Preconditioning for Accurate Solutions of Linear Systems and Eigenvalue Problems *

Qiang Ye†

Abstract

This paper develops the preconditioning technique as a method to address the accuracy issue caused by ill-conditioning. Given a preconditioner M for an ill-conditioned linear system $Ax = b$, we show that, if the inverse of the preconditioner M^{-1} can be applied to vectors *accurately*, then the linear system can be solved *accurately*. A stability concept called *inverse-equivalent* accuracy is introduced to describe higher accuracy that is achieved and an error analysis will be presented. As an application, we use the preconditioning approach to accurately compute a few smallest eigenvalues of certain ill-conditioned matrices. Numerical examples are presented to illustrate the error analysis and the performance of the methods.

1 Introduction

Solutions of large scale linear algebra problems are typically associated with an ill-conditioned matrix A where the condition number $\kappa(A) := \|A\|\|A^{-1}\|$ is large. The ill-conditioning has two effects in numerically solving a linear system $Ax = b$. It reduces the rate of convergence of iterative algorithms such as the Krylov subspace methods. It also limits the accuracy to which $Ax = b$ can be solved in finite precision. The former problem is typically addressed by a technique known as preconditioning. For the latter, there is no known good solution other than the classical diagonal scaling or iterative refinements; see [11, Sec. 2.5] and [25, p.124].

While a large condition number $\kappa(A)$ is typically associated with the two difficulties discussed above in solving linear systems, it also causes two similar problems for eigenvalue computations. First, a large $\kappa(A)$ is often associated with a spectrum that has one or both ends clustered, which results in slow convergence for

*2010 Mathematics Subject Classification: 65F08, 65F15, 65F35, 65G50. Key words: Preconditioning; ill-conditioned linear systems; accuracy; error analysis; eigenvalue.

†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.

methods such as the Lanczos/Arnoldi algorithms. A large $\kappa(A)$ also limits the accuracy of those smaller eigenvalues of A computed in finite precision; see [13, 14] or §4 for some discussions. The shift-and-invert transformation and its variants are efficient ways of dealing with clustering; see [2] for example. The relative accuracy issue has also been studied extensively and several algorithms have been developed for various structured matrices for which all singular values or eigenvalues can be computed to an accuracy independent of the condition number; see [1, 3, 12, 13, 14, 15, 16, 17, 18, 19, 21, 22, 27, 34] and the references contained therein.

The preconditioning technique is a general methodology that has been highly successful in overcoming the effect of ill-conditioning on the speed of convergence of iterative methods for solving a linear system $Ax = b$. Given an invertible $M \approx A$, we implicitly transform the linear system to the well-conditioned one, $M^{-1}Ax = M^{-1}b$, which can be solved iteratively with accelerated convergence. This poses the natural question: Do we also obtain a more accurate solution by solving the preconditioned system $M^{-1}Ax = M^{-1}b$? The answer is generally no. Because M is a good preconditioner to an ill-conditioned A, it is necessarily ill-conditioned and hence there are potentially large roundoff errors encountered in forming the preconditioned system either explicitly or implicitly; see §3 and §5 for more details and examples. On the other hand, if $M^{-1}Ax = M^{-1}b$ can be formed exactly or sufficiently accurately, solving that will clearly give an accurate solution. Indeed, diagonal scaling is one such example where M is chosen to be a diagonal matrix of powers of 2 so that no roundoff error is generated when applying M^{-1}. Thus, the goal of this paper is to investigate to what accuracy inverting M in preconditioning can lead to improved solution accuracy.

We will develop the preconditioning technique as a method to solve the accuracy issue caused by ill-conditioning for both linear systems and eigenvalue problems. We will show that preconditioning can indeed lead to highly satisfactory solution accuracy of a linear system if the inverse of the preconditioner, M^{-1}, can be applied sufficiently *accurately*. To study precisely the accuracy that is needed for M^{-1} and that can be attained by the final solution, we will introduce a stability concept called *inverse-equivalent* accuracy, which is one equivalent to multiplying exact inverses. An error analysis together with numerical examples will be presented to demonstrate the stability gained. While the present paper is focused on linear systems, we will also use this accurate preconditioning method to accurately compute a few smallest eigenvalues of an ill-conditioned matrix through the accurate inverse approach presented in [36].

We remark that the only requirement for the accurate preconditioning process is that M be inverted with the *inverse-equivalent* accuracy. This can be done if M^{-1} is explicitly available or M has an accurate rank-revealing decomposition (see [13, 20]). In [13], several classes of structured matrices have been shown to have an

accurate rank-revealing decomposition, which include graded matrices, total signed compound matrices such as acyclic matrices, Cauchy matrices, totally positive matrices, diagonally scaled totally unimodular matrices, and matrices arising in certain simple finite element problems. We have also shown in [35] that diagonally dominant matrices have an accurate rank-revealing decomposition. Thus, the accurate preconditioning method is applicable to a broad class of matrices that can be well preconditioned by any of these structured matrices.

The rest of the paper is organized as follows. We present in §2 the concept of *inverse-accurate* accuracy. We then develop in §3 the accurate preconditioning method and an error analysis for linear systems. In §4, we discuss applying the accurate preconditioning method to accurately compute a few smallest eigenvalues of a matrix. Finally, in §5, we present some numerical examples for both linear systems and eigenvalue problems, followed by some concluding remarks in §6.

1.1 Notation and Preliminaries

Throughout this paper, $\|\cdot\|$ denotes a general norm for vectors and its induced operator norm for matrices. $\|\cdot\|_p$ denotes the p-norm. Inequalities and absolute value involving matrices and vectors are entrywise.

For error analysis in a floating point arithmetic, $\mathbf{u}$ denotes 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. We assume throughout that matrices and vectors given have floating point number entries. We assume the following standard model for roundoff errors in basic matrix computations [25, p.66]:

$$fl(x+y) = x+y+e \quad \text{with} \quad |e| \le \mathbf{u}(|x+y|) \tag{1}$$

$$fl(Ax) = Ax+e \quad \text{with} \quad |e| \le \mathbf{u}N|A||x| + \mathcal{O}(\mathbf{u}^2), \tag{2}$$

where N is the maximal number of nonzero entries per row of A. Using (2.4.12) of [25, p.64] and equivalence of any two norms in a finite dimensional space, we may also simply rewrite (2) as

$$\|fl(Ax) - Ax\| \le \mathcal{O}(\mathbf{u})N\|A\|\|x\|. \tag{3}$$

This bound is based on explicitly multiplying A with x and $N \le n$ can be absorbed into the $\mathcal{O}(\mathbf{u})$ term. More generally, if A is not explicitly given and Ax is computed as an operator, (3) may still be valid if we allow N to be a suitable constant associated with the operator Ax.

2 Inverse-equivalent Accuracy

In this section, we introduce a stability concept called *inverse-equivalent* accuracy for solving linear systems in finite precision.

Given an invertible matrix $A \in \mathbb{R}^{n\times n}$ and $b \in R^n$, all standard dense algorithms for solving the linear system $Ax = b$ in a floating point arithmetic computes a solution $\hat{x}$ that is backward stable, i.e. it satisfies $(A + E)\hat{x} = b$ for some E with $\|E\|/\|A\| = \mathcal{O}(\mathbf{u})$. An iterative method computes a solution $\hat{x}$ with a residual that at best satisfies $\|b - A\hat{x}\| = \mathcal{O}(\mathbf{u})\|A\|\|\hat{x}\|$, which is equivalent to the backward stability. In both cases, the solution error is bounded as

$$\frac{\|\hat{x} - x\|}{\|x\|} \leq \mathcal{O}(\mathbf{u})\kappa(A), \quad \text{where} \quad \kappa(A) = \|A\|\|A^{-1}\|. \tag{4}$$

This backward stable solution accuracy may be unsatisfactory for ill-conditioned problems, but for a general linear system, this is the best one may hope for because the solution is not well determined by the matrix A under perturbations. For many ill-conditioned linear systems arising in applications, however, the underlying solution may be much more stable when considering the solution as determined from the original problem data rather than from the matrix. For example, discretization of a differential equation typically gives rise to an ill-conditioned linear system, but its solution, approximating the solution of PDE, is stably determined by the input data of the PDE. Namely, the solution is stable if we only consider perturbations to the problem data in spite of ill-conditioning of the matrix. In that case, we are interested in special algorithms that can solve such ill-conditioned linear systems more accurately.

Before we study algorithms, we first address how high an accuracy one may reasonably expect to achieve for a linear system. Ideally, we may strive for the full relative accuracy

$$\frac{\|\hat{x} - x\|}{\|x\|} \leq \mathcal{O}(\mathbf{u}) \tag{5}$$

but a bound totally independent of A will obviously require very stringent conditions on A, as a perturbation to b alone will produce errors proportional to A^{-1}. Note that b typically corresponds to problem data and then some perturbations/uncertainties in b should be assumed. Furthermore, the ideal accuracy (5) may not be necessary in many applications. Indeed, the accuracy we introduce now is often sufficient in applications.

Definition 1 *Given A, we say that an algorithm for solving linear systems with coefficient A is inverse-equivalent if, for any b, it produces in a floating point arithmetic a computed solution $\hat{x}$ to $Ax = b$ such that*

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

We also say such a solution $\hat{x}$ has an inverse-equivalent accuracy.

