Accurate Inverses for Computing Eigenvalues of Extremely Ill-conditioned Matrices and Differential Operators *

Qiang Ye[†]

Abstract

This paper is concerned with computations of a few smallest eigenvalues (in absolute value) of a large extremely ill-conditioned matrix. It is shown that a few smallest eigenvalues can be accurately computed for a diagonally dominant matrix or a product of diagonally dominant matrices by combining a standard iterative method with the accurate inversion algorithms that have been developed for such matrices. Applications to the finite difference discretization of differential operators are discussed. In particular, a new discretization is derived for the 1-dimensional biharmonic operator that can be written as a product of diagonally dominant matrices. Numerical examples are presented to demonstrate the accuracy achieved by the new algorithms.

1 Introduction

In this paper, we are concerned with accurate computations of a few smallest eigenvalues (in absolute value) of a large extremely ill-conditioned matrix. Here, we consider a matrix A extremely ill-conditioned if $\mathbf{u}\kappa_2(A)$ is almost of order 1, where $\mathbf{u}$ is the machine roundoff unit, $\kappa_2(A) := \|A\|\|A^{-1}\|$ is the spectral condition number of A, and $\|\cdot\|$ is the 2-norm. We are mainly interested in large sparse matrices arising in discretization of differential operators, which may lead to extremely ill-conditioned matrices when a very fine discretization is used. In that case, existing eigenvalue algorithms may compute a few smallest eigenvalues (the lower end of the

*2010 Mathematics Subject Classification: 65F15, 65F35, 65N06, 65N25. Key words: Eigenvalue; ill-conditioned matrix; accuracy; Lanczos Method; differential eigenvalue problem, biharmonic operator.

[†]Department of Mathematics, University of Kentucky, Lexington, KY 40506. qye3@uky.edu.
Research supported in part by NSF Grants DMS-1317424, DMS-1318633 and DMS-1620082.

spectrum) with little or no accuracy owing to roundoff errors in a floating point arithmetic.

Consider an $n \times n$ symmetric positive definite matrix A and let $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be its eigenvalues. Conventional dense matrix eigenvalue algorithms (such as the QR algorithm) are normwise backward stable, i.e., the computed eigenvalues $\{\widehat{\lambda}_i\}$ in a floating point arithmetic are the exact eigenvalues of $A + E$ with $\|E\| = \mathcal{O}(\mathbf{u})\|A\|$; see [38, p.381]. Here and throughout, $\mathcal{O}(\mathbf{u})$ denotes a term bounded by $p(n)\mathbf{u}$ for some polynomial $p(n)$ in n. Eigenvalues of large (sparse) matrices are typically computed by an iterative method (such as the Lanczos algorithm), which produces an approximate eigenvalue $\widehat{\lambda}_i$ and an approximate eigenvector $\widehat{x}_i$ whose residual satisfies $\|A\widehat{x}_i - \widehat{\lambda}_i\widehat{x}_i\|/\|\widehat{x}_i\| < \eta$ for some threshold η. Since the roundoff error occurring in computing $A\widehat{x}_i$ is of order $\mathbf{u}\|A\|\|\widehat{x}_i\|$, then this residual can converge at best to $\mathcal{O}(\mathbf{u})\|A\|$. Then, for both the dense and iterative eigenvalue algorithms, the error of the computed eigenvalue $\widehat{\lambda}_i$ is at best $|\widehat{\lambda}_i - \lambda_i| \leq \mathcal{O}(\mathbf{u})\|A\| = \mathcal{O}(\mathbf{u})\lambda_n$. Thus

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{\lambda_i} \leq \mathcal{O}(\mathbf{u})\frac{\lambda_n}{\lambda_i} \approx \begin{cases} \mathcal{O}(\mathbf{u}) & \text{if } \lambda_i \approx \lambda_n \\ \mathcal{O}(\mathbf{u})\kappa_2(A) & \text{if } \lambda_i \approx \lambda_1. \end{cases} \tag{1}$$

It follows that larger eigenvalues (i.e. those $\lambda_i \approx \lambda_n$) can be computed with a relative error of order $\mathbf{u}$, but for smaller eigenvalue (i.e. those $\lambda_i \approx \lambda_1$), we may expect a relative error of order $\mathcal{O}(\mathbf{u})\kappa_2(A)$. Hence, little accuracy may be expected of these smaller eigenvalues if the matrix A is extremely ill-conditioned. This is the case regardless of the magnitude of λ_1.

Large matrices arising from applications are typically inherently ill-conditioned. Consider the eigenvalue problems for a differential operator $\mathcal{L}$. Discretization of $\mathcal{L}$ leads to a large and sparse matrix eigenvalue problem. Here it is usually a few smallest eigenvalues that are of interest and are well approximated by the discretization. Then, as the discretization mesh size h decreases, the condition number increases and then the relative accuracy of these smallest eigenvalues as computed by existing algorithms deteriorates. Specifically, the condition numbers of finite difference discretization are typically of order $\mathcal{O}(h^{-2})$ for second order differential operators, but for fourth order operators, it is of order $\mathcal{O}(h^{-4})$. This also holds true for other discretization methods; see [6, 11, 34, 40, 53] and the references contained therein for some recent discussions on discretization of fourth order operators. Thus, for a fourth order operator, little accuracy may be expected of the computed eigenvalues when h is near 10^{-4} in the standard double precision (see numerical examples in §4).

Indeed, computing eigenvalues and eigenvectors of a biharmonic operator Δ^2 has been a subject of much discussion; see [5, 7, 11, 12, 13, 14, 15, 39, 48]. It has been noted by Bjorstad and Tjostheim [8] that several earlier numerical results obtained in [5, 15, 39] based on coarse discretization schemes are inaccurate in the sense that they either miss or misplace some known nodal lines for the first eigenfunction. Indeed, to obtain more accurate numerical results that agree with various known theoretical properties of eigenvalues and eigenfunctions, they had to use the quadruple precision to compute the eigenvalues of the matrix obtained from the spectral Legendre-Galerkin method with up to 5000 basis functions in [8]. Clearly, the existing matrix eigenvalue algorithms implemented in the standard double precision could not provide satisfactory accuracy at this resolution.

The computed accuracy of smaller eigenvalues of a matrix has been discussed extensively in the context of the dense matrix eigenvalue problems in the last two decades. Starting with a work by Demmel and Kahan [23] on computing singular values of bidiagonal matrices, there is a large body of literature on when and how smaller eigenvalues or singular values can be computed accurately. Many special classes of matrices have been identified for which the singular values (or eigenvalues) are determined and can be computed to high relative accuracy (i.e. removing $\kappa_2(A)$ in (1)); see [4, 18, 19, 21, 22, 24, 25, 26, 28, 33, 43] and the recent survey [30] for more details.

In this paper, we study a new approach of using an iterative algorithm combined with the inverse (or the shift-and-invert transformation more generally) to accurately compute a few smallest eigenvalues of a diagonally dominant matrix or one that admits factorization into a product of diagonally dominant matrices. A key observation is that those smallest eigenvalues can be accurately computed from the corresponding largest eigenvalues of the inverse matrix, provided the inverse operator can be computed *accurately*. In light of extreme ill-conditioning that we assume for the matrix, accurate inversion is generally not possible but, for diagonally dominant matrices, we can use the accurate LDU factorization that we recently developed, with which the inverse (or linear systems) can be solved sufficiently accurately. As applications, we will show that we can compute a few smallest eigenvalues accurately for most second order self-adjoint differential operators and some fourth order differential operators in spite of extreme ill-conditioning of the discretization matrix. Numerical examples will be presented to demonstrate the accuracy achieved.

We note that using the Lanczos algorithm with the shift-and-invert transformation is a standard way for solving differential operator eigenvalue problems, where the lower end of the spectrum is sought but is clustered. The novelty here is the use of the accurate LDU factorization, that has about the same computational complexity as the Cholesky factorization but yields sufficiently accurate applications of the inverse operator.

The paper is organized as follows. We discuss in §2 an accurate LDU factorization algorithms for diagonally dominant matrices and then in §3, we show how they can be used to compute a few smallest eigenvalues accurately. We then apply this to discretization of differential operators together with various numerical examples in §4. We conclude with some remarks in §5.

Throughout, $\|\cdot\|$ denotes the 2-norm for vectors and matrices, unless otherwise specified. Inequalities and absolute value involving matrices and vectors are entrywise. $\mathbf{u}$ is the machine roundoff unit and $\mathcal{O}(\mathbf{u})$ denotes a term bounded by $p(n)\mathbf{u}$ for some polynomial $p(n)$ in n. We use $fl(z)$ to denote the computed result of an algebraic expression z. $\mathcal{R}(M)$ denote the range space of a matrix M and $\otimes$ is the Kronecker product.

2 Accurate *LDU* Factorization of Diagonally Dominant Matrices

Diagonally dominant matrices arise in many applications. With such a structure, it has been shown recently that several linear algebra problems can be solved much more accurately; see [1, 2, 18, 19, 51, 52]. In this section, we discuss related results on the LDU factorization of diagonally dominant matrices, which will yield accurate solutions of linear systems and will be the key in our method to compute a few smallest eigenvalues of extremely ill-conditioned matrices.

A key idea that makes more accurate algorithms possible is a representation (or re-parameterization) of diagonally dominant matrices as follows.

Definition 1 *Given $M = (m_{ij}) \in \mathbb{R}^{n\times n}$ with zero diagonals and $v = (v_i) \in \mathbb{R}^n$, we use $\mathcal{D}(M, v)$ to denote the matrix $A = (a_{ij})$ whose off-diagonal entries are the same as M and whose ith diagonal entry is $v_i + \sum_{j\neq i} |m_{ij}|$, i.e.*

$$a_{ij} = m_{ij} \textit{ for } i \neq j; \textit{ and } a_{ii} = v_i + \sum_{j\neq i} |m_{ij}|.$$

In this notation, for any matrix $A = [a_{ij}]$, let A_D be the matrix whose off-diagonal entries are the same as A and whose diagonal entries are zero and let $v_i = a_{ii} - \sum_{j\neq i} |a_{ij}|$ and $v = (v_1, v_2, \cdots, v_n)^T$, then we can write $A = \mathcal{D}(A_D, v)$, which will be called the representation of A by the diagonally dominant parts v. Through this equation, we use (A_D, v) as the data (parameters) to define the matrix (operator) A. The difference between using all the entries of A and using (A_D, v) lies in the fact that, under small entrywise perturbations, (A_D, v) determines all entries of A to the same relative accuracy, but not vice versa. Namely, (A_D, v) contains more information than the entries of A do.

