomenon can be attributed to two primary factors. Firstly, the low-precision linear system solution results in an increased number of iterations. Secondly, the initial estimate, which is generated randomly, is found to be unreasonable and results in slower convergence. The combination of these two factors results in a negative improvement in the original mixed-precision algorithm. The new mixed-precision algorithm is capable of producing a more reasonable initial estimate as a consequence of the single-precision stage calculation results, thereby mitigating the negative impact.

2) The impact of bandwidth: In the context of banded matrices, the bandwidth exerts a significant influence on the solution of the eigenvalue problem, particularly in the case of iterative methods, which directly impact the number of required iteration cycles. To explore the effect of bandwidth, we randomly generated a set of symmetric banded dense matrices with dimension 5000 and semi-bandwidth sizes ranging from 32 to 256. The SPIKE library [31] is used to solve these banded linear systems. Information for these test matrices is shown in Table III.

Figure 4 shows the relative runtime and speedup of two types of mixed-precision algorithms in comparison to the pure double-precision algorithm, it also documents the iteration cycles of all algorithms. The comparative advantage of the new mixed-precision algorithm lies in the number of iterations in the pure single-precision stage. When the number of iterations of the original mixed-precision algorithm is greater than 4, the new mixed-precision algorithm can achieve an average speedup of $1.58\times$ times and a maximum speedup of $1.79\times$ over the pure double-precision version. Meanwhile, our approach can achieve an average speedup of $1.30\times$ compared with the original mixed-precision algorithm, and the maximum speedup of $1.43\times$ when solving the eigenvalue problem composed of the matrix N5000_B64.

However, it can also be observed that when the bandwidth is 32, our approach shows a negative improvement compared to the original mixed-precision algorithm. This is because the original mixed-precision algorithm requires fewer iterations due to the small bandwidth. At this point, the performance improvement of the pure single-precision stage cannot outweigh the cost of the additional operations, resulting in a negative

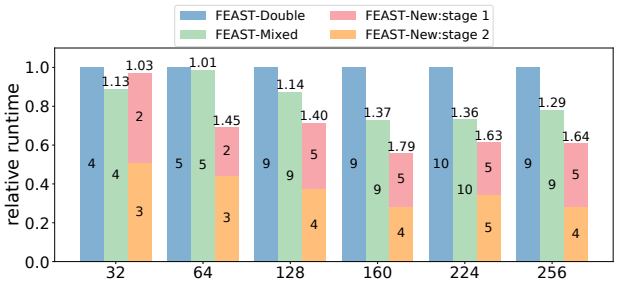


Fig. 4. Relative runtime of two mixed-precision algorithms compared to the double-precision algorithms. We used a set of matrices with a dimension of 5000 and different bandwidths for the experiment, the x-axis represents six different settings of semi-bandwidth, spanning a range from 32 to 256.

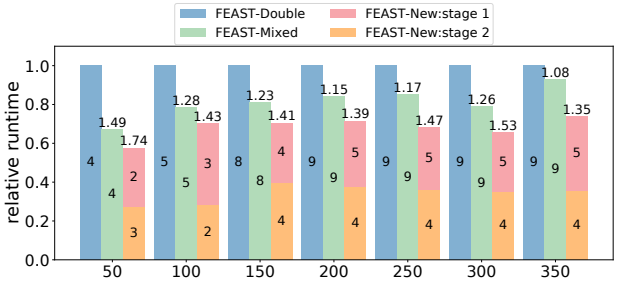


Fig. 5. Relative runtime of two mixed-precision algorithms compared to the double-precision algorithms. Taking the matrix N5000_B128 as an example, seven cases with the estimated number of eigenvalues ranging from 50 to 350 are selected to explore the impact of the search interval size.

performance improvement. As the bandwidth increases, the number of iterations required to solve the eigenvalue problem also increases correspondingly. In this case, the issue is effectively mitigated, indicating that our approach performs better for matrices with larger bandwidths.

C. The number of desired eigenvalues

The size of the search interval has a significant impact on the performance of the FEAST algorithm. An overly large search interval undermines the advantage of its high parallelism, whereas a search interval that is too small will lead to an exacerbation of the negative impact of inaccurate eigenvalue estimation. To obtain a comprehensive assessment of the performance of FEAST-New, it is essential to consider a range of search interval sizes.

We take the banded matrix N5000_B128 from Section IV-B to calculate eigenpairs within the interval encompassing the estimated 50 to 350 eigenvalues. Figure 5 illustrates the relative runtime, speedup, and iteration cycles of two types of mixed-precision algorithms compared to the pure double-precision FEAST algorithm.