In the definition, we have used a general norm. Since any two norms are equivalent and $\mathcal{O}(\mathbf{u})$ can absorb any constant, the definition is equivalent to one using any particular norm in (6). The next two results explain the naming of this accuracy.

Theorem 1 *If A is such that A^{-1} is explicitly available, then solving $Ax = b$ by multiplying A^{-1} with b is an inverse-equivalent algorithm.*

Proof Recall that A and b are assumed to have floating point number entries. For A^{-1}, we have $|fl(A^{-1}) - A^{-1}| \leq \mathbf{u}|A^{-1}|$. Then $\|fl(A^{-1})b - A^{-1}b\| \leq \mathcal{O}(\mathbf{u})\|A^{-1}\|\|b\|$. It follows from (2) that $\|fl(A^{-1}b) - fl(A^{-1})b\| \leq \mathcal{O}(\mathbf{u})\|fl(A^{-1})\|\|b\|$. Combining the two, we obtain $\|fl(A^{-1}b) - A^{-1}b\| \leq \mathcal{O}(\mathbf{u})\|A^{-1}\|\|b\|$. □

Theorem 2 *Let A be an invertible matrix. There is an inverse-equivalent algorithm for A if and only if the inverse A^{-1} can be computed by some algorithm with a relative error of order $\mathcal{O}(\mathbf{u})$, i.e. the computed inverse $\widehat{X}$ satisfies*

$$\frac{\|\widehat{X} - A^{-1}\|}{\|A^{-1}\|} \leq \mathcal{O}(\mathbf{u}) \tag{7}$$

Proof First assume that there is an inverse-equivalent algorithm for A. Using this algorithm to compute the inverse A^{-1} by solving $AX = I$, let the computed inverse be $\widehat{X} = [\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_n]$ and write $X = A^{-1} = [x_1, x_2, \cdots, x_n]$. Then $\hat{x}_i$ is inverse-equivalent, i.e., written in the 1-norm, $\|\hat{x}_i - x_i\|_1 \leq \mathcal{O}(\mathbf{u})\|A^{-1}\|_1\|e_i\|_1 = \mathcal{O}(\mathbf{u})\|A^{-1}\|_1$. Thus $\|\widehat{X} - X\|_1 = \max_i \|\hat{x}_i - x_i\|_1 \leq \mathcal{O}(\mathbf{u})\|A^{-1}\|_1$. By equivalence of norms, (7) is proved.

On the other hand, if we have an algorithm that computes the inverse $\widehat{X}$ satisfies (7), then for any b, solving $Ax = b$ by computing $\hat{x} = fl(\widehat{X}b)$, we have

$$\begin{aligned}
\|\hat{x} - x\| &\leq \|fl(\widehat{X}b) - \widehat{X}b\| + \|\widehat{X}b - A^{-1}b\| \\
&\leq \mathcal{O}(\mathbf{u})\|\widehat{X}\|\|b\| + \mathcal{O}(\mathbf{u})\|A^{-1}\|\|b\| \\
&\leq \mathcal{O}(\mathbf{u})(\|\widehat{X} - A^{-1}\| + \|A^{-1}\|)\|b\| + \mathcal{O}(\mathbf{u})\|A^{-1}\|\|b\| \\
&\leq \mathcal{O}(\mathbf{u})(\mathcal{O}(\mathbf{u})\|A^{-1}\| + \|A^{-1}\|)\|b\| + \mathcal{O}(\mathbf{u})\|A^{-1}\|\|b\| \\
&= \mathcal{O}(\mathbf{u})\|A^{-1}\|\|b\|.
\end{aligned}$$

So the algorithm $\hat{x} = fl(\widehat{X}b)$ is inverse-equivalent. This completes the proof. □

The above shows that an inverse-equivalent algorithm produces solution that are comparable to the one obtained by multiplying the exact inverse with the right-hand side vector b. This should be highly satisfactory in many applications. For

example, in eigenvalue computations with the shift-and-invert transformation, using an inverse-equivalent algorithm for the inverse would produce results as accurate as the one obtained using the exact inverse; see §4.

If we rewrite (6) in the relative error form

$$\frac{\|\hat{x} - x\|}{\|x\|} \leq \mathcal{O}(\mathbf{u}) \frac{\|A^{-1}\|\|b\|}{\|x\|}, \tag{8}$$

then it is clear that this accuracy is between the full relative accuracy (5) and the backward stable solution accuracy (4) as $\|x\| \leq \|A^{-1}\|\|b\| \leq \|A^{-1}\|\|A\|\|x\|$. Note that the bound (8) has also appeared in the study of perturbation theory for $Ax = b$ when only the right-hand side vector b is perturbed; see [8, 26]. It has been observed that the bound (8) may be substantially smaller than (4); see [8, 20]. For example, this occurs as long as b has a significant projection on some right singular vector u_k of A corresponding to a singular value σ_k that is far less than the largest one. Namely, if $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$ are the singular values of A, then $\|x\|_2 = \|A^{-1}b\|_2 \geq |u_k^T b|/\sigma_k$ and hence

$$\frac{\|A^{-1}\|_2\|b\|_2}{\|x\|_2} \leq \frac{\sigma_k}{\sigma_n} \frac{\|b\|_2}{|u_k^T b|} \ll \|A\|_2\|A^{-1}\|_2 \tag{9}$$

if

$$\frac{\sigma_k}{\cos \angle(b, u_k)} \ll \sigma_1. \tag{10}$$

See [8, 20] for some more detailed discussions.

We remark that b, being the input data in a practical problem, is unlikely to be nearly orthogonal to all singular vectors corresponding to smaller singular values. For example, if b is a random vector, (10) may be easily satisfied. So we may expect the inverse-equivalent accuracy (8) to be significantly better than the backward stable one (4) when b is chosen with no constraint.

3 Accurate solutions for linear systems

In this section, we present an accurate preconditioning method for solving a linear system where the inversion of the preconditioner is computed with an inverse-equivalent algorithm. We show that this results in inverse-equivalent accuracy and we present our analysis in two subsections, one for direct methods and one for iterative ones for solving the preconditioned equation. We first briefly discuss the accuracy that may be expected when a standard backward stable algorithm is used for the preconditioner.

Preconditioning a linear system $Ax = b$ is commonly used to accelerate convergence of an iterative method. Given a preconditioner $M \approx A$ such that $M^{-1}A$ is

well-conditioned, applying an iterative method to $M^{-1}Ax = M^{-1}b$ results in accelerated convergence. Since $M^{-1}Ax = M^{-1}b$ is a well-conditioned system, it might be argued that solving the preconditioned equation should produce more accurate solutions. However, inverting M encounters roundoff errors which change the preconditioned system and the final solution. We analyze this error as follows.

First, we observe that for $M^{-1}A$ to be well-conditioned, M is necessarily ill-conditioned (i.e. has a condition number comparable to A). This is because

$$\frac{\kappa(A)}{\kappa(M^{-1}A)} \leq \kappa(M) \leq \kappa(M^{-1}A)\kappa(A). \tag{11}$$

Then the application of M^{-1} on A and on b can not be computed accurately. For example, assuming M is inverted by a backward stable algorithm, the computed result of the right-hand side $M^{-1}b$ is $M^{-1}b + f$ with the error f bounded by $\|f\|/\|M^{-1}b\| = \mathcal{O}(\mathbf{u})\kappa(M)$. Similarly, the computed result of $M^{-1}A$ is $M^{-1}A + E$ with $\|E\|/\|M^{-1}A\| = \mathcal{O}(\mathbf{u})\kappa(M)$. Thus, the preconditioned system obtained is

$$(M^{-1}A + E)y = M^{-1}b + f, \tag{12}$$

and then even its exact solution y can only be bounded as

$$\frac{\|y - x\|}{\|x\|} \leq \mathcal{O}(\mathbf{u})\kappa(M)\kappa(M^{-1}A) \tag{13}$$

which by (11) is approximately $\mathcal{O}(\mathbf{u})\kappa(A)$. We conclude that the computed solution to $M^{-1}Ax = M^{-1}b$, after accounting the errors of inverting M, has a relative error of order $\mathbf{u}\kappa(A)$. So, the solution accuracy can not be improved by preconditioning in general; see numerical examples in §5.

Note that the discussion above is for a general M solved by a backward stable algorithm. The diagonal scaling, where M is chosen to be a diagonal matrix (typically with entries being powers of 2) [25, p.124], is an effective method for improving solution accuracy provided the diagonal matrix is a good preconditioner. With such a preconditioner, the preconditioning transformation is performed exactly and the resulting solution accuracy is indeed improved. This leads us to the following questions: Can more accurately inverting M lead to a more accurate solution of the original system, and if so, what accuracy is needed for M^{-1}? The rest of this section provides answers to these questions.