In general, a matrix $A = (a_{ij})$ is said to be diagonally dominant if $|a_{ii}| \geq \sum_{j\neq i} |a_{ij}|$ for all i. Throughout this work, we consider a diagonally dominant A with nonnegative diagonals i.e. $v_i = a_{ii} - \sum_{j\neq i} |a_{ij}| \geq 0$ for all i. Diagonally dominant matrices with some negative diagonals can be scaled by a negative sign in the corresponding rows to turn into one with nonnegative diagonals. The need for scaling in such cases clearly does not pose any difficulties for the problem of solving linear systems.

In [51], we have developed a variation of the Gaussian elimination to compute the LDU factorization of an $n \times n$ diagonally dominant matrix represented as $A = \mathcal{D}(A_D, v)$ with $v \geq 0$ such that the diagonal matrix D has entrywise relative accuracy in the order of machine precision while L and U are well-conditioned with normwise accuracy (see Theorem 1 below). The algorithm is based on the observation that the Gaussian elimination can be carried out on A_D and v, and the entries of v can be computed with no subtraction operation, generalizing our earlier algorithm for diagonally dominant M-matrices [1, 2] and the GTH algorithm [35]. For completeness, we present the algorithm and its roundoff error properties below.

Algorithm 1 *([51]) LDU* FACTORIZATION OF $\mathcal{D}(A_D, v)$

```
Input: $A_D = [a_{ij}]$ and $v = [v_i] \geq 0$;
Initialize: $P = I$, $L = I$, $D = 0$, $U = I$.
For $k = 1 : (n-1)$
        For $i = k : n$
            $a_{ii} = v_i + \sum_{j=k, j \neq i}^{n} |a_{ij}|$;
        End For
        If $\max_{i \geq k} a_{ii} = 0$, stop;
        Choose a permutation $P_1$ for pivoting s.t. $A = P_1 A P_1$ satisfies one of:
            a) if diagonal pivoting: $a_{kk} = \max_{i \geq k} a_{ii}$;
            b) if column diagonal dominance pivoting: $0 \neq a_{kk} \geq \sum_{i=k+1}^{n} |a_{ik}|$;
        $P = P_1 P$; $L = P_1 L P_1$; $U = P_1 U P_1$; $d_k = a_{kk}$;
        For $i = (k+1) : n$
            $l_{ik} = a_{ik}/a_{kk}$; $u_{ki} = a_{ki}/a_{kk}$; $a_{ik} = 0$;
            $v_i = v_i + |l_{ik}| v_k$;
            For $j = (k+1) : n$
                    $p = \mathrm{sign}(a_{ij} - l_{ik} a_{kj})$;
                    $s = \mathrm{sign}(a_{ij}) p$;
                    $t = -\mathrm{sign}(l_{ik}) \mathrm{sign}(a_{kj}) p$;
                    If $j = i$
                            $s = 1$; $t = \mathrm{sign}(l_{ik}) \mathrm{sign}(a_{ki})$;
                    End if
                    $v_i = v_i + (1-s)|a_{ij}| + (1-t)|l_{ik} a_{kj}|$;
                    $a_{ij} = a_{ij} - l_{ik} a_{kj}$;
            End for
        End For
End for
$a_{nn} = v_n$; $d_n = a_{nn}$.
```

In output, we have $PAP^T = LDU$. We have considered two possible pivoting strategies in line 8. The column diagonal dominance pivoting ensures that L is column diagonally dominant while U is still row diagonally dominant. These theoretically guarantee that L and U are well-conditioned with

$$\kappa_\infty(L) := \|L\|_\infty \|L^{-1}\|_\infty \leq n^2 \quad \text{and} \quad \kappa_\infty(U) \leq 2n; \tag{2}$$

see [47]. In practice, however, the diagonal pivoting (i.e. $a_{kk} = \max_{i \geq k} a_{ii}$ at line 8a) is usually sufficient to result in well-conditioned L and U, but for the theoretical purpose, we assume that the column diagonal dominance pivoting will be used so that (2) holds.

The following theorem characterizes the accuracy achieved by Algorithm 1.

Theorem 1 *Let $\widehat{L} = [\widehat{l}_{ik}]$, $\widehat{D} = diag\{\widehat{d}_i\}$ and $\widehat{U} = [\widehat{u}_{ik}]$ be the computed factors of LDU-factorization of $\mathcal{D}(A_D, v)$ by Algorithm 1 and let $L = [l_{ik}]$, $D = diag\{d_i\}$ and $U = [u_{ik}]$ be the corresponding factors computed exactly. We have*

$$\begin{aligned} \|\widehat{L} - L\|_\infty &\leq \left(n\nu_{n-1}\mathbf{u} + \mathcal{O}(\mathbf{u}^2)\right) \|L\|_\infty, \\ |\widehat{d}_i - d_i| &\leq \left(\xi_{n-1}\mathbf{u} + \mathcal{O}(\mathbf{u}^2)\right) d_i, \quad \text{for } 1 \leq i \leq n, \\ \|\widehat{U} - U\|_\infty &\leq \left(\nu_{n-1}\mathbf{u} + \mathcal{O}(\mathbf{u}^2)\right) \|U\|_\infty. \end{aligned}$$

where $\nu_{n-1} \leq 14n^3$ and $\xi_{n-1} \leq 6n^3$.

The above theorem was originally proved in [51, Theorem 3] with $\nu_{n-1} \leq 6 \cdot 8^{n-1} - 2$ and $\xi_{n-1} \leq 5 \cdot 8^{n-1} - \frac{5}{2}$ but improved to the polynomial bound above in [27, Theorem 4]. The bounds demonstrate that the computed L and U are normwise accurate and D is entrywise accurate, regardless of the condition number of the matrix. Since the permutation P for pivoting does not involve any actual computations and roundoff errors, for the ease of presentation, we will assume from now on that $P = I$; that is the permutation has been applied to A.

The above accurate factorization was used in [51] to accurately compute all singular values of A. Based on the Jacobi algorithm, the algorithm there is suitable for small matrices only. Here, we will show that the accurate LDU factorization can also be used to solve a linear system more accurately. Specifically, with the factorization $A = \widehat{L}\widehat{D}\widehat{U}$, we solve $Ax = b$ by the standard procedure:

$$\widehat{L}y = b; \quad \widehat{D}z = y; \quad \text{and} \quad \widehat{U}x = z, \tag{3}$$

where the systems involving $\widehat{L}$ and $\widehat{U}$ are solved with forward and backward substitutions respectively, and the diagonal system is solved by $z_i = y_i/d_{ii}$.

The accuracy of the computed solution $\widehat{x}$ from (3) has been investigated by Dopico and Molera [29] in the context of a rank revealing factorization. A factorization $A = XDY$ is said to be rank revealing if X and Y are well conditioned and D is diagonal. Let $\widehat{X}$, $\widehat{D}$, and $\widehat{Y}$ be the computed factors of a rank revealing factorization $A = XDY$. We say it is an accurate rank revealing factorization of A (see [22]) if $\widehat{X}$ and $\widehat{Y}$ are normwise accurate and $\widehat{D}$ is entrywise accurate, i.e.,

$$\frac{\|\widehat{X} - X\|}{\|X\|} \leq \mathbf{u}p(n); \quad \frac{\|\widehat{Y} - Y\|}{\|Y\|} \leq \mathbf{u}p(n); \qquad \text{and } |\widehat{D} - D| \leq \mathbf{u}p(n)|D|, \tag{4}$$

where $p(n)$ is a polynomial in n. By (2), the LDU factorization defined by Algorithm 1 is a rank revealing factorization. Furthermore, it follows from Theorem 1 that the computed factors by Algorithm 1 form an accurate rank revealing factorization. The following theorem describes the accuracy of the computed solution of $Ax = b$. In the theorem below, the norm $\|\cdot\|$ can be any matrix operator norm satisfying $\|\text{diag}\{d_i\}\| = \max_i |d_i|$.

Theorem 2 *([29, Theorem 4.2]) Let $\widehat{X}$, $\widehat{D}$, and $\widehat{Y}$ be the computed factors of a rank revealing factorization of $A = XDY$ and assume that they satisfy (4) where $p(n)$ is a polynomial of n and X, D, and Y are the corresponding exact factors. Assume also that the systems $Xs = b$ and $Yx = w$ are solved with a backward stable algorithm that when applied to any linear system $Bz = c$, computes a solution $\hat{z}$ that satisfies $(B + \Delta B)\hat{z} = c$; with $\|\Delta B\| \leq \mathbf{u}q(n)\|B\|$ where $q(n)$ is a modestly growing function of n such that $q(n) \geq 4\sqrt{2}/(1 - 12\mathbf{u})$. Let $g(n) := p(n) + q(n) + \mathbf{u}p(n)q(n)$. Then, if $\hat{x}$ is the computed solution of $Ax = b$ through solving*

$$\widehat{X}y = b; \quad \widehat{D}z = y; \quad \text{and} \quad \widehat{Z}x = z,$$

and if $\mathbf{u}g(n)\kappa(Y) < 1$ and $\mathbf{u}g(n)(2 + \mathbf{u}g(n))\kappa(X) < 1$, then

$$\begin{aligned} \frac{\|\widehat{x} - x\|}{\|x\|} &\leq \frac{\mathbf{u}g(n)}{1 - \mathbf{u}g(n)\kappa(Y)} \left(\kappa(Y) + \frac{1 + (2 + \mathbf{u}g(n))\kappa(X)}{1 - \mathbf{u}g(n)(2 + \mathbf{u}g(n))\kappa(X)} \frac{\|A^{-1}\|\|b\|}{\|x\|} \right) \\ &= \left(\mathbf{u}g(n) + \mathcal{O}(\mathbf{u}^2)\right) \max\{\kappa(X), \kappa(Y)\} \frac{\|A^{-1}\|\|b\|}{\|x\|} \end{aligned}$$

Applying the above theorem to $A = LDU$ with the infinity norm and using the bounds on $\widehat{L}$, $\widehat{D}$ and $\widehat{U}$ in Theorem 1, we have that the computed solution $\widehat{x}$ by (3) satisfies the bound above with $p(n) = 14n^4$, $q(n) = n^2$. Hence, further using (2), we obtain

$$\frac{\|\widehat{x} - x\|_\infty}{\|x\|_\infty} \leq \left(\mathbf{u}n^2 g(n) + \mathcal{O}(\mathbf{u}^2)\right) \frac{\|A^{-1}\|_\infty \|b\|_\infty}{\|x\|_\infty}. \tag{5}$$

where $g(n) = 14n^6 + n^4$. We note that this worst-case bound with a large coefficient $n^2 g(n)$ is derived for a dense matrix from combining the bounds for the LDU factorization, for solving linear systems, and for the condition numbers, each of which is pessimistic. For a large sparse matrix that has $\mathcal{O}(n)$ nonzero entries, the coefficient can be significantly reduced if we assume the L and U factors also have $\mathcal{O}(n)$

nonzero entries. In any case, the bound can be expected to be too pessimistic to be useful for deriving a numerical bound in a practical setting. The main interest of the bound is to demonstrate the independence of the error on any conditioning of the problem.

