SIAM J. MATRIX ANAL. APPL.
Vol. 24, No. 3, pp. 787–801

A SUBSPACE ERROR ESTIMATE FOR LINEAR SYSTEMS*

YANG CAO† AND LINDA PETZOLD†

Abstract. This paper proposes a new method for estimating the error in the solution of linear systems. A condition number is defined for a linear function of the solution components. This definition of the condition number is quite versatile. It reduces to the component condition number proposed by Chandrasekaran and Ipsen [*SIAM J. Matrix Anal. Appl.*, 16 (1995), pp. 93–112] and to Skeel's definition of condition number [*J. ACM*, 26 (1979), pp. 494–526] in some special cases, and it can be used to estimate the error in a subspace. The estimate is based on the adjoint equation in combination with small sample statistical theory. It can be implemented simply and is inexpensive to compute. Numerical examples are presented which illustrate the power and effectiveness of this error estimate.

Key words. condition number, adjoint method, linear system, subspace error estimate

AMS subject classifications. 65F35, 15A12

PII. S0895479801390649

1. Conditioning and error estimation for linear systems.

Perturbation theory for linear systems has been studied for many years. The basic question is, How sensitive is the solution to perturbations in the data? First order analysis is often used in estimating errors; for instance, for stability analysis of algorithms or for the condition number of an eigenvalue.

Consider the linear system

$$Ax = b, \tag{1.1}$$

where $A \in R^{n\times n}$. The basic question of perturbation theory is, How much will x change if A and b are perturbed? Suppose we are solving a perturbed linear system $(A+\Delta A)\tilde{x} = b+\Delta b$. We would like to estimate the relative error $\|x-\tilde{x}\|/\|x\|$. Here we skip the details of which norm we are using and what kind of perturbation we are assuming. Traditionally the relative error is estimated using the condition number $K(A) = \|A\|\|A^{-1}\|$ and the backward error. The following results are well known [6, p. 133].

If $\frac{\|\Delta A\|}{\|A\|} < \mu$, $\frac{\|\Delta b\|}{\|b\|} < \mu$, and $\mu K(A) < 1$, then

$$\frac{\|x-\tilde{x}\|}{\|x\|} \le \frac{2\mu K(A)}{1-\mu K(A)}. \tag{1.2}$$

Here we take μ to be a multiple of the relative machine precision ϵ_{mach}. The error estimate can be given in terms of the residual $r = A\tilde{x} - b$ by

$$\frac{\|x-\tilde{x}\|}{\|x\|} \le \frac{K(A)\|r\|}{\|A\|\|x\|}. \tag{1.3}$$

*Received by the editors January 9, 2002; accepted for publication (in revised form) by I. C. F. Ipsen August 27, 2002; published electronically January 31, 2003. This work was supported by grants NSF/ITR ACI-0086061, NSF/KDI ATM-9873133, DOE DE-F603-00ER25430, and LLNL ISCR 00-15.

http://www.siam.org/journals/simax/24-3/39064.html

†Department of Computer Science, University of California Santa Barbara, Santa Barbara, CA 93106 (ycao@cs.ucsb.edu, petzold@engineering.ucsb.edu).

When the condition number $K(A)$ is very large, the system is considered to be ill-conditioned and the solution may not be accurate. We call $K(A)$ the standard condition number in the following.

Many examples have demonstrated that the standard condition number may lead to an overly pessimistic estimate for the overall error and that it may underestimate the relative error for some components. Consider the following problems.

Example 1. Let

$$A = \begin{bmatrix} 1 & 0 \\ 0 & \delta \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ \delta \end{bmatrix},$$

where δ is very small. The solution is $\binom{1}{1}$ The condition number is $\frac{1}{\delta}$. Although for small ϵ the condition number is very large, the solution is accurate. In fact, the solution always has a high relative accuracy for any right-hand side b (assuming a relative perturbation in A and b).

Example 2. Let

$$A = \begin{bmatrix} 1 & 1+\delta \\ 1-\delta & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1+\delta+\delta^2 \\ 1 \end{bmatrix},$$

where δ is a small parameter. Choosing $\delta = 10^{-5}$, the estimate (1.2) will not produce a warning in Matlab [9]. However, the true value of x_2 is 10^{-5} and the result computed by Matlab is 8.8818×10^{-6}, which has relative error of 0.112. There is not even one digit of accuracy! On the other hand, when $\delta = 10^{-4}$, the computed result is 1.000888×10^{-4}, with a relative error of 8.89×10^{-5}. The computed result has four digits of accuracy. The discrepancy can be explained using sensitivity analysis of individual solution components [2].