Let $A = M + K$ where K is small in norm and M is such that there is an inverse-equivalent algorithm for inverting M. Then using M as a preconditioner, we form the preconditioned system

$$Bx = c, \quad \text{where} \quad B := I + M^{-1}K, \;\; c := M^{-1}b. \tag{14}$$

This system may be formed explicitly or implicitly depending on whether we solve it by a direct or an iterative method respectively, but it is important that B or its product with vectors is formed in the way as given in (14). We call this process accurate preconditioning and we will show that solving the well-conditioned system (14) by any backward stable algorithm leads to an inverse-equivalent accurate solution (6). Namely, *accurate preconditioning with an inverse-equivalent algorithm for inverting M is an inverse-equivalent algorithm for A.*

The following two subsections provide detailed analysis by considering solving (14) first using a direct method and then using an iterative one.

3.1 Direct Method for Preconditioned Systems

We consider forming (14) explicitly and then solving it by a backward stable direct method such as the Gaussian elimination with partial pivoting. In this regard, we first need to compute $M^{-1}K$ column by column by solving n linear systems. Assume that these linear systems are solved by an inverse-equivalent algorithm for M. Then, each column of the computed result of $M^{-1}K$ has inverse-equivalent accuracy. We denote the computed result as $\widehat{Z}$ and it satisfies $\|\widehat{Z} - M^{-1}K\| \leq \mathcal{O}(\mathbf{u})\|M^{-1}\|\|K\|$. Furthermore the coefficient matrix $B = I + M^{-1}K$ is computed as $fl(I + \widehat{Z})$, which has an error term bounded by $\mathbf{u}(1 + \|\widehat{Z}\|)$ by (1). Combining the two error terms together and denoting the final computed result $fl(I + \widehat{Z})$ as $\widehat{B}$, we can write the total error as

$$\widehat{B} = I + M^{-1}K + E = B + E, \quad \text{with} \quad \|E\| \leq \mathcal{O}(\mathbf{u})(1 + \|M^{-1}\|\|K\|). \tag{15}$$

Similarly, the computed result of $M^{-1}b$, denoted by $\hat{c} := fl(M^{-1}b)$ satisfies

$$\|\hat{c} - c\| \leq \mathcal{O}(\mathbf{u})\|M^{-1}\|\|b\|. \tag{16}$$

Theorem 3 *Let $A = M+K$ with A and M being invertible and let $Ax = b$. Assume that there is an inverse-equivalent algorithm for inverting M so that the computed results of $B := I + M^{-1}K$ and $c := M^{-1}b$, denoted by $\widehat{B}$ and $\hat{c}$ respectively, satisfy (15) and (16). Let $\hat{x}$ be the computed solution to $\widehat{B}x = \hat{c}$ by a backward stable algorithm so that $\hat{x}$ satisfies*

$$(\widehat{B} + F)\hat{x} = \hat{c}, \quad \textit{with} \quad \frac{\|F\|}{\|\widehat{B}\|} \leq \mathcal{O}(\mathbf{u}). \tag{17}$$

Let $\delta := (\|E\| + \|F\|)\|B^{-1}\|$ and assume that $\delta < 1$. Then

$$\frac{\|\hat{x} - x\|}{\|A^{-1}\|\|b\|} \leq \mathcal{O}(\mathbf{u})\frac{\kappa(B)}{1 - \delta}\left(4 + \frac{\|K\|\|x\|}{\|b\|}\right). \tag{18}$$

In particular, if $\|M^{-1}\|\|K\| < 1$, *then*

$$\frac{\|\hat{x} - x\|}{\|A^{-1}\|\|b\|} \leq \frac{\mathcal{O}(\mathbf{u})}{(1-\delta)(1-\|M^{-1}\|\|K\|)^2}.$$

Proof First, let $f = \hat{c} - c$ or $\hat{c} = c + f$. Then $\|f\| \leq \mathcal{O}(\mathbf{u})\|M^{-1}\|\|b\|$. Let $\widetilde{B} = \widehat{B} + F = B + E + F$ and rewrite (17) as $\widetilde{B}\hat{x} = c + f$. From $(\|E\| + \|F\|)\|B^{-1}\| < 1$, it follows that $\widetilde{B}$ is invertible and

$$\|\widetilde{B}^{-1}\| \leq \frac{\|B^{-1}\|}{1 - (\|E\| + \|F\|)\|B^{-1}\|} = \frac{\|B^{-1}\|}{1-\delta} \tag{19}$$

We also have

$$\|M^{-1}\| = \|BA^{-1}\| \leq \|B\|\|A^{-1}\| \tag{20}$$

and

$$\|B^{-1}\|\|\widehat{B}\| \leq \|B^{-1}\|(\|B\| + \|E\|) \leq \|B^{-1}\|\|B\| + \delta \leq 2\|B^{-1}\|\|B\| \tag{21}$$

Furthermore, using

$$1 = \|I\| = \|B - M^{-1}K\| \leq \|B\| + \|M^{-1}\|\|K\|, \tag{22}$$

we can bound (15) as

$$\|E\| \leq \mathcal{O}(\mathbf{u})(\|B\| + 2\|M^{-1}\|\|K\|) = \mathcal{O}(\mathbf{u})(\|B\| + \|M^{-1}\|\|K\|) \tag{23}$$

where in the last equality we have combined the coefficient 2 of $\|M^{-1}\|\|K\|$ into $\mathcal{O}(\mathbf{u})$ (i.e. $2\mathcal{O}(\mathbf{u}) = \mathcal{O}(\mathbf{u})$ with our notation). Now, clearly $Bx = c$ and then $\widetilde{B}x = c + Ex + Fx$. Combining this with $\widetilde{B}\hat{x} = c + f$, we have

$$\hat{x} - x = -\widetilde{B}^{-1}Ex - \widetilde{B}^{-1}Fx + \widetilde{B}^{-1}f.$$

Bounding the above and using (23), (17), (19), (20), and (21), we have

$$\begin{aligned}
\|\hat{x} - x\| &\leq \|\widetilde{B}^{-1}\|\|E\|\|x\| + \|\widetilde{B}^{-1}\|\|F\|\|x\| + \|\widetilde{B}^{-1}\|\|f\| \\
&\leq \mathcal{O}(\mathbf{u})\|\widetilde{B}^{-1}\|(\|B\| + \|M^{-1}\|\|K\|)\|x\| \\
&\quad + \mathcal{O}(\mathbf{u})\|\widetilde{B}^{-1}\|\|\widehat{B}\|\|x\| + \mathcal{O}(\mathbf{u})\|\widetilde{B}^{-1}\|\|M^{-1}\|\|b\| \\
&\leq \frac{\mathcal{O}(\mathbf{u})}{1-\delta}\|B^{-1}\|\|B\|\|x\| + \frac{\mathcal{O}(\mathbf{u})}{1-\delta}\|B^{-1}\|\|M^{-1}\|\|K\|\|x\| \\
&\quad + \frac{\mathcal{O}(\mathbf{u})}{1-\delta}\|B^{-1}\|\|\widehat{B}\|\|x\| + \frac{\mathcal{O}(\mathbf{u})}{1-\delta}\|B^{-1}\|\|M^{-1}\|\|b\| \\
&\leq \frac{\mathcal{O}(\mathbf{u})}{1-\delta}(\|B^{-1}\|\|B\|\|x\| + \|B^{-1}\|\|B\|\|A^{-1}\|\|K\|\|x\| \\
&\quad + 2\|B^{-1}\|\|B\|\|x\| + \|B^{-1}\|\|B\|\|A^{-1}\|\|b\|) \\
&\leq \frac{\mathcal{O}(\mathbf{u})}{1-\delta}\|B^{-1}\|\|B\|\left(3\|x\| + \|A^{-1}\|\|b\|\frac{\|K\|\|x\|}{\|b\|} + \|A^{-1}\|\|b\|\right) \\
&\leq \frac{\mathcal{O}(\mathbf{u})}{1-\delta}\kappa(B)\left(4 + \frac{\|K\|\|x\|}{\|b\|}\right)\|A^{-1}\|\|b\|.
\end{aligned}$$

where we have used $\|x\| \le \|A^{-1}\|\|b\|$ in the last inequality. This proves (18).

Finally, if $\|M^{-1}\|\|K\| < 1$, then $B = I + M^{-1}K$ satisfies $\|B\| \le 1 + \|M^{-1}\|\|K\|$ and $\|B^{-1}\| \le \frac{1}{1-\|M^{-1}\|\|K\|}$. Thus, it follows from $A^{-1} = B^{-1}M^{-1}$ that

$$\frac{\|K\|\|x\|}{\|b\|} \le \|K\|\|A^{-1}\| \le \|K\|\|M^{-1}\|\|B^{-1}\| \le \frac{\|K\|\|M^{-1}\|}{1 - \|K\|\|M^{-1}\|}$$

Thus

$$\kappa(B)\left(4 + \frac{\|K\|\|x\|}{\|b\|}\right) \le \frac{1 + \|M^{-1}\|\|K\|}{1 - \|M^{-1}\|\|K\|}\frac{4 - 3\|M^{-1}\|\|K\|}{1 - \|M^{-1}\|\|K\|} \le \frac{5}{(1 - \|M^{-1}\|\|K\|)^2}$$

where we have used $(1 + \|M^{-1}\|\|K\|)(4 - 3\|M^{-1}\|\|K\|) \le 4 + \|M^{-1}\|\|K\| \le 5$. Substituting this into (18) and combing the factor 5 into the $\mathcal{O}(\mathbf{u})$ term, we obtain the second bound of the theorem. □

The second bound of the theorem shows that we can obtain an inverse-equivalent solution if $\|M^{-1}\|\|K\|$ is bounded away from 1. Note that $\delta = (\|E\| + \|F\|)\|B^{-1}\| \le \mathcal{O}(\mathbf{u})\|B^{-1}\|(1 + \|\widehat{B}\| + \|M^{-1}\|\|K\|)$ can be expected to be much smaller than 1 and hence the factor $(1 - \delta)^{-1}$ is insignificant. When $\|M^{-1}\|\|K\| \ge 1$, only the first bound (18) holds, which implies that the inverse-equivalent accuracy of the solution may deteriorate by a factor of $\kappa(B)$ or $\frac{\|K\|\|x\|}{\|b\|}$. Such a dependence on $\kappa(B)$ and K is expected however, as otherwise there would be inverse-equivalent algorithm for any A.