For the convenience of later uses, we rewrite (5) in the 2-norm as

$$\|\widehat{x} - x\| \le \mathcal{O}(\mathbf{u})\|A^{-1}\|\|b\|. \tag{6}$$

3 Smaller Eigenvalues of Extremely Ill-conditioned Matrices

In this section, we discuss computations of a few smallest eigenvalues of an extremely ill-conditioned matrix A, namely we assume that $\kappa(A)\mathbf{u}$ is of order 1. For simplicity, we consider a symmetric positive definite matrix A. Many of the discussions are applicable to nonsymmetric problems, although there may be additional difficulties associated with sensitivity of individual eigenvalues caused by non-normality of the matrix.

Let $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$ be the eigenvalues of A and suppose we are interested in computing the smallest eigenvalue λ_1. For an ill-conditioned matrix, λ_1 is typically clustered; namely the relative spectral gap

$$\frac{\lambda_2 - \lambda_1}{\lambda_n - \lambda_2} = \frac{\lambda_2 - \lambda_1}{\lambda_1}\frac{1}{\lambda_n/\lambda_1 - \lambda_2/\lambda_1} \approx \frac{\lambda_2 - \lambda_1}{\lambda_1}\frac{1}{\kappa(A)} \tag{7}$$

is small unless $\lambda_2 \approx \lambda_n$. Therefore, although one can compute λ_1 using the Lanczos algorithm, its speed of convergence is determined by the relative spectral gap, and then a direct application to A will result in slow or no convergence. An efficient way to deal with the clustering is to apply the Lanczos method to the inverse A^{-1}, which has a much better relative spectral gap $\frac{\lambda_1^{-1}-\lambda_2^{-1}}{\lambda_2^{-1}-\lambda_n^{-1}} > \frac{\lambda_2-\lambda_1}{\lambda_1}$.

Consider using the Lanczos method on A^{-1} or simply inverse iteration and compute its largest eigenvalue $\mu_1 = \lambda_1^{-1}$. One may argue that, by (1), since μ_1 is the largest eigenvalue, it can be computed accurately, from which $\lambda_1 = \mu_1^{-1}$ can be recovered accurately. However, we can compute μ_1 accurately only if A^{-1} is explicitly available or the matrix-vector multiplication $A^{-1}v$ can be accurately computed, which is not the case if A is ill-conditioned. Specifically, if $u = A^{-1}v$ is computed by factorizing A and solving $Au = v$, then the computed solution $\widehat{u}$ is backward stable. That is $(A + \Delta A)\widehat{u} = v$ for some ΔA satisfying $\|\Delta A\| \le \mathcal{O}(\mathbf{u})\|A\|$. Then

$$\widehat{u} - u = -A^{-1}\Delta A\widehat{u} \ \text{ and } \ \|\widehat{u} - u\| \le \mathcal{O}(\mathbf{u})\|A^{-1}\|\|A\|\|\widehat{u}\|. \tag{8}$$

So, if $\mathbf{u}\|A^{-1}\|\|A\|$ is of order 1, very little accuracy can be expected of the computed result $\widehat{u}$. Then little accuracy can be expected of the computed eigenvalue as its accuracy is clearly limited by that of the matrix operator $A^{-1}v$. We further illustrate this point by looking at how an approximate eigenvalue is computed.

Note that all iterative methods is based on constructing an approximate eigenvector x_1, from which an approximate eigenvalue is computed, essentially as the Rayleigh quotient[1] $\rho_1 := \frac{x_1^T A^{-1} x_1}{x_1^T x_1}$. Unfortunately, for extremely ill-conditioned A, this Rayleigh quotient can not be computed with much accuracy. Specifically, to compute ρ_1, we first need to compute $A^{-1}x_1$. If $\widehat{u}_1$ is the computed solution to $Au_1 = x_1$, the computed Rayleigh quotient satisfies

$$\hat{\rho}_1 = fl\left(\frac{x_1^T \widehat{u}_1}{x_1^T x_1}\right) = \frac{x_1^T \widehat{u}_1}{x_1^T x_1} + e \ \text{ with } \ |e| \le \mathcal{O}(\mathbf{u}) \frac{|x_1|^T |\widehat{u}_1|}{x_1^T x_1} \le \mathcal{O}(\mathbf{u}) \frac{\|\widehat{u}_1\|}{\|x_1\|}. \tag{9}$$

Now, it follows from (8) that

$$|\hat{\rho}_1 - \rho_1| = \left| \frac{x_1^T(\widehat{u}_1 - u_1)}{x_1^T x_1} + e \right| \le \mathcal{O}(\mathbf{u})\kappa_2(A) \frac{\|\widehat{u}_1\|}{\|x_1\|}.$$

Using (8) again and $\|u_1\| = \|A^{-1}x_1\| \le \mu_1 \|x_1\|$, we have

$$\frac{\|\widehat{u}_1\|}{\|x_1\|} \le \frac{\|u_1\|}{\|x_1\|} + \frac{\|\widehat{u}_1 - u_1\|}{\|x_1\|} \le \mu_1 + \mathcal{O}(\mathbf{u})\kappa_2(A) \frac{\|\widehat{u}_1\|}{\|x_1\|}$$

from which it follows that

$$\frac{\|\widehat{u}_1\|}{\|x_1\|} \le \frac{\mu_1}{1 - \mathcal{O}(\mathbf{u})\kappa_2(A)}.$$

Therefore,

$$|\hat{\rho}_1 - \rho_1| \le \frac{\mathcal{O}(\mathbf{u})\kappa_2(A)}{1 - \mathcal{O}(\mathbf{u})\kappa_2(A)} \mu_1 = \mathcal{O}(\mathbf{u})\kappa_2(A)\mu_1.$$

With $\rho_1 \le \mu_1$, the relative error of the computed Rayleigh quotient $\hat{\rho}_1$ is expected to be of order $\mathcal{O}(\mathbf{u})\kappa_2(A)$. Note that this is independent of the algorithm that we use to obtain an approximate eigenvector x_1. Indeed, this is the case even when x_1 is an exact eigenvector. That is, even if we have the exact eigenvector, we are still not able to compute a corresponding eigenvalue with any accuracy if it is computed numerically through the Rayleigh quotient.

We point out that the same difficulty occurs if we compute approximate eigenvalue λ_1 directly using the Rayleigh quotient of A, i.e. using $\rho_1 = \frac{x_1^T A x_1}{x_1^T x_1}$. In this case, as in (9), the computed Rayleigh quotient $\hat{\rho}_1$ satisfies

$$\hat{\rho}_1 = fl\left(\frac{x_1^T \widehat{y}_1}{x_1^T x_1}\right) = \frac{x_1^T \widehat{y}_1}{x_1^T x_1} + e \ \text{ with } \ |e| \le \mathcal{O}(\mathbf{u}) \frac{|x_1|^T |\widehat{y}_1|}{x_1^T x_1} \le \mathcal{O}(\mathbf{u}) \frac{\|\widehat{y}_1\|}{\|x_1\|}$$

[1] In the Lanczos algorithm, the Ritz value is computed as the eigenvalue of the projection matrix, which is still the Rayleigh quotient of the Ritz vector.

where $\widehat{y}_1 = fl(Ax_1)$. Set $y_1 = Ax_1$. Then $\widehat{y}_1 = fl(Ax_1) = y_1 + f$ with $|f| \leq \frac{n\mathbf{u}}{1-n\mathbf{u}}|A||x_1|$. Hence, $\|\widehat{y}_1\| \leq (1+\mathcal{O}(\mathbf{u}))\|A\|\|x_1\|$ and $\|\widehat{y}_1 - y_1\| \leq \mathcal{O}(\mathbf{u})\|A\|\|x_1\|$. It follows that $|e| \leq \mathcal{O}(\mathbf{u})\|A\|$ and, for $\rho_1 \approx \lambda_1$,

$$\frac{|\hat{\rho}_1 - \rho_1|}{\rho_1} = \frac{1}{\rho_1}\left|\frac{x_1^T(\widehat{y}_1 - y_1)}{x_1^T x_1} + e\right| \leq \frac{\mathcal{O}(\mathbf{u})\|A\|}{\rho_1} \approx \mathcal{O}(\mathbf{u})\|A^{-1}\|\|A\|. \tag{10}$$

Again, the accuracy of $\hat{\rho}_1$ is limited by the condition number. This also shows that, if we apply the Lanczos method directly to A, then the smallest Ritz eigenvalue can not be accurately computed, even if the process converges.

The above discussions demonstrate why all existing methods will have difficulties computing the smallest eigenvalue accurately. It also highlights that the roundoff errors encountered in computing $A^{-1}v$ are the root cause of the problem. This also readily suggests a remedy: compute $A^{-1}v$ more accurately. It turns out that a more accurate solution to $Au = v$, in the sense of (6), is sufficient, and this can be done for a diagonally dominant matrix or a product of diagonally dominant matrices.

Let $A = \mathcal{D}(A_D, v)$ be a (possibly nonsymmetric) diagonally dominant matrix with $v \geq 0$. We apply the Lanczos method (or the Arnoldi method in the nonsymmetric case) to A^{-1} or simply using inverse iteration. Then at each iteration, we need to compute $u = A^{-1}v$ for some v. This is done by computing the accurate LDU factorization of A by Algorithm 1 and then solve $Au = v$ by (3). The computed solution $\widehat{u}$ then satisfies (5), i.e.

$$\|\widehat{u} - u\| \leq \mathcal{O}(\mathbf{u})\|A^{-1}\|\|v\|. \tag{11}$$

This error depends on $\|A^{-1}\|$ but is as accurate as multiplying the exact A^{-1} with v. Specifically, if we have A^{-1} exactly, say $A^{-1} = B$, and we compute $A^{-1}v$ by computing Bv in a floating point arithmetic, the error is bounded as

$$\|fl(Bv) - Bv\| \leq \frac{n^2\mathbf{u}}{1-n\mathbf{u}}\|B\|\|v\| = \frac{n^2\mathbf{u}}{1-n\mathbf{u}}\|A^{-1}\|\|v\|.$$

So, the error of $\widehat{u}$ is of the same order as that of $fl(Bv)$.