As illustrated in Figure 5, the new mixed-precision algorithm outperforms both the pure double-precision FEAST and the original mixed-precision FEAST in all intervals. Compared to the original mixed-precision algorithm, our approach achieves an average speedup of $1.19\times$. As the search interval increases, the performance improvement of our method over the original mixed-precision algorithm demonstrates an overall

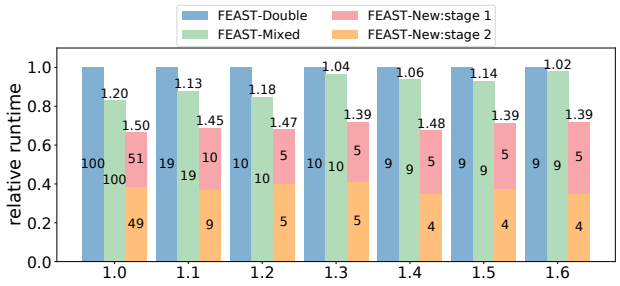


Fig. 6. Relative runtime of two mixed-precision algorithms compared to the double-precision algorithms. The x-axis represents the seven cases where M_0 is 1.0 to 1.6 times the estimated number of eigenvalues.

upward trend, reaching a maximum speedup of $1.26\times$ in the interval estimated to contain 350 eigenvalues.

Additionally, we note that for the interval with an estimated 50 eigenvalues, our approach executes 2 and 3 iterations in the pure single-precision and mixed-precision stages, respectively, totaling one more iteration than the original mixed-precision algorithm. Despite this, the new mixed-precision algorithm achieved a $1.17\times$ speedup due to the advantages of single-precision computation and accurate construction of the initial subspace in the mixed-precision stage.

D. The subspace dimension M_0

FEAST, like other iterative subspace algorithms, needs to choose an appropriate iterative subspace dimension M_0 to ensure that the eigenvalues in the interval can be captured quickly [3]. If M_0 is too small, the algorithm will not converge. However, if the subspace is too large, it will contain more irrelevant information, leading to slower convergence. Numerical analysis of FEAST recommends selecting a subspace dimension about 1.5 times the number of eigenvalues in the interval [5]. Our method, by constructing a pure single-precision stage, utilizes the accurate approximation of the desired subspace to construct the initial trial subspace for the mixed-precision stage. This allows for a reduction in the subspace dimension, thereby enhancing the performance. In this experiment, we take the generated banded matrix `N5000_B128` as an example with an interval estimated to contain 200 eigenvalues, and M_0 is 1.0 to 1.6 times the number of eigenvalues.

Figure 6 shows the relative runtime of two mixed-precision algorithms compared to the pure double-precision FEAST algorithm, as well as the numbers of iterations. When M_0 is relatively small, reaching only 1.0 and 1.1 times the estimated number of eigenvalues in the interval, the initial subspace needs to be close to the space spanned by the desired eigenvectors. However, the randomly generated subspace $\mathrm{span}(X^{(0)})$ often fails to meet this requirement effectively, resulting in more iterations and longer calculation times. Across multiple sets of matrices we have tested, the shortest computation times were typically observed when M_0 was 1.3 to 1.6 times the estimated number of eigenvalues. This aligns with Tang's recommendation of using 1.5 times the estimated number of eigenvalues [5].

It can be observed that the new mixed-precision algorithm consistently outperforms both FEAST-Double and the original mixed-precision algorithm across all M_0 settings. On average, the new algorithm achieves a $1.24\times$ speedup compared to the original mixed-precision algorithm, with a maximum speedup of $1.33\times$ when M_0 is set to 1.3 times the estimated number of eigenvalues. This indicates that our approach allows for greater flexibility in choosing M_0.

E. The number of processes

FEAST, as a type of parallel algorithm, allows the solution of linear equations to be performed in parallel at each quadrature node. Therefore, investigating the performance of new mixed-precision algorithms in multi-process situations is also an important factor. In this section, the effect of different numbers of processes on the algorithm is investigated.

We select the Gauss–Legendre quadrature rule with eight nodes on the upper semicircle to approximate the contour integral. Three matrices with different bandwidths are employed to analyze the impact of the number of processes: `N5000_B32`, `N5000_B64`, and `N5000_B128`. Figure 7 documents the performance improvements of the two mixed-precision algorithms compared to FEAST-Double.

It can be observed that while both mixed-precision algorithms demonstrate notable performance enhancements in comparison to the original FEAST algorithm, the relative speedup over FEAST-Double diminishes to a certain extent as the number of processes increases. This decline is more pronounced for the original mixed-precision algorithm. Consequently, as the number of processes increases, the performance improvement of the new mixed-precision algorithm compared to the original mixed-precision algorithm demonstrates an upward trend, with a maximum speedup of $1.3\times$ and an average speedup of $1.23\times$.