3.2 Iterative Method for Preconditioned Systems

For large scale problems, we are more interested in solving the preconditioned system $Bx = c$ by an iterative method. In general, the accuracy of the approximate solution obtained by an iterative method for $Ax = b$ is obviously limited by the accuracy of the matrix-vector multiplication Av. Namely, the residual $\|b - A\widehat{x}\|$ of an approximate solution $\widehat{x}$ computed in a floating point arithmetic is at best of order $\mathbf{u}N\|A\|\|\widehat{x}\|$. A careful implementation, possibly using residual replacements [32], can ensure that the residual converges with this level of accuracy. Note that such a solution $\widehat{x}$ is backward stable (see [11, Theorem 2.2]). We first briefly discuss some related results on the best accuracy that can be achieved.

Most iterative methods update approximate solutions and the corresponding residuals at each iteration by general formulas of the forms $x_k = x_{k-1} + q_k$ and $r_k = r_{k-1} - Aq_k$. In a convergent iteration, the best residual $\|b - Ax_k\|$ one may obtain in finite precision is determined by the deviation between the computed (or updated) residual r_k, which is the one computed in an algorithm through the updating formula $r_k = r_{k-1} - Aq_k$, and the true residual defined as $b - Ax_k$ for

x_k that is computed through $x_k = x_{k-1} + q_k$. This deviation phenomenon of the two kinds of residuals has been extensively studied; see [30, 23, 24, 29, 32] and the references cited therein. Typically, the computed residuals r_k of a convergent method maintains the theoretical convergence property (e.g. monotonicity) even in a floating point arithmetic and can decrease arbitrarily close to 0, but the true residuals $b - Ax_k$ will stagnate at some level. This deviation of the two residuals is due to the roundoff errors at each step, the most significant of which, among others, is $\mathcal{O}(\mathbf{u})N\|A\|\|q_k\|$ incurred in computing Aq_k, where N is a constant associated with the error in $fl(Av)$ as defined in (3). Then, for x_L at step L, the deviation is made up of the accumulated deviations over L iterations, i.e. $\mathcal{O}(\mathbf{u})\sum_{k=1}^{K} N\|A\|\|q_k\|$ which, since $x_L = fl(\sum_{k=1}^{L} q_k)$, is at least $\mathcal{O}(\mathbf{u})N\|A\|\|x_L\|$, the error incurred in computing $fl(Ax_L)$.

Indeed, the accumulated roundoff errors $\mathcal{O}(\mathbf{u})\sum_{k=1}^{L} N\|A\|\|q_k\|$, and hence the true residual, may be much larger than $\mathcal{O}(\mathbf{u})N\|A\|\|x_L\|$ if there are large intermediate iterates q_k, which occur often in nonsymmetric solvers such as BiCG and CGS. In that case, a residual replacement strategy [32, Algorithm 3] has been developed that replaces the computed residual by the true residual at some selected steps so that its convergence property remains intact but the deviation of the two residuals is reset to 0. Indeed, it is shown in [32, Theorem 3.6] that if an iterative method for solving $Ax = b$ is implemented with the residual replacement and the algorithm terminates at step L with the computed residual satisfying $\|r_L\| < \mathbf{u}\|A\|\|x_L\|$, then the true residual $\|b - Ax_L\|$ will be in the order of $\mathbf{u}N\|A\|\|x_L\|$.

Now, consider solving the preconditioned system (14) by such an iterative method. To determine the accuracy that can be obtained from solving this well-conditioned system, we first analyze the accuracy of computing matrix-vector multiplication Bv.

Lemma 1 *Let B be defined in (14) and consider computing $Bv = v + M^{-1}Kv$ as in this expression for any $v \in \mathbb{R}^n$. Assume that there is an inverse-equivalent algorithm for inverting M. If $M^{-1}Kv$ is computed by the inverse-equivalent algorithm and if we denote the final computed result of Bv by $fl(Bv)$, then*

$$\|fl(Bv) - Bv\| \leq \mathcal{O}(\mathbf{u})(1 + \|M^{-1}\|\|K\|)\|v\|. \tag{24}$$

Proof Let $u := Bv$ and denote the final computed result $fl(Bv)$ by $\hat{u}$. To compute Bv, we first compute Kv to get $fl(Kv) = Kv + e_1$ with $|e_1| \leq n\mathbf{u}|K||v|$. Then computing $M^{-1}fl(Kv)$ by the inverse-equivalent algorithm, the computed result, denoted by $\hat{w}$, satisfies

$$\begin{aligned} \|\hat{w} - M^{-1}fl(Kv)\| &\leq \mathcal{O}(\mathbf{u})\|M^{-1}\|\|fl(Kv)\| \\ &\leq \mathcal{O}(\mathbf{u})\|M^{-1}\|(\|K\|\|v\| + \mathcal{O}(\mathbf{u})\|K\|\|v\|) \\ &\leq \mathcal{O}(\mathbf{u})\|M^{-1}\|\|K\|\|v\|. \end{aligned}$$

Let $e_2 = \hat{w} - M^{-1}Kv$. Then

$$\begin{aligned} \|e_2\| &= \|\hat{w} - M^{-1}fl(Kv) + M^{-1}e_1\| \\ &\leq \mathcal{O}(\mathbf{u})\|M^{-1}\|\|K\|\|v\| + \mathcal{O}(\mathbf{u})\|M^{-1}\|\|K\|\|v\| \\ &= \mathcal{O}(\mathbf{u})\|M^{-1}\|\|K\|\|v\|. \end{aligned}$$

Now, $\hat{u} = fl(v + \hat{w}) = v + \hat{w} + e_3$ with $|e_3| \leq \mathbf{u}(|v| + |\hat{w}|)$. Then

$$\begin{aligned} \|e_3\| &\leq \mathcal{O}(\mathbf{u})\|v\| + \mathcal{O}(\mathbf{u})(\|M^{-1}Kv\| + \|e_2\|) \\ &\leq \mathcal{O}(\mathbf{u})(\|v\| + \|M^{-1}\|\|K\|\|v\| + \mathcal{O}(\mathbf{u})\|M^{-1}\|\|K\|\|v\|) \\ &= \mathcal{O}(\mathbf{u})(\|v\| + \|M^{-1}\|\|K\|\|v\|). \end{aligned}$$

Finally, we have $\hat{u} = v + M^{-1}Kv + e_2 + e_3 = u + e_2 + e_3$ and

$$\|\hat{u} - u\| \leq \|e_2\| + \|e_3\| \leq \mathcal{O}(\mathbf{u})(1 + \|M^{-1}\|\|K\|)\|v\|.$$

□

Now, when applying some convergent iterative method to the system (14), using the residual replacement strategy if necessary, the true residual $\|c - Bx_L\|$ is expected to converge to $\mathcal{O}(\mathbf{u})(1 + \|M^{-1}\|\|K\|)\|v\|$. The next theorem demonstrate that such a solution has an inverse-equivalent accuracy. Note that since (14) is well-conditioned, most iterative methods should have fast convergence. In that case, the error accumulations are insignificant and the residual replacement is usually not necessary in practice.

Theorem 4 *Consider solving (14) by an iterative method where the matrix-vector product $Bv = v + M^{-1}Kv$ is computed by an inverse-equivalent algorithm for inverting M. Assume that the iterative method produces an approximate solution x_L with $\|c - Bx_L\| \leq \mathcal{O}(\mathbf{u})(1 + \|M^{-1}\|\|K\|)\|v\|$ and $\|b - Ax_L\| \leq \|b\|$. Then*

$$\begin{aligned} \frac{\|x - x_L\|}{\|A^{-1}\|\|b\|} &\leq \mathcal{O}(\mathbf{u})\kappa(B)\left(1 + \frac{\|K\|\|x_L\|}{\|b\|}\right) \\ &\leq \mathcal{O}(\mathbf{u})\kappa(B)\left(1 + 2\|A^{-1}\|\|K\|\right). \end{aligned}$$

Proof First we note that $x_L = x - A^{-1}(b - Ax_L)$ and then

$$\|x_L\| \leq \|x\| + \|A^{-1}\|\|b - Ax_L\| \leq \|x\| + \|A^{-1}\|\|b\| \leq 2\|A^{-1}\|\|b\|.$$

As in the proof of Theorem 3, we have (22). Then

$$\|c - Bx_L\| \leq \mathcal{O}(\mathbf{u})(\|B\|\|x_L\| + 2\|M^{-1}\|\|K\|\|x_L\|).$$

We now bound $x - x_L = B^{-1}(c - Bx_L)$ as

$$\begin{aligned}\|x - x_L\| &\leq \mathcal{O}(\mathbf{u})\|B^{-1}\|(\|B\|\|x_L\| + 2\|M^{-1}\|\|K\|\|x_L\|) \\ &\leq \mathcal{O}(\mathbf{u})\|B^{-1}\|(2\|B\|\|A^{-1}\|\|b\| + 2\|B\|\|A^{-1}\|\|K\|\|x_L\|) \\ &= \mathcal{O}(\mathbf{u})\|A^{-1}\|\|b\|\kappa(B)\left(1 + \frac{\|K\|\|x_L\|}{\|b\|}\right),\end{aligned}$$

where we have used (20) and combine the factor 2 into the $\mathcal{O}(\mathbf{u})$ term. This proves the first bound. Bounding $\|x_L\|$ by $2\|A^{-1}\|\|b\|$ again, we obtain the second bound of the theorem. □

The theorem shows that the inverse-equivalent accuracy is also achieved when using an iterative method for the preconditioned system.