Therefore, applying the Lanczos algorithm to A^{-1} by solving $Au = v$ using Algorithm 1 and (3) has essentially the same roundoff effects as applying it to B in a floating point arithmetic. In that case, $\mu_1 := \lambda_1^{-1}$, being the largest eigenvalue of $B = A^{-1}$, is computed accurately (see (1)). In light of the earlier discussions with respect to the Rayleigh quotient, we also show that the Rayleigh quotient $\rho_1 := \frac{x_1^T A^{-1} x_1}{x_1^T x_1}$ for an approximate eigenvector x_1 can be computed accurately. Specifically, if $\widehat{u}_1$ is the computed solution to $Au_1 = x_1$, then $\|\widehat{u}_1 - u_1\| \leq \mathcal{O}(\mathbf{u})\|A^{-1}\|\|x_1\|$ by (11) and the computed Rayleigh quotient $\hat{\rho}_1 := fl\left(\frac{x_1^T\widehat{u}_1}{x_1^T x_1}\right)$ satisfies (9). Thus, the error is bounded as

$$|\hat{\rho}_1 - \rho_1| = \left|\frac{x_1^T(\widehat{u}_1 - u_1)}{x_1^T x_1} + e\right| \leq \mathcal{O}(\mathbf{u})\|A^{-1}\| = \mathcal{O}(\mathbf{u})\mu_1,$$

where we have used

$$|e| \leq \mathcal{O}(\mathbf{u})\frac{\|\widehat{u}_1\|}{\|x_1\|} \leq \mathcal{O}(\mathbf{u})\frac{\|\widehat{u}_1 - u_1\|}{\|x_1\|} + \mathcal{O}(\mathbf{u})\frac{\|u_1\|}{\|x_1\|} \leq \mathcal{O}(\mathbf{u})\|A^{-1}\|.$$

With $\rho_1 \approx \mu_1$, thus, ρ_1 is computed with a relative error of order $\mathbf{u}$.

Now, suppose a matrix A is not diagonally dominant but can be written as a product of two or more diagonally dominant matrices, say $A = A_1A_2$. Then we can factorize A_1 and A_2 using Algorithm 1 and then solve $Au = v$ through $A_1w = v$ and then $A_2u = w$ as in (3). Let $\widehat{w}$ and $\widehat{u}$ be the computed solutions to $A_1w = v$ and $A_2u = \widehat{w}$ respectively. Using (6), they satisfy

$$\|\widehat{w} - w\| \leq \mathcal{O}(\mathbf{u})\|A_1^{-1}\|\|v\|$$

and

$$\begin{aligned} \|\widehat{u} - u\| &\leq \mathcal{O}(\mathbf{u})\|A_2^{-1}\|\|\widehat{w}\| \\ &\leq \mathcal{O}(\mathbf{u})\|A_2^{-1}\|\|w\| + \mathcal{O}(\mathbf{u})\|A_2^{-1}\|\|\widehat{w} - w\| \\ &\leq \mathcal{O}(\mathbf{u})\|A_2^{-1}\|\|A_1^{-1}\|\|v\|. \end{aligned}$$

Thus, writing $\gamma = \|A_1^{-1}\|\|A_2^{-1}\|/\|A^{-1}\|$, we have

$$\|\widehat{u} - u\| \leq \mathcal{O}(\mathbf{u})\gamma\|A^{-1}\|\|v\|. \tag{12}$$

So, as discussed above, the largest eigenvalue of A^{-1} can be computed to an accuracy of order $\mathcal{O}(\mathbf{u})\gamma$, which should be satisfactory as long as γ is not an extremely large constant.

In the next section, we present examples from differential operators to demonstrate that the approaches discussed in this section result in accurate computed eigenvalues.

4 Eigenvalues of Differential Operators

The eigenvalue problem for self-adjoint differential operator arises in dynamical analysis of vibrating elastic structures such as bars, beams, strings, membranes, plates, and solids. It takes the form of $\mathcal{L}u = \lambda u$ in Ω with a suitable homogeneous boundary condition, where $\mathcal{L}$ is a differential operator and Ω is a bounded domain in one, two, or three dimensions. For example, the natural vibrating frequencies of a thin membrane stretched over a bounded domain Ω and fixed along the boundary $\partial\Omega$ is determined by the eigenvalue problem for a second order differential operator [3, p.653]

$$-\nabla(\mathbf{c}(x,y)\nabla u(x,y)) = \lambda u(x,y) \quad \text{in } (x,y) \in \Omega, \tag{13}$$

with a homogeneous boundary condition $\alpha u + \beta \frac{\partial u}{\partial n} = 0$ for $(x, y) \in \partial\Omega$. On the other hand, a vibrating plate on Ω is described by one for a fourth order differential operator [50, p.16]

$$\Delta^2 u(x, y) = \lambda u(x, y) \quad \text{in } (x, y) \in \Omega, \tag{14}$$

with the natural boundary condition $u = \Delta u = 0$ (if the plate is simply supported at the boundary) or the Dirichlet boundary condition $u = \frac{\partial u}{\partial n} = 0$ (if the plate is clamped at the boundary) for $(x, y) \in \partial\Omega$. The spectrum of such a differential operator consists of an infinite sequence of eigenvalues with an accumulation point at the infinity:

$$\lambda_1 \le \lambda_2 \le \cdots\cdots \uparrow +\infty. \tag{15}$$

When computing eigenvalues for practical problems, it is usually lower frequencies, i.e. the eigenvalues at the left end of the spectrum, such as λ_1, that are of interest.

Numerical solutions of the differential eigenvalue problems typically involve first discretizing the differential equations and then solving a matrix eigenvalue problem. Then, it is usually a few smallest eigenvalues of the matrix that well approximate the eigenvalues of the differential operators and are of interest. The finite difference discretization leads to a standard eigenvalue problem $Ax = \lambda x$, while the finite element methods or the spectral methods result in a generalized eigenvalue problem $Ax = \lambda Bx$. Although our discussions may be relevant to all three discretization methods, we shall consider the finite difference method here as they often give rise to the diagonal dominant structure that can be utilized in this work.

Let A_h be an $n \times n$ symmetric finite difference discretization of a self-adjoint positive definite differential operator $\mathcal{L}$ with h being the meshsize and let

$$0 < \lambda_{1,h} \le \lambda_{2,h} \le \cdots \le \lambda_{n,h} \tag{16}$$

be its eigenvalues. Under some conditions on the operator and the domain [41], we have for a fixed i, $\lambda_{i,h} \to \lambda_i$ as $h \to 0$, while the largest eigenvalue $\lambda_{n,h}$ approaches ∞. Then, the condition number $\kappa_2(A_h)$ of A_h approaches the infinity as the meshsize decreases to 0. As discussed in the introduction (see (1)), if $\widehat{\lambda}_{1,h}$ is the computed smallest eigenvalue, its relative error is expected to be of order $\kappa_2(A_h)\mathbf{u}$. Hence, we have a situation that, as the meshsize decreases, the discretization error $\lambda_{1,h} - \lambda_1$ decreases, but our ability to accurately compute $\lambda_{1,h}$ also decreases. Indeed, the overall error

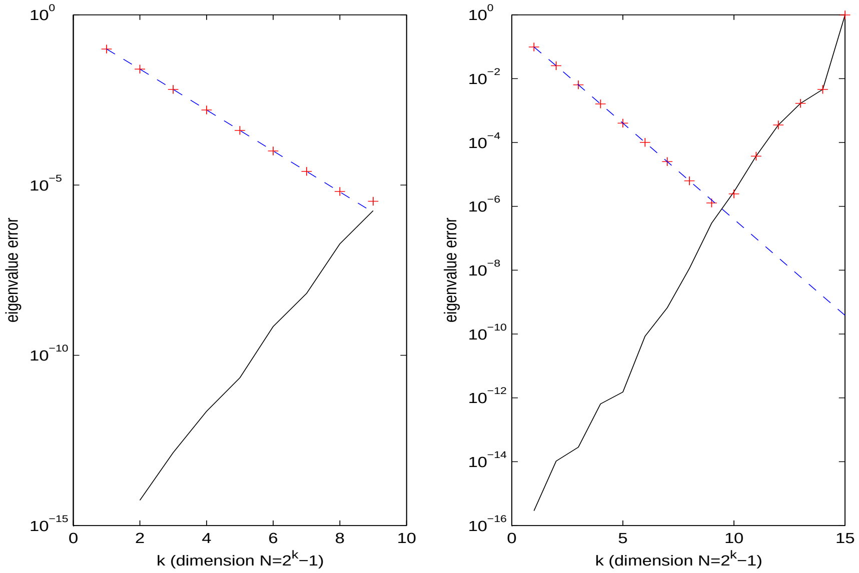
$$|\widehat{\lambda}_{1,h} - \lambda_1| \le |\widehat{\lambda}_{1,h} - \lambda_{1,h}| + |\lambda_{1,h} - \lambda_1| \le \lambda_{1,h}\kappa_2(A_h)\mathcal{O}(\mathbf{u}) + |\lambda_{1,h} - \lambda_1|.$$

will at some point be dominated by the computation error $|\widehat{\lambda}_{1,h} - \lambda_{1,h}|$. We illustrate with an example.

Example 1: Consider the biharmonic eigenvalue problem $\Delta^2 u = \lambda u$ on the unit square $[0, 1]^2$. With the simply supported boundary condition $u = \frac{\partial^2 u}{\partial n^2} = 0$, the eigenvalues are known explicitly [5] and the smallest one is $\lambda_1 = 4\pi^4$. The 13-point finite difference discretization with a meshsize $h = 1/(N+1)$ leads to a block five-diagonal matrix (see [5] or [44, p.131]). It is easy to check that the discretization matrix can be written as $A_h = \frac{1}{h^4} T^2_{N\times N}$, where $T_{N\times N} = T_N \otimes I + I \otimes T_N$, T_N is the $N \times N$ tridiagonal matrix with diagonals being 2 and off-diagonals being -1, and $\otimes$ is the Kronecker product. The eigenvalues of A_h are also known exactly and $\lambda_{1,h} = \frac{1}{h^4} 64 \sin^4(\pi h/2)$.

We compute the smallest eigenvalue of A_h using the implicitly restarted Lanczos algorithm with shift-and-invert (i.e. using `eigs(A,1,`σ`)` of MATLAB with $\sigma = 0$ with its default termination criterion) and let $\widehat{\lambda}_{1,h}$ be the computed result. We compare the discretization error $|\lambda_{1,h} - \lambda_1|$, the computation error $|\widehat{\lambda}_{1,h} - \lambda_{1,h}|$ and the overall approximation error $|\widehat{\lambda}_{1,h} - \lambda_1|$. We use increasingly fine mesh (with $N = 2^k - 1$ for $k = 1, 2, \cdots$) and plot the corresponding errors against k in Figure 1 (left). We test the value of k up to $k = 8$ for the 2-D problem. To test larger values of k, we use the corresponding 1-D biharmonic problem where $A_h = \frac{1}{h^4} T_N^2$, $\lambda_{1,h} = \frac{16}{h^4} \sin^4(\pi h/2)$ and $\lambda_1 = \pi^4$. Here we use k up to $k = 15$ and plot the result in Figure 1 (right).