V. Conclusion

In this paper, we propose a new mixed-precision FEAST algorithm that enhances the existing mixed-precision algorithm by introducing a pure single-precision stage. In this stage, both linear system solving and iterative updates are performed at a lower precision, significantly boosting performance with minimal impact on convergence. Using the results from the pure single-precision stage, our algorithm constructs a more efficient and accurate subspace as the initial guess for subsequent stages, further reducing the execution time. Numerical experiments show that the new mixed-precision method produces excellent results for both banded and sparse matrices. The advantage of our approach stems from the pure single-precision stage, making it particularly well-suited for problems requiring numerous iteration cycles, such as when dealing with large matrix dimensions and a significant number of eigenvalues within the interval. For banded matrices, the performance gains are even more pronounced for matrices

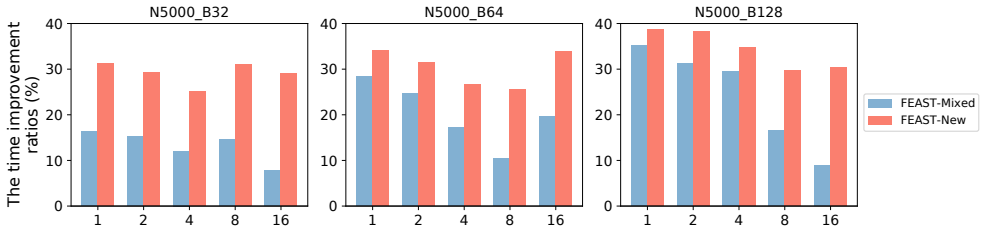


Fig. 7. The time saved by the two mixed-precision algorithms compared to the double-precision algorithm when varying the number of processes. Experiments are carried out on three matrices `N5000_B32`, `N5000_B64`, and `N5000_B128`, where the number of quadrature nodes on the upper semicircle is 8 and the processes are evenly distributed on each quadrature node. The x-axis is the number of processes.

with larger bandwidths. At the same time, our method performs better in situations with more parallel resources and can be adapted to the practical application environment of high performance computing.

ACKNOWLEDGMENTS

The authors would like to thank the referees for their valuable comments. Y. Xie, S. Li, T. Li, and R. Ren were supported in part by the National Key R&D Program of China under grant agreement 2023YFB3001702. M. Shao was supported in part by the National Key R&D Program of China under grant agreement 2023YFB3001603.