3.3 Accurate Inversion of Preconditioner

The key requirement of the accurate preconditioning method is that there is an inverse-equivalent algorithm for inverting the preconditioner M. This is obviously the case if the inverse M^{-1} is explicitly available. More generally, if a preconditioner M has an *accurate* rank-revealing decomposition (RRD), then the solution to $Mx = b$ computed from the RRD is inverse-equivalent. The *accurate* rank-revealing decomposition is introduced by Demmel et. al. [13] to accurately compute the singular value decomposition of a matrix. Here is its definition.

Definition 2 (See [13]) *A factorization $A = XDY$ of $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ is said to be rank-revealing if $X \in \mathbb{R}^{m\times r}$ and $Y \in \mathbb{R}^{r\times n}$ are well-conditioned and $D \in \mathbb{R}^{r\times r}$ is diagonal and invertible, where $r \leq \min\{m, n\}$. Consider an algorithm for computing a rank-revealing decomposition $A = XDY$ and let $\widehat{X}$, $\widehat{D}$, and $\widehat{Y}$ be the computed factors. We say $\widehat{X}\widehat{D}\widehat{Y}$ is an* accurate rank-revealing decomposition *of A if $\widehat{X}$ and $\widehat{Y}$ are normwise accurate and $\widehat{D}$ is entrywise accurate, i.e.,*

$$\frac{\|\widehat{X} - X\|}{\|X\|} \leq \mathbf{u}p(n); \quad \frac{\|\widehat{Y} - Y\|}{\|Y\|} \leq \mathbf{u}p(n); \qquad \text{and } |\widehat{D} - D| \leq \mathbf{u}p(n)|D|, \tag{25}$$

where $p(n)$ is a polynomial in n.

As noted in [13], the precise meaning of "well-conditioned" in the definition is not important as all related results involving this will be stated in terms of the condition numbers $\kappa(X)$ and $\kappa(Y)$, but in general, it refers to matrices with a condition number within a modest bound dependent on the problem at hand.

For our purpose, we consider $n \times n$ invertible matrices, i.e. $r = n$. Then, if A has an accurate RRD, it is shown by Dopico and Molera [20] that using it to solve linear systems gives an inverse-equivalent algorithm. We state this result in the following theorem.

Theorem 5 ([20, Theorem 4.2]) *Let $\widehat{X}$, $\widehat{D}$, and $\widehat{Y}$ be the computed factors of a rank-revealing decomposition of $A = XDY$ and assume that they satisfy (25). 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 polynomial in 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*

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

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

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

Several classes of matrices have been shown to have accurate RRD by Demmel et. al. [13], which include graded matrices, total signed compound matrices such as acyclic matrices, Cauchy matrices, totally positive matrices, diagonally scaled totally unimodular matrices, and matrices arising in certain simple finite element problems. Diagonally dominant matrices have also been shown to have accurate rank-revealing decomposition; see [1, 10, 34]. Specifically, in [34, Algorithm 1], a variation of the Gaussian elimination is developed to compute an accurate *LDU* factorization that is shown to be an accurate rank-revealing decomposition. The computational cost of this accurate *LDU* algorithm is about the same as the standard Gaussian elimination. Since discretizations of differential equations are often close to being diagonally dominant, we can construct a diagonally dominant preconditioner, for which the accurate *LDU* factorization provides an inverse-accurate algorithm. This will be used in our numerical examples in §5.

We remark that if two matrices A_1 and A_2 both have accurate rank-revealing decomposition, then solving $A_1A_2x = b$ through $A_1y = b$ and $A_2x = y$ will produce an inverse-equivalent solution provided $\|A_1^{-1}\|\|A_2^{-1}\|/\|(A_1A_2)^{-1}\|$ is a modest number; see [36]. In particular, we may also consider a preconditioner that is a product of diagonally dominant matrices; see Examples 2 and 4 in §5.

4 Application to Eigenvalue Problems

In this section we discuss an application of inverse-equivalent algorithms to computing a few smallest eigenvalues (in absolute value) of a matrix through accurate inverses.

In general, the relative accuracy of the computed smallest eigenvalue of a matrix in finite precision depends on the condition number $\kappa_2(A)$. To illustrate, we consider

an $n \times n$ symmetric positive definite matrix A. Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be its eigenvalues. A backward stable algorithm computes an approximate eigenvalue-eigenvector pair $(\widehat{\lambda}_i, \widehat{x}_i)$ with $\|\widehat{x}_i\| = 1$ such that the residual $\|A\widehat{x}_i - \widehat{\lambda}_i\widehat{x}_i\|$ is of order $\mathbf{u}\|A\|$. Then, $|\widehat{\lambda}_i - \lambda_i| \leq \mathcal{O}(\mathbf{u})\|A\|$ and hence

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{\lambda_i} \leq \mathcal{O}(\mathbf{u})\frac{\lambda_n}{\lambda_i} \tag{26}$$

It follows that *larger eigenvalues* (i.e. $\lambda_i \approx \lambda_n$) are computed to the accuracy of machine precision, but for *smaller eigenvalue* (i.e. $\lambda_i \approx \lambda_1$), a relative error of order $\mathcal{O}(\mathbf{u})\kappa(A)$ is expected.

Since the larger eigenvalues can be computed accurately, to compute a few smallest eigenvalues of an ill-conditioned matrix, we may compute correspondingly a few largest eigenvalues of A^{-1}. However, a difficulty with this approach is that, A^{-1}, or its multiplications on vectors, can not be computed accurately since A is assumed to be ill-conditioned. For diagonally dominant matrices, this can be remedied by using the accurate LDU factorizations [34, 35, 36]. Specifically, for large scale problems, we apply in [36] the Lanczos method to A^{-1} or simply use the inverse iteration and compute its largest eigenvalue $\mu_1 = \lambda_1^{-1}$. At each iteration, the accurate LDU factorizations is used to compute the matrix-vector product $A^{-1}v$ (i.e. solving $Au = v$), which produces a solution that would be equivalent to the one produced by multiplying the exact A^{-1} with v. Hence the resulting residual error will be of order $\mathbf{u}\|A^{-1}\|_2 = \mathbf{u}\mu_1$, which implies a relative error for μ_1 in the order of machine precision. Finally $\lambda_1 = \mu_1^{-1}$ is computed accurately.

Now, consider a general symmetric matrix A that can be preconditioned by a diagonally dominant matrix. Then using the accurate preconditioning scheme of §3, we can form $A^{-1}v$ (i.e. solving $Au = v$) with the inverse-equivalent accuracy. Then in the same way as discussed above, a few largest eigenvalues in absolute values can be computed accurately for A^{-1}, from which a few smallest eigenvalues in absolute values for A are computed accurately.

The same discussion can also be extended to nonsymmetric matrices with the modification of the bound (26) as

$$\frac{|\widehat{\lambda}_i - \lambda_i|}{|\lambda_i|} \approx \mathcal{O}(\mathbf{u})\frac{1}{c_i}\frac{\|A\|}{|\lambda_i|}$$

where c_i is the cosine of the angle between the left and the right eigenvectors of A corresponding to λ_i; see [11, Theorem 4.4]. The additional factor $1/c_i$ defines the sensitivity caused by nonnormality of the matrix, which is also studied through pseudospectra (see [31]). This factor is not changed with the inverse. Thus, when $A^{-1}v$ is computed with inverse-equivalent accuracy, a few largest eigenvalues of A^{-1}, and then the corresponding eigenvalues of A, are computed with an accuracy

independent of the condition number $\kappa(A)$. However, the accuracy is still expected to depend on $1/c_i$. See [9, Theorem 6.3] for some some related discussions for diagonally dominant matrices.

Finally, we discuss an application to discretizations of differential operators, which is a large source of ill-conditioned problems. For the discretization of differential eigenvalue problems, it is usually a few smallest eigenvalues that are of interest but their computed accuracy is reduced by the condition number of the discretization matrix. For operators involving high order differentiations such as biharmonic operators, the matrix may easily become extremely ill-conditioned and then little accuracy may be expected of the computed eigenvalues; see [6, 7, 36] and §5.