Figure 1: Example 1. Left: 2-D problem; Right: 1-D problem. solid line: computation error; dashed line: discretization error; +-line: overall error.



It is clear that as the meshsize decreases, the discretization error (dashed line) decreases, while the computation error (solid line) increases. At some point ($k = 9$ in this case), the computation error dominates the discretization error and the total error (+-line) increases from that point on to $\mathcal{O}(1)$. So, at some point, using finer discretizations actually increases the final error.

We note that the large error in the computed eigenvalue is not due to possible numerical complications of the Lanczos method we used to compute the eigenvalues of A_h. Indeed, other than the implicitly restarted Lanczos method (`eigs`), we have also used simply inverse iteration and the Lanczos algorithm with full reorthogonalization as combined with the Cholesky factorization for inverting A to compute the smallest eigenvalue and obtained similar results.

We also note that for this particular problem, the eigenvalues can be computed as the square of the eigenvalues of $T_{N\times N}$ (or T_N in the 1-dimensional case). This is not the case for a general operator involving weight functions or a different boundary condition. Here, we compute the eigenvalues directly from A_h to illustrate the difficulty encountered by ill-conditioning.

4.1 Second Order Differential Operators.

A standard finite difference discretization of second order differential operators such as (13) is the five-point scheme, which typically has a condition number of order $\mathcal{O}(h^{-2})$. Then the relative error for the smallest eigenvalue computed is expected to be $\mathcal{O}(h^{-2})\mathbf{u}$. For modestly small h, this relative error may be smaller than the discretization error and would not cause any problem. However, there are situations where λ_1 may be 0 or very close to 0, resulting in a condition number significantly greater than h^{-2} and hence low accuracy of the computed eigenvalue even when h is modestly small. This is the case if we have a periodic boundary condition or Neumann boundary condition. Then the smallest eigenvalue may be computed with little accuracy (see Example 2 below). Note that for a zero eigenvalue, we need to consider the absolute error, which will be proportional to $\mathcal{O}(h^{-2})\mathbf{u}$.

For the second order differential operators, fortunately, the finite difference discretization matrices are typically diagonally dominant; see [49, p.211]. Then, a few smallest eigenvalues can be computed accurately by applying standard iterative methods to A^{-1} as discussed in §3. We demonstrate this in the following numerical example.

In this and Examples 3 and 4 later, we have used both the Lanczos algorithm with full reorthogonalization for A^{-1} and inverse iteration and found the results to be similar. With the smallest eigenvalue well separated, both methods converge quickly but the Lanczos algorithm (without restart) typically requires fewer iterations. On the other hand, inverse iteration sometimes improves the residuals by up to one order of magnitude. We report the results obtained by inverse iteration only.

Example 2: Consider the eigenvalue problem for $\mathcal{L}u = -\Delta u + \rho(x,y)u$ on the 2-d unit square $[0,1]^2$ with the periodic boundary condition $u(0,y) = u(1,y)$ and $u(x,0) = u(x,1)$. If $\rho(x,y) = 0$, the smallest eigenvalue of $\mathcal{L}$ is 0. If $\rho(x,y) \leq \epsilon$ for some small ϵ, then the smallest eigenvalue is at most ϵ. In particular, we test the case that $\rho(x,y) = \rho$ is a small constant in this example. Then the smallest eigenvalue of $\mathcal{L}$ is exactly ρ with $u = 1$ being a corresponding eigenfunction.

The 5-point finite difference discretization with a meshsize $h = 1/(N+1)$ leads to $A_h = \frac{1}{h^2}(\widehat{T}_{N+1} \otimes I + I \otimes \widehat{T}_{N+1}) + \rho I$ where $\widehat{T}_{N+1}$ is the $(N+1) \times (N+1)$ matrix that is the same as T_{N+1} in Example 1 (i.e tridiagonal with 2 on the diagonal and -1 on the subdiagonals) except at the $(1, N+1)$ and $(N+1, 1)$ entries which are -1. The smallest eigenvalue of A_h is still ρ while the largest is of order h^{-2}.

We compute $\lambda_{1,h}$ by applying inverse iteration with the inverse operator $A_h^{-1}x_k$ computed using the Cholsky factorization or using the accurate LDL^T factorization (Algorithm 1) of A_h. Denote the computed eigenvalues by μ_1^{chol} and μ_1^{aldu} respectively in Table 1, where the test is run with $h = 2^{-k}$ with $k = 3, 4, \cdots, 9$ and $\rho = 1e - 8$. The termination criterion is $\|A^{-1}x_k - \mu_k x_k\|/\|x_k\| \leq n\mathbf{u}|\mu_k|$.

For this problem, ρ is always the smallest eigenvalue of A_h. As h decreases, the condition number of A_h increases, which cause the relative error of the computed eigenvalue μ_1^{chol} to increase steadily. On the other hand, μ_1^{aldu} computed by the new approach using the accurate factorization has a relative error in the order of machine precision $\mathbf{u}$ independent of h.

Table 1: Example 2: approximation of $\lambda_1 = 1e - 8$ (μ_1^{chol} - computed eigenvalue by Cholesky; μ_1^{aldu} - computed eigenvalue by Algorithm 1.)

h	μ_1^{chol}	$\frac{\lvert\lambda_1 - \mu_1^{chol}\rvert}{\lambda_1}$	μ_1^{aldu}	$\frac{\lvert\lambda_1 - \mu_1^{aldu}\rvert}{\lambda_1}$
1.3e-1	1.00000132998311550e-8	1.3e-6	9.99999999999999860e-9	1.7e-16
6.3e-2	9.99988941000340750e-9	1.1e-5	1.00000000000000020e-8	1.7e-16
3.1e-2	9.99983035717776210e-9	1.7e-5	1.00000000000000020e-8	1.7e-16
1.6e-2	1.00008828596534290e-8	8.8e-5	1.00000000000000000e-8	0
7.8e-3	9.99709923055774120e-9	2.9e-4	1.00000000000000000e-8	0
3.9e-3	1.00122462366335050e-8	1.2e-3	1.00000000000000050e-8	5.0e-16
2.0e-3	1.00148903549009850e-8	1.5e-3	1.00000000000000040e-8	3.3e-16

4.2 Biharmonic operator with the natural boundary condition

A thin vibrating elastic plate on Ω that is simply supported at the boundary $\partial\Omega$ is described by the eigenvalue problem for the biharmonic operator (14) with the natural boundary condition

$$u = \Delta u = 0 \quad \text{for } (x, y) \in \partial\Omega. \tag{17}$$

Discretization of such operators are not diagonally dominant. However, the biharmonic operator Δ^2 has a form as a composition of two second order differential operators. Indeed, with the simply supported boundary condition (17), the eigenvalue equation has the product form

$$\begin{cases} -\Delta u = v, & \text{with } u = 0 \text{ for } (x,y) \in \partial\Omega \\ -\Delta v = \lambda u & \text{with } v = 0 \text{ for } (x,y) \in \partial\Omega \end{cases}$$

Then, the two Laplacian operators with the same boundary condition will have the same discretization matrix A_h. Thus the final discretization has the product form $B_h = A_h^2$. Since A_h is typically diagonally dominant, B_h is a product of two diagonally dominant matrices, and then as discussed in Section 3, its eigenvalue can also be computed accurately by using accurate factorizations of A_h. Note that, with respect to (12), we have $\gamma = \|A_h\|\|A_h\|/\|B_h\| = 1$.

For this problem, the eigenvalue can also be obtained by computing the eigenvalue of A_h first and then squaring. With A_h having more modest ill-conditioning, its eigenvalues can be computed more accurately. However, this approach does not generalize to problems such as stretched plates with a nonuniform stretch factor $\rho(x,y)$ as described by $\Delta^2 u(x,y) - \rho\Delta u(x,y) = \lambda u(x,y)$. On the other hand, its discretization can still be expressed in a product form; that is $B_h = (A_h + D)A_h$ where $D = \text{diag}\{\rho(x_i, y_j)\}$. Then if $\rho(x,y) \geq 0$, $A_h + D$ is also diagonally dominant. Thus the method described in §3 can compute a few smallest eigenvalues of B_h accurately. We illustrate this in the following numerical example. We use the 1-dimensional problem so as to test some really fine meshes.

Example 3: Consider computing the smallest eigenvalue of the eigenvalue problem: $\frac{d^4v}{dx^4} - \rho(x)\frac{d^2v}{dx^2} = \lambda v$ on $[0,1]$ with the natural boundary condition $v(0) = v''(0) = v(1) = v''(1) = 0$. This models vibration of beams; see [50, p.15]. Discretization on a uniform mesh with $h = 1/(N+1)$ leads to

$$B_h u_N = \lambda h^4 u_N; \quad \text{where} \quad B_h = T_N^2 + h^2 D T_N = (T_N + h^2 D)T_N \tag{18}$$

where $D = \text{diag}\{\rho(ih)\}$. We test the case with a constant stretch factor $\rho = 1$ so that $D = I$ and the eigenvalues of the differential operator are exactly known. Namely, $\lambda_1 = \pi^4 + \pi^2$.

We compute $\lambda_{1,h}$ by applying inverse iteration[2] to B_h and we use the termination criterion $\|B_h^{-1}x_k - \mu_k x_k\|/\|x_k\| \leq n\mathbf{u}|\mu_k|$. In applying B_h^{-1}, we compare the methods of using the Cholesky factorization of B_h and using the accurate LDL^T factorization of $T_N + h^2 D$ and T_N in (18). We denote the computed eigenvalues by μ_1^{chol} and μ_1^{aldu} respectively and present the results in Table 2.

From the table, the error of μ_1^{chol} decreases quadratically for h up to $h \approx 5e-4$, at which point further decrease of h actually increases the error. When $h \approx 1e-5$, there is no accuracy left in the computed result μ_1^{chol}. On the other hand, the error

[2] As mentioned before, similar results are obtained by applying the Lanczos algorithm with full reorthogonalization to B_h^{-1}.

Table 2: Example 3: approximation of $\lambda_1 = \pi^4 + \pi^2$ (μ_1^{chol} - computed eigenvalue by Cholesky; μ_1^{aldu} - computed eigenvalue by Algorithm 1.)

h	μ_1^{chol}	$\frac{\lvert\lambda_1 - \mu_1^{chol}\rvert}{\lambda_1}$	μ_1^{aldu}	$\frac{\lvert\lambda_1 - \mu_1^{aldu}\rvert}{\lambda_1}$
7.8e-3	1.07268420705487220e2	9.6e-5	1.07268420682006790e2	9.6e-5
3.9e-3	1.07276126605835940e2	2.4e-5	1.07276126662112130e2	2.4e-5
2.0e-3	1.07278050199309870e2	6.0e-6	1.07278053236552470e2	6.0e-6
9.8e-4	1.07278541687712870e2	1.4e-6	1.07278534885126120e2	1.5e-6
4.9e-4	1.07278608040777410e2	8.1e-7	1.07278655297579520e2	3.7e-7
2.4e-4	1.07276683387428220e2	1.9e-5	1.07278685400712600e2	9.4e-8
1.2e-4	1.07255355496649590e2	2.2e-4	1.07278692926496390e2	2.3e-8
6.1e-5	1.10218434482696150e2	2.7e-2	1.07278694807942740e2	5.8e-9
3.1e-5	1.16309014323327090e2	8.4e-2	1.07278695278302880e2	1.5e-9
1.5e-5	2.50710434033222100e2	1.3e0	1.07278695395894270e2	3.7e-10

for μ_1^{aldu} continues to decrease quadratically and maintains an accuracy to the order of machine precision.