REFERENCES

[1] N. Marzari, A. Ferretti, and C. Wolverton, "Electronic-structure methods for materials design," *Nat. Mater.*, vol. 20, no. 6, pp. 736–749, 2021.
[2] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Soc. Inf. Sci.*, vol. 41, pp. 391–407, 1990.
[3] Y. Saad, *Numerical Methods for Large Eigenvalue Problems: Revised Edition*. Philadelphia, PA, USA: SIAM, 2011.
[4] E. Polizzi, "Density-matrix-based algorithm for solving eigenvalue problems," *Phys. Rev. B*, vol. 79, no. 11, p. 115112, 2009.
[5] P. T. P. Tang and E. Polizzi, "FEAST as a subspace iteration eigensolver accelerated by approximate spectral projection," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 2, pp. 354–390, 2014.
[6] S. Mdaa, A. Ouali Alami, R. Guivarch, and S. Mouysset, "Parallel spectral clustering with FEAST library," in *Advanced Research in Technologies, Information, Innovation and Sustainability (ARTIIS 2022)*, T. Guarda, F. Portela, and M. Augusto, Eds. Cham, Switzerland: Springer, 2022, pp. 127–138.
[7] A. Horning and A. Townsend, "FEAST for differential eigenvalue problems," *SIAM J. Numer. Anal.*, vol. 58, no. 2, pp. 1239–1262, 2020.
[8] A. Haidar, H. Bayraktar, S. Tomov, J. Dongarra, and N. J. Higham, "Mixed-precision iterative refinement using tensor cores on GPUs to accelerate solution of linear systems," *Proc. Royal Soc. Lond.*, vol. 476, no. 2243, p. 20200110, 2020.
[9] K. Halbiniak, K. Rojek, S. Iserte, and R. Wyrzykowski, "Unleashing the potential of mixed precision in AI-accelerated CFD simulation on Intel CPU/GPU architectures," in *Computational Science—ICCS 2024*, L. Franco, C. de Mulatier, M. Paszynski, V. V. Krzhizhanovskaya, J. J. Dongarra, and P. M. A. Sloot, Eds. Cham, Switzerland: Springer, 2024, pp. 203–217.
[10] A. Abdelfattah, H. Anzt, E. G. Boman, E. Carson, T. Cojean, J. Dongarra, A. Fox, M. Gates, N. J. Higham, X. S. Li *et al.*, "A survey of numerical linear algebra methods utilizing mixed-precision arithmetic," *Int. J. High Perform. Comput. Appl.*, vol. 35, no. 4, pp. 344–369, 2021.
[11] N. J. Higham and T. Mary, "Mixed precision algorithms in numerical linear algebra," *Acta Numer.*, vol. 31, pp. 347–414, 2022.
[12] E. Polizzi, "FEAST eigenvalue solver v4.0 user guide," *arXiv preprint arXiv:2002.04807*, 2020.
[13] S. Li, X. Wu, J. E. Roman, Z. Yuan, R. Wang, and L. Cheng, "A parallel direct eigensolver for sequences of Hermitian eigenvalue problems with no tridiagonalization," *arXiv preprint arXiv:2012.00506*, 2020.
[14] T. A. Davis and Y. Hu, "The University of Florida sparse matrix collection," *ACM Trans. Math. Software*, vol. 38, no. 1, pp. 1–25, 2011.
[15] T. Hoshi, H. Imachi, A. Kuwata, K. Kakuda, T. Fujita, and H. Matsui, "Numerical aspect of large-scale electronic state calculation for flexible device material," *Jpn J. Indust. Appl. Math.*, vol. 36, pp. 685–698, 2019.
[16] J. Demmel, *Applied Numerical Linear Algebra*. Philadelphia, PA, USA: SIAM, 1997.
[17] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. Van der Vost, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*. Philadelphia, PA, USA: SIAM, 2000.
[18] C. Bischof, B. Lang, and X. Sun, "Parallel tridiagonalization through two-step band reduction," in *Proceedings of the Scalable High-Performance Computing Conference*. Los Alamitos, CA, USA: IEEE, 1994, pp. 23–27.
[19] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD, USA: The Johns Hopkins University Press, 1996.
[20] Z. Bujanović, D. Kressner, and C. Schröder, "Iterative refinement of Schur decompositions," *Numer. Algorithms*, vol. 92, no. 1, pp. 247–267, 2023.
[21] T. Ogita and K. Aishima, "Iterative refinement for symmetric eigenvalue decomposition," *Jpn J. Indust. Appl. Math.*, vol. 35, pp. 1007–1035, 2018.
[22] ——, "Iterative refinement for symmetric eigenvalue decomposition II: clustered eigenvalues," *Jpn J. Indust. Appl. Math.*, vol. 36, pp. 435–459, 2019.
[23] ——, "Iterative refinement for symmetric singular value decomposition based on matrix multiplication," *J. Comput. Appl. Math.*, vol. 369, p. 112512, 2020.
[24] Y. M. Tsai, P. Luszczek, and J. Dongarra, "Mixed-precision algorithm for finding selected eigenvalues and eigenvectors of symmetric and hermitian matrices," in *2022 IEEE/ACM Workshop on Latest Advances in Scalable Algorithms for Large-Scale Heterogeneous Systems (ScalAH)*. IEEE, 2022, pp. 43–50.
[25] W. Gao, Y. Ma, and M. Shao, "A mixed precision Jacobi SVD algorithm," *arXiv preprint arXiv:2209.04626*, 2022.
[26] J. J. Dongarra, "Algorithm 589: SICEDR: A FORTRAN subroutine for improving the accuracy of computed matrix eigenvalues," *ACM Trans. Math. Software*, vol. 8, no. 4, pp. 371–375, 1982.
[27] D. Kressner, Y. Ma, and M. Shao, "A mixed precision LOBPCG algorithm," *Numer. Algorithms*, vol. 94, pp. 1653–1671, 2023.
[28] A. Alvermann, A. Basermann, H.-J. Bungartz, C. Carbogno, D. Ernst, H. Fehske, Y. Futamura, M. Galgon, G. Hager, S. Huber *et al.*, "Benefits from using mixed precision computations in the ELPA-AEO and ESSEX-II eigensolver projects," *Jpn J. Indust. Appl. Math.*, vol. 36, pp. 699–717, 2019.
[29] L. Lin, Y. Saad, and C. Yang, "Approximating spectral densities of large matrices," *SIAM Rev.*, vol. 58, no. 1, pp. 34–65, 2016.
[30] E. Di Napoli, E. Polizzi, and Y. Saad, "Efficient estimation of eigenvalue counts in an interval," *Numer. Linear Algebra Appl.*, vol. 23, no. 4, pp. 674–692, 2016.
[31] E. Polizzi and A. H. Sameh, "A parallel hybrid banded system solver: the SPIKE algorithm," *Parallel Comput.*, vol. 32, no. 2, pp. 177–194, 2006.