In [36], we have used the accurate inverse approach to accurately compute a few smallest eigenvalues of a differential operator whose discretization matrix is diagonally dominant. For differential operators whose discretizations are not diagonally dominant, they can often be preconditioned by a diagonally dominant matrix. For example, consider the finite difference discretization of the convection-diffusion operator

$$-\triangle u + \beta u_x + \gamma u_y = \lambda u \quad \text{on} \quad (0,1)^2;$$

with the homogeneous Dirichlet boundary condition. The discretization matrix A is not diagonally dominant, but the discretization of the diffusion operator $-\triangle$ is. The convection operator is dominated by the diffusion operator, if β, γ are not too large. Then, the discretization matrix can be well preconditioned by that of the diffusion operator. Hence, using accurate preconditioning, we can accurately compute a few smallest eigenvalues of the convection-diffusion operator. See Examples 3 in §5 for some numerical results.

The convection-diffusion operator is just one example of differential operators whose discretization may be preconditioned by a diagonally dominant matrix. It will be interesting to study other differential operators with such properties but we leave it to a future work.

5 Numerical Examples

In this section, we present four numerical examples to demonstrate performance of the accurate preconditioning scheme. All tests were carried out on a PC in MATLAB (R2016b) with a machine precision $\mathbf{u} \approx$ `2e-16`. The first two examples concern solving linear systems and the last two examples use two similar matrices with known exact eigenvalues for the eigenvalue problems.

We consider linear systems arising in finite difference discretizations of some differential equations scaled so that the resulting matrix has integer entries. We construct an integer solution x so that $b = Ax$ can be computed exactly. Then x

is the exact solution. In our testing, we are interested in systems with a random b, as this resembles practical situations where b is usually the input data. By (10), a random b is also likely to yield a system where an inverse-equivalent accurate solution is significantly more accurate than a backward stable solution. To construct a random integer vector b with integer solution x, we first construct a random vector $\mathtt{b_0} = \mathtt{rand(n,1)}$ and set $x_0 = \mathtt{A\backslash b_0}$, from which we construct a scaled integer solution $x = \mathtt{round(x_0 * 1e8/norm(x_0, inf))}$, all in MATLAB functions. Then $b = Ax$ is computed exactly and is approximately a scaled random vector b_0.

We solve the systems by a preconditioned iterative method with the preconditioner solved by the usual Cholesky factorization and by the accurate LDU factorization ([34, Algorithm 1]). In all examples here, the preconditioners are symmetric; so we actually compute the LDL^T factorization which has about half of the cost. We compare the computed solutions $\widehat{x}$ with respect to the errors

$$\eta_{ie} := \frac{\|\widehat{x} - x\|_2}{\|A^{-1}\|_2\|b\|_2} \quad \text{and} \quad \eta_{rel} := \frac{\|\widehat{x} - x\|_2}{\|x\|_2}.$$

η_{ie} measures the inverse-equivalent accuracy and η_{rel} is the relative accuracy. They differ by a fixed ratio $\frac{\|x\|_2}{\|A^{-1}\|_2\|b\|_2} \leq 1$. For a backward stable solution $\widehat{x}$, η_{rel} is approximately $\kappa_2(A)\mathbf{u}$ but for an inverse-equivalent accurate solution, η_{ie} is in the order of machine precision $\mathbf{u}$.

Example 1. Consider the 1-dimensional convection-diffusion equation

$$-u''(x) - u'(x) = f(x) \quad \text{on} \quad (0, \gamma);$$

with the Dirichlet boundary condition $u(0) = u(\gamma) = 0$. Discretizing on a uniform grid of size $h = \gamma/(n+1)$ by the center difference scheme, we obtain $A_n = \frac{1}{h^2}T_n - \frac{1}{2h}K_n$, where T_n is the $n \times n$ tridiagonal matrix with diagonals being 2 and off-diagonals being -1, and K_n is the skew-symmetric $n \times n$ tridiagonal matrix with 1 on the superdiagonal above the main diagonal. To construct b and the exact solution $x = A_n^{-1}b$, we scale A_n by $2\gamma^2/(n+1)$ and use an integer value for γ so that the resulting matrix $2(n+1)T_n - \gamma K_n$ has integer entries. We then construct a random integer vector b and the corresponding exact solution x as discussed at the beginning of this section.

T_n is diagonally dominant and has an accurate LDL^T factorization. A_n is neither symmetric nor diagonally dominant but, if γ is not too large, preconditioning by $2(n+1)T_n$ yields a well-conditioned matrix $B = I - \frac{h}{2}T_n^{-1}K_n$. We solve the preconditioned system by the GMRES method with the preconditioning equations solved in two ways: 1. using the Cholesky factorization of T_n, and 2. the accurate LDU factorization of T_n. The GMRES is implemented with restart after 50 iterations and the stopping tolerance for relative residual is set as $\sqrt{n}\mathbf{u}$. As a reference, we also solve the original system using MATLAB's division operator $\mathtt{A_n\backslash b}$. We compare the computed solutions $\widehat{x}$ by the three methods with respect to η_{ie} and η_{rel}.

In Table 1, we present the results for a mildly ill-conditioned case with $n = 2^{13} - 1 = 8,191$ and a more ill-conditioned case with $n = 2^{19} - 1 = 524,287$. For each case of n, we test $\gamma = 10^1, 10^2, \cdots, 10^6$, resulting in an A_n that is increasingly not symmetric and not diagonally dominant. In the table, in addition to the errors η_{ie} and η_{rel}, we also present the condition numbers $\kappa_2(A_n)$ and, for the smaller n case, $\kappa_2(B)$ as well. In the columns for accurate LDU preconditioning, we also list $\rho := \|\gamma K_n\|_1 \|x\|_1 / \|b\|_1$ which is a factor in the error bound for η_{ie} by accurate preconditioning; see Theorem 4.

Table 1: Example 1: Accuracy for the three methods (`A\b`, Cholesky Preconditioning, Accurate LDU Preconditioning): $\eta_{ie} = \|\widehat{x} - x\|_2 / (\|A^{-1}\|_2 \|b\|_2)$ and $\eta_{rel} = \|\widehat{x} - x\|_2 / \|x\|_2$, and $\rho = \|\gamma K_n\|_1 \|x\|_1 / \|b\|_1$.

		`A\b`		Cholesky Precond.			Accurate Precond.		
γ	$\kappa_2(A_n)$	η_{ie}	η_{rel}	η_{ie}	η_{rel}	$\kappa_2(B)$	η_{ie}	η_{rel}	ρ
$n = 2^{13} - 1 = 8,191$									
1e1	1e7	3e-12	4e-12	2e-12	3e-12	8e0	3e-15	4e-15	3e3
1e2	2e6	3e-11	4e-11	3e-13	3e-13	2e2	4e-15	5e-15	4e3
1e3	2e5	3e-12	4e-12	3e-14	4e-14	7e3	7e-15	9e-15	4e3
1e4	2e4	2e-17	3e-17	8e-15	1e-14	1e5	7e-15	9e-15	4e3
1e5	5e3	5e-16	6e-16	2e-14	3e-14	1e6	2e-14	3e-14	4e3
1e6	5e3	1e-15	2e-15	6e-13	8e-13	4e6	6e-13	8e-13	4e3
$n = 2^{19} - 1 = 524,287$									
1e1	6e10	4e-10	5e-8	1e-9	2e-7	-	2e-16	2e-14	2e3
1e2	7e9	7e-9	1e-7	8e-10	2e-8	-	8e-15	2e-13	2e4
1e3	7e8	8e-9	2e-8	7e-10	2e-9	-	1e-13	4e-13	1e5
1e4	7e7	2e-9	2e-9	1e-10	2e-10	-	5e-14	6e-14	3e5
1e5	7e6	6e-11	8e-11	1e-11	2e-11	-	6e-14	8e-14	3e5
1e6	7e5	1e-18	2e-18	1e-12	2e-12	-	6e-14	8e-14	3e5

We observe that in all cases, the accurate preconditioning produces an inverse-equivalent accuracy η_{ie} roughly in the order of machine precision, regardless of the condition number $\kappa_2(A_n)$. Taking into consideration the results of Example 2 below, η_{ie} appears to be proportional to $(1 + \rho)\mathbf{u}$ as indicated by the theory. For the first case where $\kappa_2(B)$ is computed, η_{ie} increases slightly with $\kappa_2(B)$ but this effect appears to emerges only when $\kappa_2(B) \geq 10^5$. With η_{ie} in the order of machine precision, the relative error η_{rel} is improved accordingly, which, in this case, is near the machine precision. In contrast, the solutions by `A\b` and by the Cholesky preconditioning have relative errors η_{rel} of order $\kappa_2(A)\mathbf{u}$ as expected, which determines a corresponding η_{ie}. With larger γ, A_n becomes less ill-conditioned and the accuracy attained by `A\b` increases. When $\gamma \geq 10^4$ (the first n case) or $\gamma = 10^6$ (the second n case), it becomes more accurate than the one by the accurate preconditioning, but

since $\kappa_2(B)$ is larger than $\kappa_2(A_n)$ in those cases, the preconditioning is obviously not expected to be effective.