4.3 Biharmonic operator with the Dirichlet boundary condition

A thin vibrating plate on Ω that is clamped at the boundary $\partial\Omega$ is described by the eigenvalue problem for the biharmonic operator (14) with the Dirichlet boundary condition

$$u = \frac{\partial u}{\partial n} = 0 \quad \text{on } (x, y) \in \partial\Omega. \tag{19}$$

With this boundary condition, the standard 13-point discretization does not have a product form; see [31]. For the 1-dimensional case, it turns out that we can derive a suitable discretization at the boundary so that the resulting matrix is a product of two diagonally dominant matrices. Here we present the details of this scheme.

We consider one dimensional fourth order bi-harmonic operator

$$\frac{d^4 v}{dx^4} = \lambda v \quad \text{on} \quad [0, 1] \tag{20}$$

with the Dirichlet boundary condition

$$v(0) = v'(0) = v(1) = v'(1) = 0 \tag{21}$$

We discretize the equation on the uniform mesh $0 = x_0 \leq x_1 \leq \cdots \leq x_N \leq x_{N+1} = 1$ with $x_i = ih$ and $h = 1/(N + 1)$. Let $\mathbf{v} := [v(x_1), v(x_2), \cdots, v(x_N)]^T$. A standard discretization corresponding to the 13-point scheme [31] uses the second order center difference for $\frac{d^2}{dx^2}$ and $v(-h) = v(h) + \mathcal{O}(h^3)$ for the boundary condition $v'(0) = 0$ with a similar one for $v'(1) = 0$. This results in $A_h\mathbf{v} = \lambda\mathbf{v} + e$, where λ is the eigenvalue of the operator (20) and

$$A_h = \frac{1}{h^4}\begin{pmatrix} 7 & -4 & 1 & & & & & \\ -4 & 6 & -4 & 1 & & & & \\ 1 & -4 & 6 & -4 & 1 & & & \\ & 1 & -4 & 6 & -4 & 1 & & \\ & & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & & 1 & -4 & 6 & -4 & 1 \\ & & & & 1 & -4 & 6 & -4 \\ & & & & & 1 & -4 & 7 \end{pmatrix}, \quad \mathbf{e} = \begin{pmatrix} \mathcal{O}(h^{-1}) \\ \mathcal{O}(h^2) \\ \vdots \\ \mathcal{O}(h^2) \\ \mathcal{O}(h^{-1}) \end{pmatrix}. \qquad (22)$$

We note that the local truncation error e is of order $\mathcal{O}(h^{-1})$ only at the two boundary points, but it is shown in [10, 42] that this is sufficient to get a second order convergence. Namely, there is some eigenvalue of A_h such that $|\lambda_h - \lambda| \leq Ch^2$ where C is a constant dependent on the eigenvector v.

The standard discretization A_h has a condition number of order h^{-4}. It is not diagonally dominant and it does not appear to have a factorization as a product of diagonally dominant matrices. Thus, direct computations of a few smallest eigenvalue of A_h will have low accuracy when h becomes small; see Example 4.

To be able to accurately compute a few smallest eigenvalues, we now derive a discretization that is a product of diagonally dominant matrices. This is done by exploring the product form $\frac{d^2}{dx^2}\frac{d^2}{dx^2}$ of the operator. Let $w = -\frac{d^2v}{dx^2}$ and $u = -\frac{d^2w}{dx^2} = \frac{d^4v}{dx^4}$. Let

$$\mathbf{v} = \begin{pmatrix} v(x_1) \\ v(x_2) \\ \vdots \\ v(x_N) \end{pmatrix}, \quad \mathbf{w} = \begin{pmatrix} w(x_0) \\ w(x_1) \\ \vdots \\ w(x_N) \\ w(x_{N+1}) \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} u(x_1) \\ u(x_2) \\ \vdots \\ u(x_N) \end{pmatrix} \text{ and } \hat{\mathbf{u}} = \begin{pmatrix} u(x_0) \\ u(x_1) \\ \vdots \\ u(x_N) \\ u(x_{N+1}) \end{pmatrix}.$$

Then, we have, for $1 \leq i \leq N$,

$$\begin{aligned} w(x_i) &= \frac{-v(x_{i-1}) + 2v(x_i) - v(x_{i+1})}{h^2} + \frac{1}{12}v^{(4)}(x_i)h^2 + \mathcal{O}(h^4) \\ &= \frac{-v(x_{i-1}) + 2v(x_i) - v(x_{i+1})}{h^2} + \frac{1}{12}u(x_i)h^2 + \mathcal{O}(h^4) \end{aligned}$$

and

$$\begin{aligned} u(x_i) &= \frac{-w(x_{i-1}) + 2w(x_i) - w(x_{i+1})}{h^2} + \frac{1}{12}w^{(4)}(x_i)h^2 + \mathcal{O}(h^4) \\ &= \frac{-w(x_{i-1}) + 2w(x_i) - w(x_{i+1})}{h^2} + \frac{1}{12}u''(x_i)h^2 + \mathcal{O}(h^4). \end{aligned}$$

Among many possible discretizations of the boundary condition $v'(x_0) = v'(x_{N+1}) = 0$, we use the following scheme to maintain the product form of the operator:

$$w(x_0) = \frac{2v(x_1) - v(x_2)}{h^2} + \mathcal{O}(h) \text{ and } w(x_{N+1}) = \frac{2v(x_N) - v(x_{N-1})}{h^2} + \mathcal{O}(h),$$

which is derived, in the case of the left end, by expanding $v(x_1)$, $v(x_2)$ about x_0 using $v(x_0) = v'(x_0) = 0$. Then,

$$\mathbf{w} = \frac{1}{h^2}\widehat{S}_N \mathbf{v} + \frac{1}{12}\hat{\mathbf{u}} h^2 + f$$

with $f = [\mathcal{O}(h), \mathcal{O}(h^4), \cdots, \mathcal{O}(h^4), \mathcal{O}(h)]^T$, and

$$\mathbf{u} = \frac{1}{h^2}\widehat{T}_N \mathbf{w} + \mathcal{O}(h^2) \tag{23}$$

where

$$\widehat{S}_N = \left(\begin{array}{c} 2 \quad -1 \qquad\qquad\qquad \\ \hline \\ T_N \\ \\ \hline \qquad\qquad\qquad -1 \quad 2 \end{array} \right), \quad T_N = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 2 \end{pmatrix}$$

are respectively $(N+2) \times N$ and $N \times N$ matrices and

$$\widehat{T}_N = \begin{pmatrix} -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \end{pmatrix} \in \mathbb{R}^{N \times (N+2)}$$

Note that by applying (23) to $\hat{\mathbf{u}}$, we have $\frac{1}{h^2}\widehat{T}_N \hat{\mathbf{u}} = \mathbf{v}^{(6)} + \mathcal{O}(h^2)$ where $\mathbf{v}^{(6)} = [v^{(6)}(x_i)]_{i=1}^N$. Hence,

$$\begin{aligned} \mathbf{u} &= \frac{1}{h^4}\widehat{T}_N \widehat{S}_N \mathbf{v} + \frac{1}{12h^2}\widehat{T}_N \hat{\mathbf{u}} h^2 + \frac{1}{12h^2}\widehat{T}_N f + \mathcal{O}(h^2) \\ &= \frac{1}{h^4}\widehat{T}_N \widehat{S}_N \mathbf{v} + f_1 \end{aligned}$$

with

$$f_1 := \frac{1}{12}\widehat{T}_N \hat{\mathbf{u}} + \frac{1}{12h^2}\widehat{T}_N f + \mathcal{O}(h^2) = [\mathcal{O}(h^{-1}), \mathcal{O}(h^2), \cdots, \mathcal{O}(h^2), \mathcal{O}(h^{-1})]^T, \tag{24}$$

where we note that $\widehat{T}_N\hat{\mathbf{u}} = \mathbf{v}^{(6)}h^2 + \mathcal{O}(h^4)$ and $\widehat{T}_N f = [\mathcal{O}(h), \mathcal{O}(h^4), \cdots, \mathcal{O}(h^4), \mathcal{O}(h)]^T$. Thus,

$$\frac{1}{h^4}\widehat{T}_N\widehat{S}_N\mathbf{v} + f_1 = \lambda\mathbf{v} \tag{25}$$

Omitting the f_1 term, we obtain the following discretization

$$\widehat{T}_N\widehat{S}_N\mathbf{v}_h = \lambda_h h^4 \mathbf{v}_h. \tag{26}$$

The so derived discretization is in a factorized form, but $\widehat{T}_N$ and $\widehat{S}_N$ are not square matrices. However, they can be reduced to a product of square matrices as follows.

$$\begin{aligned} B_h := \widehat{T}_N\widehat{S}_N &= T_N^2 - \left(\begin{array}{ccc} 2 & -1 & \\ \hline & & \\ & 0 & \\ & & \\ \hline & -1 & 2 \end{array}\right) \\ &= T_N^2 - E_N T_N = S_N T_N \end{aligned} \tag{27}$$

where

$$E_N = \left(\begin{array}{ccc} 1 & 0 & \\ \hline & & \\ & 0 & \\ & & \\ \hline & 0 & 1 \end{array}\right) \text{ and } S_N = T_N - E_N = \begin{pmatrix} 1 & -1 & & & \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ & & & -1 & 1 \end{pmatrix}$$

are both $N \times N$ matrices. Now, both S_N and T_N are diagonally dominant and can be factorized accurately using Algorithm 1.

Note that the local truncation error f_1 of (24) for this discretization is of order h^2 except at the boundary points where it is of order h^{-1}. By [10, 42], this is sufficient to imply a second order approximation of eigenvalues, namely, for any eigenvalue of the differential operator[3], there is some eigenvalue of B_h such that $|\lambda_h - \lambda| \leq Ch^2$ where C is a constant dependent on the eigenvector v (specifically, $v^{(6)}$). Namely, the finite difference method considers any operator eigenvalue λ and approximates its associated equation (25) by (26), from which *some* eigenvalue λ_h of B_h is found to approximate λ. It is not given that the pairing of λ and λ_h follows any ordering of the eigenvalue sequences. (In contrast, projection methods such as finite elements have each eigenvalue of the discretization, say the i-th smallest,

[3] This may appears impossible with only a finite number of eigenvalues of a discretization matrix but as explained by Keller [41], the constant C in the bound depends on the smoothness of v and for a higher eigenvalue, their corresponding eigenvector is highly oscillatory and thus C is large enough that the bound Ch^2 is only meaningful for correspondingly small h.

approximating the i-th smallest eigenvalue of the operator as $h \to 0$ as implied from the minimax theorem.) Then, B_h may have eigenvalues that do not approximate any eigenvalue of the differential operator as $h \to 0$. Such eigenvalues are called spurious eigenvalues. In our particular discretization, it is easy to show that 0 is always an eigenvalue of B_h of multiplicity 1 (the next theorem) and is therefore a spurious eigenvalue as the differential operator does not posses any zero eigenvalue. Thus, in actual computations, we need to deflate the zero eigenvalue and compute the smallest nonzero eigenvalues of B_h.

Theorem 3 *$B_h = S_N T_N$ is diagonalizable with real eigenvalues and 0 is a simple eigenvalue with $e = [1, 1, \cdots, 1]^T$ as a left eigenvector and $v_0 = T_N^{-1} e$ as a right eigenvector. Furthermore, letting $P_0 = I - v_0 e^T / e^T v_0$ be the spectral projection associated with the nonzero eigenvalue, then $B_h \big|_{\mathcal{R}(P_0)}$, the restriction of B_h on $\mathcal{R}(P_0)$ (the spectral subspace complementary to span$\{v_0\}$), is invertible with its eigenvalues being the nonzero eigenvalues of B_h.*

Proof By writing $B_h = S_N T_N = T_N^{-1/2}(T_N^{1/2} S_N T_N^{1/2}) T_N^{1/2}$, B_h is diagonalizable with real eigenvalues. It is straightforward to check that $B_h v_0 = 0$ and $e^T B_h = 0$. It follows from rank$(S_N) = N-1$ that 0 is a simple eigenvalue of B_h. The invertibility of $B_h \big|_{\mathcal{R}(P_0)}$ and its spectral decomposition follows from the eigenvalue decomposition of B_h. □

Remark: The invertibility of $B_h \big|_{\mathcal{R}(P_0)}$ implies that $B_h x = y$ has a unique solution $x \in \mathcal{R}(P_0)$ for any $y \in \mathcal{R}(P_0)$. In particular, the eigenvalues of the inverse are the reciprocals of the non-zero eigenvalues of B_h. Also, the unique solution to $B_h x = y$ can also be expressed as $x = B_h^D y \in \mathcal{R}(P_0)$ using the Drazin inverse B_h^D.

To compute a few smallest nonzero eigenvalue of B_h, we can use the deflation by restriction to $\mathcal{R}(P_0)$. Namely, we compute the eigenvalues of $B_h \big|_{\mathcal{R}(P_0)}$. Indeed, to compute them accurately, we compute a few largest eigenvalues of the inverse of $B_h \big|_{\mathcal{R}(P_0)}$, which can be computed accurately as follows.

First note that using the accurate LDL^T factorization of T_N, $v_0 = T_N^{-1} e$ can be computed accurately. To apply the inverse of $B_h \big|_{\mathcal{R}(P_0)}$ on a vector $y \in \mathcal{R}(P_0)$, we need to solve

$$B_h x = y \quad \text{for} \quad x, y \in \mathcal{R}(P_0).$$

Since S_N is also diagonally dominant, we first compute an accurate LDL^T factorization $S_N = L_s D_s L_s^T$. Indeed, for this matrix, we have the factorization exactly with

$$L_s = \begin{pmatrix} 1 & & & \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix} \quad \text{and} \quad D_s = \text{diag}\{1, \cdots, 1, 0\}. \tag{28}$$

Then $L_s^T e = e_N$, which also follows from $S_N e = 0$, where $e_N = [0, \cdots, 0, 1]^T$. Thus, for $y \in \mathcal{R}(P_0)$, we have $e_N^T(L_s^{-1}y) = e^T y = 0$; namely the last entry of $L_s^{-1}y$ is zero. Thus a particular solution to $B_h x = y$ that is not necessarily in $\mathcal{R}(P_0)$ can be obtained by solving $L_s x_1 = y$, $D_s x_2 = x_1$ (by setting $x_2(i) = x_1(i)$ for $1 \leq i \leq N-1$, and $x_2(N) = 0$), $L_s^T x_3 = x_2$ and $T_N x = x_3$ (by using the accurate LDL^T factorization of T_N). Since $\text{Ker}(B_h) = \text{span}\{v_0\}$, the general solution to $B_h x = y$ is $x(t) = x + tv_0$. Now, $x(t)$ is in $\mathcal{R}(P_0)$ if and only if $t = -e^T x / e^T v_0$. Thus, $x(t) = x - v_0 \frac{e^T x}{e^T v_0} = P_0 x$ is the unique solution to $B_h x = y$ that is in $\mathcal{R}(P_0)$. We summarize this as the following algorithm.

Algorithm 2 Compute $x = \left(B_h|_{\mathcal{R}(P_0)}\right)^{-1} y$ *(solve $B_h x = y$ for $x, y \in \mathcal{R}(P_0)$)*

```
1  Input: $y \in \mathcal{R}(P_0)$;
2  Compute an accurate $LDL^T$ factorization $T_N = L_t D_t L_t^T$ by Algorithm 1;
3  Solve $T_N v_0 = e$;
4  Solve $L_s x_1 = y$; $x_2 = x_1$ and explicitly set $x_2(N) = 0$; ($L_s$ defined by (28))
5  Solve $L_s^T x_3 = x_2$;
6  Solve $T_N x = x_3$ using $T_N = L_t D_t L_t^T$;
7  $x = x - v_0 e^T x / e^T v_0$.
```

Remarks: Lines 2 and 3 only need to be computed once if we run the above algorithm multiple times for different y. In Line 4, although $x_1(N) = 0$ in theory, it may not be the case due to roundoffs. We therefore explicitly set $x_2(N) = 0$.

We now present some numerical results.

Example 4: Consider the biharmonic eigenvalue problem (20) with the Dirichlet boundary condition (21). The eigenvalues are not known exactly for this problem, but solving it as an initial value problem, we can reduce it to an algebraic equation transcendental in the eigenvalue parameter [32]. The root of the transcendental can then be computed using Mathematica's build-in root-finding routine in high precision. The following is the computed result using 50 digits as obtained by M. Embree [32]:

$$\lambda_1 \approx 500.56390174043259597023906145469523385520808092739.$$

We now compute λ_1 using the difference schemes A_h of (22) and B_h of (27). The eigenvalues of A_h and B_h are computed using inverse iteration[4] with A_h^{-1} computed using the Cholesky factorization and $\left(B_h|_{\mathcal{R}(P_0)}\right)^{-1}$ computed by Algorithm 2. We list the computed results for mesh size $h = 2^{-k}$ with $k = 4, 5, \cdots, 19$ in Table 3. Also listed are the relative errors as computed using the λ_1 above.

We note that both discretizations converge in the order of h^2. For the standard discretization A_h, the eigenvalue errors decreases for h up to 10^{-3} and starts to

[4] We have also carried out tests using the Lanczos algorithm with similar results.

Table 3: Example 4: approximations of λ_1: μ_1^{chol} - A_h with Cholesky; μ_1^{aldu} - B_h with accurate LDL^T.

h	μ_1^{chol} of A_h	$\frac{\lvert\lambda_1-\mu_1^{chol}\rvert}{\lambda_1}$	μ_1^{aldu} of B_h	$\frac{\lvert\lambda_1-\mu_1^{aldu}\rvert}{\lambda_1}$
6.3e-2	4.84875068679297440e2	3.1e-2	5.02539119245910290e2	3.9e-3
3.1e-2	4.96560468599189620e2	8.0e-3	5.01071514661422610e2	1.0e-3
1.6e-2	4.99557827787957420e2	2.0e-3	5.00691660365858750e2	2.6e-4
7.8e-3	5.00312055034394060e2	5.0e-4	5.00595894739436520e2	6.4e-5
3.9e-3	5.00500919775768520e2	1.3e-4	5.00571903322230300e2	1.6e-5
2.0e-3	5.00548153582656820e2	3.1e-5	5.00565902344106350e2	4.0e-6
9.8e-4	5.00559945766432860e2	7.9e-6	5.00564401904366210e2	1.0e-6
4.9e-4	5.00562428167766880e2	2.9e-6	5.00564026782234690e2	2.5e-7
2.4e-4	5.00570856305839580e2	1.4e-5	5.00563933000933900e2	6.2e-8
1.2e-4	5.00646588821035950e2	1.7e-4	5.00563909555575040e2	1.6e-8
6.1e-5	5.00733625655273670e2	3.4e-4	5.00563903694203870e2	3.9e-9
3.1e-5	5.48097497225735650e2	9.5e-2	5.00563902228892570e2	9.8e-10
1.5e-5	7.35072209632324980e2	4.7e-1	5.00563901862573060e2	2.4e-10
7.6e-6	1.98724756006599050e3	3.0e0	5.00563901770967450e2	6.1e-11
3.8e-6	2.91400428172778860e3	4.8e0	5.00563901748025440e2	1.5e-11
1.9e-6	-	-	5.00563901742273290e2	3.7e-12

increase afterwards. When $h \approx 10^{-6}$, there is no accuracy left in the computed eigenvalue. For $h = 1.9e - 6$, the matrix was tested as being indefinite by the Cholesky factorization algorithm (marked by the "-" entry in the table). For the new discretization B_h, the order h^2 convergence is maintained for h up to when the eigenvalue reach full machine accuracy with the relative error in the order of $n\mathbf{u} \approx 10^{-11}$. It is also interesting to note that, even before the roundoff errors overtake the discretization errors in the standard scheme A_h, the new discretization B_h results in an eigenvalue error one order of magnitude smaller.

Thus, we have accurately computed the smallest eigenvalues of a 1-dimensional biharmonic operator with the Dirichlet boundary condition by deriving a product form discretization. Our method can be applied to fourth order operator such as

$$\mathcal{L}v := \frac{d^2}{dx^2}\left(p(x)\frac{d^2v}{dx^2}\right) \quad \text{or} \quad \mathcal{L}v := \frac{d^4v}{dx^4} + p(x)\frac{d^2v}{dx^2}.$$

However, we have not been able to generalize it to problems of dimension higher than 1. This appears to be a difficulty intrinsic to the biharmonic operator in 2-dimension or higher. It is known that the biharmonic operator with the Dirichlet boundary condition do not have a so-called *positivity preserving property* in almost all kinds of domains in dimension 2 or higher; see [16, 36, 37] for details. The only exception is when the domain is a ball in $\mathbb{R}^n$ [9], which includes $[0, 1]$ of $\mathbb{R}$. Notice that the two factors of B_h are diagonally dominant M-matrices, from which it follows that B_h has the positivity preserving property. With the operator in 2 dimension not having the positivity preserving property, it appears difficult to derive a discretization of the form of B_h.