The results demonstrate that when $\kappa_2(B)$ is not very large, the accurate preconditioning indeed produces inverse-equivalent accuracy while the preconditioning solved by a backward stable algorithm does not improve the solution accuracy at all.

Example 2. Let $A_n = (n+1)^4 T_n^2 + \gamma S_n$, where T_n is as in Example 1, S_n is a random sparse integer matrix constructed using $\mathtt{S_n} = \mathtt{floor}(10 * \mathtt{sprandn}(\mathtt{n}, \mathtt{n}, 0.001))$ in MATLAB and γ is an integer parameter. Note that $(n+1)^4 T_n^2$ is a finite difference discretization of 1-dimensional biharmonic operator $\frac{d^4u}{dx^4}$ with the boundary condition $u = \frac{d^2u}{dx^2} = 0$ on a uniform mesh on $[0, 1]$ with the meshsize $1/(n+1)$. For an integer value of γ, A_n is an integer matrix and we construct a random integer vector b and the corresponding exact solution x as discussed at the beginning of this section.

If $|\gamma|$ is not too large, preconditioning with $(n+1)^4 T_n^2$ results in a well-conditioned matrix $B = I + \frac{\gamma}{(n+1)^4} T_n^{-2} S_n$. We solve the preconditioned system by GMRES with two way of solving the preconditioning equations: 1. using the Cholesky factorization of T_n^2, and 2. using the accurate LDU factorization of T_n. The GMRES is implemented with restart after 50 iterations and the stopping tolerance for relative residual is set as $\sqrt{n}\mathbf{u}$. Again, we also solve the original system using MATLAB's division operator $\mathtt{A}\backslash\mathtt{b}$. We compare the computed solutions $\widehat{x}$ by the three methods with respect to η_{ie} and η_{rel}.

In Table 2, we present the testing results for $n = 2^{10} - 1 = 1,023$ and $n = 2^{14} - 1 = 16,383$. For these two cases respectively, S_n has 1,008 and 257,572 nonzeros with $\|S_n\|_\infty = 75$ and 343. For each of the n value, we test $\gamma = 10, -10^2, 10^3, -10^4, 10^5, -10^6, 10^7$. We list in the table $\kappa_2(A_n)$ and $\kappa_2(B)$ and $\rho := \|\gamma S_n\|_1 \|x\|_1 / \|b\|_1$, in addition to η_{ie}, η_{rel}.

We observe that the accurate preconditioning produces an inverse-equivalent accuracy η_{ie} in the order of machine precision, except when $|\gamma|$ is very large. Comparing with Example 1, η_{ie} is about 3 order of magnitude smaller and this seems to be due to a corresponding decrease in $1 + \rho$. As $|\gamma|$ increases, the quality of preconditioning deteriorates. However, its effect on η_{ie} emerges only when $\kappa_2(B) \geq 10^5$. From that point on, η_{ie} appears proportional to $\kappa_2(B)(1+\rho)\mathbf{u}$ as indicated by our theory. Overall, similar behavior as in Example 1 is observed for this random sparse matrix.

The results of these two examples are in agreement with our error analysis (Theorem 4). The inverse-equivalent accuracy error η_{ie} appears proportional to $\kappa_2(B)(1+\rho)\mathbf{u}$ although its dependence on $\kappa_2(B)$ may appear only when $\kappa_2(B)$ is quite large. Indeed, its capability to produce an inverse-equivalent accuracy with large $\kappa_2(B)$ is rather surprising. This would allow a broader application of the accurate preconditioning method than what our theory might suggest.

Table 2: Example 2: Accuracy for the three methods (`A\b`, Cholesky Preconditioning, Accurate *LDU* Preconditioning): $\eta_{ie} = \|\widehat{x} - x\|_2/(\|A^{-1}\|_2\|b\|_2)$, $\eta_{rel} = \|\widehat{x} - x\|_2/\|x\|_2$, and $\rho = \|\gamma S_n\|_1\|x\|_1/\|b\|_1$.

		`A\b`		Cholesky Precond.			Accurate Precond.		
γ	$\kappa_2(A)$	η_{ie}	η_{rel}	η_{ie}	η_{rel}	$\kappa_2(B)$	η_{ie}	η_{rel}	ρ
$n = 2^{10} - 1 = 1,023$									
1e1	2e11	3e-10	1e-7	4e-10	1e-7	3e0	5e-18	2e-15	2e-2
-1e2	2e11	8e-10	3e-7	4e-10	1e-7	9e1	6e-18	2e-15	2e-1
1e3	3e11	5e-11	3e-8	4e-10	2e-7	2e4	4e-18	2e-15	2e0
-1e4	4e10	2e-10	2e-8	2e-10	2e-8	2e5	6e-16	6e-14	1e1
1e5	9e9	2e-10	6e-9	2e-10	5e-9	5e6	7e-14	2e-12	1e2
-1e6	4e9	8e-11	1e-9	9e-11	1e-9	2e8	2e-11	3e-10	1e3
1e7	1e8	6e-13	2e-11	1e-10	4e-9	6e8	2e-11	6e-10	1e2
$n = 2^{14} - 1 = 16,383$									
1e1	3e16	8e-10	5e-2	1e-9	7e-2	5e1	6e-19	3e-11	1e-6
-1e2	2e15	1e-10	4e-4	1e-9	3e-3	2e2	1e-18	3e-12	1e-5
1e3	6e14	2e-10	3e-4	6e-10	8e-4	9e3	6e-19	9e-13	1e-4
-1e4	5e14	5e-11	7e-5	3e-10	5e-4	8e5	5e-19	8e-13	9e-4
1e5	8e13	1e-10	3e-5	3e-10	7e-5	1e7	9e-17	2e-11	9e-3
-1e6	4e13	9e-11	1e-5	1e-10	2e-5	5e8	2e-15	2e-10	9e-2
1e7	9e12	3e-11	1e-6	1e-10	3e-6	1e10	8e-14	2e-9	9e-1

In the next two examples, we compute the smallest eigenvalue (in absolute value) of A accurately by computing the corresponding largest eigenvalue of A^{-1}. We have used both the Lanczos algorithm with full reorthogonalization and the power method (i.e. the inverse iteration for A) and found the results to be similar. Below, we report the results obtained by the inverse iteration only. In applying A^{-1} at each step of iteration, we solve $Au = v$ by a preconditioned iterative method. We test solving the preconditioner by the usual Cholesky factorization or by the accurate *LDU* factorization ([34, Algorithm 1]). With the two ways of solving the preconditioning equations, we compare the final approximate eigenvalues obtained.

Example 3. Consider the eigenvalue problem for the same 1-dimensional convection-diffusion operator as in Example 1: $-u''(x) - u'(x) = \lambda u(x)$ on $(0, \gamma)$ with $u(0) = u(\gamma) = 0$. The eigenvalues of this operator are exactly known [28, Theorem 1]:

$$\lambda_i = \frac{1}{4} + \frac{\pi^2 i^2}{\gamma^2}, \quad \text{for} \quad i = 1, 2, \cdots$$

Discretizing on a mesh of size $h = \gamma/(n+1)$ as in Example 1, we obtain the same matrix $A_n = \frac{1}{h^2}T_n - \frac{1}{2h}K_n$.

We approximate $\lambda_1 = \frac{1}{4} + \frac{\pi^2}{\gamma^2}$ by computing the smallest eigenvalue of A_n using

the inverse iteration. At each iteration, we solve $A_n u = v$ by the GMRES method as preconditioned by $\frac{1}{h^2}T_n$ with two ways of solving the preconditioner T_n: 1. using the Cholesky factorization of T_n, and 2. the accurate LDU factorization of T_n. We denote the computed smallest eigenvalues by μ_1^{chol} and μ_1^{aldu} respectively. The GMRES is implemented with restart after 50 iterations and the stopping tolerance for relative residual is set at $\sqrt{n}\mathbf{u}$. The stopping tolerance for the eigenvalue-eigenvector residuals of the inverse iteration is also set at $\sqrt{n}\mathbf{u}$. We use this very stringent criterion to ensure as accurate results as possible. In all cases, the inverse iteration terminates with the residual satisfying the criterion.

In Table 3, we present the testing results for $h = 2^{-6}, 2^{-8}, \cdots, 2^{-24}$ and $\gamma = 1$. We list the computed eigenvalues μ_1^{chol} and μ_1^{aldu} and their relative errors. We observe that μ_1^{chol} initially converge quadratically as h. However, as h decreases, the matrix becomes increasingly ill-conditioned and the roundoff errors associated with the standard Cholesky preconditioning increase and will dominate the discretization errors at some point (see [36]). In this example, this occurs at $h \approx 1.5e-5$, after which further decreasing h actually increases the error for μ_1^{chol}. On the other hand, the error for μ_1^{aldu} decreases quadratically to the order of machine precision. Thus, the accurate preconditioning allows us to compute the smallest eigenvalue of the convection-diffusion operator, whose discretization is nonsymmetric and not diagonally dominant, to the full accuracy of the discretization, up to the machine precision.