5 Concluding Remarks

We have presented a new method to compute a few smallest eigenvalues of large and extremely ill-conditioned matrices that are diagonally dominant or are products of diagonally dominant matrices. This can be used to compute eigenvalues of finite difference discretization of certain differential operators. In particular, the eigenvalues of the 1-dimensional biharmonic operator is accurately computed by deriving a new discretization that can be written as a product of diagonally dominant matrices. Unfortunately, it appears difficult to apply our present techniques to the 2-dimensional problems. For the future works, it will be interesting to further investigate if there is a suitable generalization to the 2-dimensional biharmonic problems. It will also be interesting to study if the present techniques can be used for other discretization methods as well as adaptive techniques [17, 45] for differential operators.

Acknowledgement: I would like to thank Professors Russell Carden, Mark Embree, and Jinchao Xu for their interests and discussions. In particular, Mark Embree provided me with his computed eigenvalue in high precision in Example 4 and Russell Carden pointed me to the reference [37]. I would also like to thank two anonymous referees for their many constructive comments that have improved the presentation of the paper.

References

[1] S. Alfa, J. Xue and Q. Ye, *Entrywise perturbation theory for diagonally dominant M-matrices with applications,* Numer. Math. 90(2002):401-414.

[2] S. Alfa, J. Xue and Q. Ye, *Accurate computation of the smallest eigenvalue of a diagonally dominant M-matrix,* Math. Comp. 71(2002):217-236.

[3] I. Babuska and J. Osborn, *Eigenvalue Problems,* in Handbook of Numerical Analysis, Edited by P.G. Ciarlet and J.L. Lions, North-Holland, Amsterdam, 1991.

[4] J. Barlow and J. Demmel, *Computing accurate eigensystems of scaled diagonally dominant matrices,* SIAM J. Num. Anal., 27(1990):762-791.

[5] L. Bauer and E. Reiss, *Block five diagonal matrices and the fast numerical computation of the biharmonic equation*, Math. Comp. 26 (1972), 311-326.

[6] M. Ben-Artzi, J. Croisille, and D. Fishelov, *A fast direct solver for the biharmonic problem in a rectangular grid,* SIAM J. Scientif. Comp., 31(2008):303-333.

[7] P. E. Bjorstad and B. P. Tjostheim, *Efficient algorithms for solving a fourth order equation with the Spectral-Galerkin method,* SIAM J. Scitif. Comp., 18 (1997), 621-632.

[8] P. E. Bjorstad and B. P. Tjostheim, *High precision solutions of two fourth order eigenvalue problems*, Computing 63(1999), 97-107.

[9] T. Boggio, *Sulle funzioni di Green dordine m,* Rend. Circ. Mat. Palermo 20 (1905), 97-135.

[10] J. H. Bramble, *A second order finite difference analog of the first biharmonic boundary value problem*, Numerische Mathematik, 9(1966), 236-249.

[11] S.C. Brenner, P. Monk, and J. Sun, *C0 interior penalty Galerkin method for biharmonic eigenvalue problems*, preprint available at `http://www.math.mtu.edu/~jiguangs/`.

[12] B. M. Brown, E. B. Davies, P. K. Jimack, and M. D. Mihajlovic, *A numerical investigation of the solution of a class of fourth order eigenvalue problems,* Proc Roy Soc London A, 456 (2000), 1505-1521.

[13] P.G. Ciarlet, *Conforming and nonconforming finite element methods for solving the plate problem.* In: Conference on the Numerical Solution of Differential Equations (G.A. Watson, ed.), pp. 21-31. Berlin: Springer 1974

[14] W. Chen, *Eigenvalue Approximation of the Biharmonic Eigenvalue Problem by Ciarlet-Raviart Scheme*, Numer Methods Partial Differential Eq., 21(2005):512-520.

[15] Chen, G., Coleman, M., Zhou, J., *Analysis of vibration eigenfrequencies of a thin plate by the Keller-Rubinow wave method I: clamped boundary conditions with rectangular or circular geometry.* SIAM J. Appl. Math., 51(1991), 967-983.

[16] C. V. Coffman and R. J. Duffin, *On the structure of biharmonic functions satisfying the clamped condition on a right angle,* Adv. Appl. Math., 1(1950), 373-389.

[17] X. Dai, J. Xu, and A. Zhou, *Convergence and optimal complexity of adaptive finite element eigenvalue computations,* Numer. Math. 110 (2008):313–355

[18] M. Dailey, F. M. Dopico, and Q. Ye, *Relative perturbation theory for diagonally dominant matrices,* SIAM J. Matrix Anal. Appl., 35(2014), 1303-1328.

[19] M. Dailey, F. M. Dopico, and Q. Ye, *New relative perturbation bounds for LDU factorizations of diagonally dominant matrices,* SIAM J. Matrix Anal. Appl., 35(2014), 904-930.

[20] J. Demmel, *Applied Numerical Linear Algebra,* SIAM, Philadelphia, 1997.

[21] J. Demmel, *Accurate SVDs of structured matrices,* SIAM J. Matrix Anal. Appl. 21(1999):562-580.

[22] J. Demmel, M. Gu, S. Eisenstat, I. Slapničar, K. Veselić and Z. Dramč, *Computing the singular value decomposition with high relative accuracy,* Linear Alg. Appl. 299(1999):21-80.

[23] J. Demmel and W. Kahan, *Accurate singular values of bidiagonal matrices,* SIAM J. Sci. Stat. Comput, 11(1990):873-912

[24] J. Demmel and P. Koev, *Accurate SVDs of weakly diagonally dominant M-matrices,* Numer. Math. 98(2004): 99-104.

[25] J. Demmel and K. Veselic, *Jacobi's method is more accurate than QR,* SIAM J. Matrix Anal. Appl., 13(1992):1204-1246.

[26] F. Dopico and P. Koev, *Accurate symmetric rank revealing decompositions and eigen decompositions of symmetric structured matrices*, SIAM J. Matrix Anal. Appl., 28 (2006), 1126-1156.

[27] F. M. Dopico and P. Koev. *Perturbation theory for the LDU factorization and accurate computations for diagonally dominant matrices,* Numer. Math., 119(2011):337-371.

[28] F. Dopico, P. Koev, and J. Molera. *Implicit standard Jacobi gives high relative accuracy,* Numer. Math., 113(2009):519–553.

[29] F. M. Dopico and J. M. Molera, *Accurate solution of structured linear systems via rank-revealing decompositions*, IMA J. Numer. Anal., 32 (2012):1096-1116.

[30] Z. Drmac, *Computing eigenvalues and singular values to high relative accuracy,* in *Handbook of Linear Algebra,* 2nd Ed., CRC Press, Boca Raton, Fl. 2014.

[31] L. Ehrlich. *Solving the biharmonic equation as coupled finite difference equations,* SIAM Journal on Numerical Analysis, 8(1971):278-287.

[32] M. Embree, private communication.

[33] K. Fernando and B. Parlett, *Accurate singular values and differential qd algorithms,* Numerische Mathematik, 67 (1994):191-229.

[34] D. Fishelov, M. Ben-Artzi, and J.-P. Croisille, *Recent advances in the study of a fourth-order compact scheme for the one-dimensional biharmonic equation*, Journal of Scientific Computing, 53(2012):55–79.

[35] W.K. Grassmann, M.J. Taksar and D.P. Heyman, *Regenerative analysis and steady-state distributions for Markov chains,* Operations Research, 33(1985):1107-1116.

[36] P.R. Garabedian, *A partial differential equation arising in conformal mapping,* Pacific J. Math. 1 (1951), 485-524.

[37] H. Grunau and F. Robert, *Positivity and almost positivity of biharmonic Greens functions under dirichlet boundary conditions,* Arch. Rational Mech. Anal. 195 (2010):865-898.

[38] G. Golub and C. Van Loan, *Matrix Computations,* 2nd ed., The Johns Hopkins University Press, Baltimore, MD, 1989.

[39] W. Hackbusch and G. Hoffmann, *Results of the eigenvalue problem for the plate equation,* ZAMP 31(1980), 730-739.

[40] Y. Jiang, B. Wang, and Y. Xu. *A fast fourier–galerkin method solving a boundary integral equation for the biharmonic equation*, SIAM Journal on Numerical Analysis, 52(2014):2530-2554.

[41] H. B. Keller, *On the acuracy of finite difference approximations to the eigenvalues of differential and integral operators*, Numer. Math. 7(1965):412-419.

[42] J. R. Kuttler, *A finite-difference approximation for the eigenvalues of the clamped plate*, Numerische Mathematik, 17(1971), 230-238.

[43] R.C. Li, *Relative perturbation theory I: eigenvalue and singular value variations,* SIAM J. Matrix Anal. Appl. 19 (1998), 956–982.

[44] A.R. Mitchell and D.F. Griffiths, *The Finite Diffrence Method in Partial Differential Equations,* John Wiley & Son, New York, 1980.

[45] V. Mehrmann and A. Miedlar, *Adaptive computation of smallest eigenvalues of elliptic partial differential equations*, Numerical Linear Algebra with Applications, 18(2011):387-409.

[46] B.N. Parlett, *The Symmetric Eigenvalue Problem,* SIAM, Philadelphia, PA, 1998.

[47] J. M. Pena, *LDU decompositions with L and U well conditioned,* Electr. Trans. Numer. Anal., 18 (2004): 198-208.

[48] R. Rannacher, *Nonconforming finite element methods for eigenvalue problem in linear plate theory,* Numer Math. 33 (1979), 23-42.

[49] R.S. Varga, *Matrix Iterative Analysis,* 2nd ed. Springer-Verlag, Berlin, 2000.

[50] H. Weinberger, *Variational Methods for Eigenvalue Approximation*, CBMS 15, SIAM Publications, 1974.

[51] Q. Ye, *Computing SVD of Diagonally Dominant Matrices to high relative accuracy*, Mathematics of Computations, 77 (2008): 2195-2230.

[52] Q. Ye. *Relative perturbation bounds for eigenvalues of symmetric positive definite diagonally dominant matrices*, SIAM J. Matrix Anal. Appl., 31(2009):11-17.

[53] S. Zhang and J. Xu. *Optimal solvers for fourth-order PDEs discretized on unstructured grids*, SIAM Journal on Numerical Analysis, 52(2014):282-307.