Table 3: Example 3: approximation of $\lambda_1 = \frac{1}{4} + \pi^2 = 10.11960440108936$ (μ_1^{chol} - computed eigenvalue by Cholesky preconditioner; μ_1^{aldu} - computed eigenvalue by accurate LDU preconditioner.)

h	μ_1^{chol}	$\frac{\lvert\lambda_1-\mu_1^{chol}\rvert}{\lambda_1}$	μ_1^{aldu}	$\frac{\lvert\lambda_1-\mu_1^{aldu}\rvert}{\lambda_1}$
1.6e-2	10.11732544149765	2.3e-4	10.11732544149762	2.3e-4
3.9e-3	10.11946195350748	1.4e-5	10.11946195350759	1.4e-5
9.8e-4	10.11959549807000	8.8e-7	10.11959549806623	8.8e-7
2.4e-4	10.11960384467139	5.5e-8	10.11960384465017	5.5e-8
6.1e-5	10.11960436740018	3.3e-9	10.11960436631197	3.4e-9
1.5e-5	10.11960440025146	8.3e-11	10.11960439891543	2.1e-10
3.8e-6	10.11960357476229	8.2e-8	10.11960440095356	1.3e-11
9.5e-7	10.11959786966499	6.5e-7	10.11960440107954	9.7e-13
2.4e-7	10.11960179526253	2.6e-7	10.11960440108836	9.9e-14
6.0e-8	10.11996930306172	3.6e-5	10.11960440108905	3.0e-14

Example 4: Consider computing the smallest eigenvalue of the 1-dimensional biharmonic problem: $\frac{d^4v}{dx^4} + \rho v = \lambda v$ on $[0, 1]$ with the natural boundary condition $v(0) = v''(0) = v(1) = v''(1) = 0$. Discretizing on a uniform mesh of size $h =$

$1/(n+1)$ leads to $A_n = \frac{1}{h^4}T_n^2 + \rho I$. where T_n is the discretization of 1-dimensional Laplacian defined in Example 1. The eigenvalues of A_n are $\lambda_{j,h} = \frac{1}{h^4}16\sin^4(j\pi h/2) + \rho$ (see [11, Lemma 6.1]). We consider $n = 2^{16} - 1 = 65,535$ for this example and $\rho = \pm 1, \pm 10, \pm 10^2, \pm 10^3$. This results in an extremely ill-conditioned A_n with $\kappa_2(A_n) \approx 10^{18}$ except in the case of $\rho = -10^2$ when $\kappa_2(A_n) \approx 10^{20}$. A_n also becomes indefinite when $\rho = -10^2$ or -10^3.

We compute the smallest eigenvalue in absolute value, denoted by λ_{absmin}, of A_n by applying the inverse iteration to A_n. Note that this eigenvalue may not be $\lambda_{1,h}$ if A_n is indefinite. In carrying out the inverse iterations, we solve $A_n x = b$ by the CG (or MINRES if $\gamma < 0$) method as preconditioned by $\frac{1}{h^4}T_n^2$ with two way of solving the preconditioner T_n^2: 1. using the Cholesky factorization of T_n^2, and 2. using accurate LDU factorization of T_n. We denote the computed smallest eigenvalues in absolute value by μ_1^{chol} and μ_1^{aldu} respectively. The stopping tolerance for relative residual of CG or MINRES is set at $\sqrt{n}\mathbf{u}$. The stopping tolerance for the eigenvalue-eigenvector residuals of the inverse iteration is also set at $\sqrt{n}\mathbf{u}$. In our tests, the inverse iteration with the accurate LDU factorization preconditioning produces a residual satisfying the stopping criterion in all cases. The one with the Cholesky factorization preconditioning, however, results in stagnating residuals mostly around 10^{-11} that is slightly above the threshold. The latter can be attributed to the inaccuracy in the operator A_n^{-1}.

In Table 4, we present, for each case of ρ, the exact eigenvalue λ_{absmin}, the computed eigenvalues μ_1^{chol} and μ_1^{aldu} and their relative errors. For all the cases of ρ here, the preconditioned matrix $B = I + \rho h^2 T_n^{-2}$ is well conditioned with $\kappa(B)$ raging between 1 and 40. As a result, the accurate preconditioning produces μ_1^{aldu} that is accurate to the machine precision in all cases. The eigenvalues computed using the preconditioning with the Cholesky factorization μ_1^{chol} have in most cases one digit of accuracy. In the case $\rho = -10^2$, it has even the sign wrong. Again we see that the accurate preconditioning accurately computes the smallest eigenvalue of this extremely ill-conditioned matrix, even when the matrix is indefinite.

6 Concluding Remarks

We have presented an accurate preconditioning method to solve linear systems with inverse-equivalent accuracy. An error analysis is developed to demonstrate the accuracy that may be achieved by this approach. Numerical examples confirm the analysis but also show that the method works even when the quality of preconditioner is rather low. As an application, we use it to accurately compute the smallest eigenvalue of some differential operator discretizations that are indefinite or non-symmetric.

For future works, it will be interesting to study a related perturbation theory

Table 4: Example 4: approximation of the smallest eigenvalue in absolute value λ_{absmin} (μ_1^{chol} and μ_1^{aldu} - computed eigenvalue by Cholesky preconditioning and by accurate LDU preconditioner respectively. $e^{chol} := \frac{|\lambda_{\text{absmin}} - \mu_1^{chol}|}{|\lambda_{\text{absmin}}|}$; $e^{aldu} := \frac{|\lambda_{\text{absmin}} - \mu_1^{aldu}|}{|\lambda_{\text{absmin}}|}$)

ρ	λ_{absmin}	μ_1^{chol}	e^{chol}	μ_1^{aldu}	e^{aldu}
1e0	98.409090996696	107.104718485058	9e-2	98.409090996693	3e-14
-1e0	96.409090996696	105.104718492797	9e-2	96.409090996693	3e-14
1e1	107.409090996696	116.104718499209	8e-2	107.409090996693	3e-14
-1e1	87.409090996696	96.104718508722	1e-1	87.409090996693	3e-14
1e2	197.409090996696	206.104718499414	4e-2	197.409090996691	3e-14
-1e2	-2.590909003304	6.104718530850	3e0	-2.590909003309	2e-12
1e3	1097.40909099669	1106.10471864325	8e-3	1097.40909099668	1e-14
-1e3	558.545454156402	716.309982554411	3e-1	558.545454156396	1e-14

and to investigate what appears to be a very mild dependence of the accuracy on the condition number of the preconditioned matrix. It will also be interesting to study whether our method can be used with preconditioners that are defined through their inverses, such as multilevel preconditioners [33] and sparse approximate inverse preconditioners [4, 5].

Acknowledgement: I would like to thank Prof. Jinchao Xu for some interesting discussions on multilevel preconditioners that have inspired this work. I would also like to thank Kasey Bray for many helpful comments on a draft of this paper.

References

[1] 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.

[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe and H. van der Vorst, editors, *Templates for the solution of Algebraic Eigenvalue Problems: A Practical Guide.* SIAM, Philadelphia, 2000.

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

[4] M. Benzi, J. K. Cullum, and M. Tuma. Robust approximate inverse preconditioning for the conjugate gradient method. *SIAM J. Sci. Comp.*, 22:1318–1332, 2000.

[5] M. Benzi and M. Tuma. A robust incomplete factorization preconditioner for positive definite matrices. *Num. Lin. Alg. Appl.*, 10:385–400, 2003.

[6] 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.

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

[8] T. Chan and D. Foulser, *Effectively Well-Conditioned Linear Systems,* SIAM J. Sci. Stat. Comput, 9 (1988), 963–969.

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

[10] 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.

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

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

[13] 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.

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

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

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

[17] 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.

[18] 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.

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

[20] 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.

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

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

[23] A. Greenbaum, *Estimating the attainable accuracy of recursively computed residual methods*, SIAM J. Matrix Anal. Appl. 18. pp. 535-551 (1997)

[24] M. Gutknecht, *Lanczos-type solvers for nonsymmetric linear systems of equations*, Acta Numerica 6, pp. 271-397 (1997)

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

[26] N.J. Higham, *Accuracy and Stability of Numerical Algorithms,* SIAM, Philadelphia, PA, 2002.

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

[28] S. Reddy and L. Trefethen, *Pseudospectra of the convection-diffusion operator,* SIAM J. APPL. MATH., 54(1994):1634-1649.

[29] G. Sleijpen and H. van der Vorst, *Reliable updated residuals in hybrid Bi-CG methods*, Computing 56., pp. 144-163 (1996)

[30] G. L. G. Sleijpen, H. A. van der Vorst and D. R. Fokkema, *BICGSTAB(ℓ) and other hybrid Bi-CG methods*, Numerical Algorithms, 7, pp. 75–109 (1994)

[31] L. N. Trefethen and M. Embree *Spectra and Pseudospectra: The Behavior of Nonnormal Matrices and Operators,* Princeton University Press, 2005.

[32] H. van der Vorst and Q. Ye, *Residual Replacement Strategies for Krylov Subspace Iterative Methods for the Convergence of True Residuals,* SIAM J. Sci. Comp. 22 (2000):836-852.

[33] J. Xu, *Iterative methods by space decomposition and subspace correction,* SIAM Rev. 34 (1992):581-613.

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

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

[36] Q. Ye. *Accurate inverses for computing eigenvalues of extremely ill-conditioned matrices and differential operators*, Math. Comp., to appear.