Example 3. The numerical solution [12] of certain high-index differential-algebraic equations (DAEs) by a fully implicit method yields an ill-conditioned system of linear equations to be solved at each time step. But the propagation of error to future time steps depends only on a well-conditioned subspace. Consider the following simple index-2 DAE system:

$$\left\{ \begin{array}{rcl} \dot{x}_1 & = & x_3 + 1, \\ \dot{x}_2 & = & x_3 + 2, \\ 0 & = & x_1 + x_2 - 1. \end{array} \right. \tag{1.4}$$

Discretization by the backward Euler method yields a linear system with the matrix

$$A = \begin{bmatrix} 1 & 0 & -h \\ 0 & 1 & -h \\ 1 & 1 & 0 \end{bmatrix}. \tag{1.5}$$

The stepsize h at each time step may be very small. The condition number of A is $O(\frac{1}{h})$ [12, p. 144]. Thus the linear system can be very poorly conditioned for small stepsizes. However, the propagation of error to future time steps for this DAE depends only on errors in the lower-index variables x_1 and x_2. Thus, it is much more critical to get an accurate solution for these variables than for the higher-index variable x_3. In computation, we find that the linear system is solved quite accurately (using Gaussian elimination (GE) with partial pivoting) for x_1 and x_2, and it is only the variable x_3 that is affected by the ill-conditioning. The standard condition number cannot distinguish between the error in the two subspaces.

Many other definitions of condition number have been proposed. References [6], [7], [14], and [16] give some historical review. A precise analysis was given by Skeel [15], leading to a componentwise definition of the condition of the linear system,

$$\operatorname{cond}(A, x) = \frac{\| \, |A^{-1}||A||x| \, \|_\infty}{\|x\|_\infty}, \tag{1.6}$$

where $|A| = \{|a_{ij}|\}$, and the condition number of A,

$$\operatorname{cond}(A) = \||A^{-1}||A|\|_\infty. \tag{1.7}$$

This definition applies to componentwise relative perturbations. It can deal with Example 1 easily and leads to a well-conditioned matrix A for that example. This definition is also a special case of the componentwise analysis described in [6, p. 135]. Unfortunately, the cost to compute A^{-1} is large. In practice, the 1-norm of A^{-1} is estimated [6, p. 290]. In [2], the concept of a componentwise condition number, which yields a condition number for each component of the solution x, was proposed. Thus, for Example 2 we can compute the condition number for x_2 directly and obtain a better error estimate. Example 3 could be handled by computing the component condition number, but for larger DAE systems this could become awkward and expensive.

In this paper we will define a condition number that is applicable in even more general situations. From our experience with solving DAE systems and optimal control problems, we believe that whether or not a solution is acceptable depends on the requirements of the problem. In Example 2, if we are concerned only with the accuracy of component x_2, then the solution is unacceptable. The normwise condition number of the vector x cannot discern this. In Example 3, since we are mainly concerned with the accuracy of x_1 and x_2 but not of x_3, the solution is acceptable although the standard condition number may be very high. This suggests for us to define a condition number that can vary with different requirements. We will use the concept of "derived function" introduced in section 2 to derive such a condition number.

To estimate the condition number, it is not necessary to compute A^{-1} exactly. Typically, one only wants to know the condition number within a factor of 10. Condition estimators with $O(n^2)$ cost, based on the use of random vectors, have been proposed in a number of papers [1], [3], [5], [9], [10], [11]. A detailed review can be found in [6, Chap. 14]. Generally, these estimators yield poorer estimates than the standard condition number but are cheaper to compute. In this paper we will also propose a method that makes use of random vectors to perform error estimation. Our method makes use of the idea and analysis for small sample statistical estimate in [8] although we will not estimate A^{-1} directly. In [4], the complexity of computing error bounds for linear systems is analyzed. The analysis reveals that $O(n^2)$ condition estimators cannot be free of counterexamples. In particular, our $O(n^2)$ condition estimator has low-probability counterexamples which arise from some choices of the random vectors.

The main contribution of this paper has two parts. First, we define a condition number that resolves many of the problems with the standard condition number. It reduces to the component condition number in some special cases and it can be used to estimate the error in a subspace. A subspace condition number is proposed, which helps to separate a well-conditioned subspace from an ill-conditioned system. Second, we provide a means, using small sample statistical theory, of accurately and efficiently computing this condition.

This paper is organized as follows. In section 2 we introduce the concept of an error estimate for a derived function. In section 3 we present our condition estimator for general derived functions and apply it to some examples. In section 4, numerical results are presented which compare this definition with the standard condition number, Skeel's definition [15], and Matlab's condition estimator (which uses Higham's modification of Hager's method [6]). The numerical tests are based on randomly generated dense or banded matrices and right-hand side vectors.

2. Estimating the error of a derived function. Given the linear system (1.1), a derived function is a function $g(x)$ of the solution. We are concerned with the relative error in the derived function, $\|g(x) - g(\tilde{x})\|/\|g(x)\|$. But since the true solution x can only be approximated by the numerical solution $\tilde{x}$, it is more practical to compute $\|g(x) - g(\tilde{x})\|/\|g(\tilde{x})\|$. Before we begin our discussion, we need to specify the norm and what kind of perturbation we are concerned with. In the following, if we do not state a particular norm, the vector norm can be any monotone norm, which satisfies the requirement that if $|x| \leq |y|$, then $\|x\| \leq \|y\|$. For example, the p-norm and the ∞-norm meet this requirement. The matrix norm takes the operator norm.

Generally speaking, we cannot talk about errors without specifying some assumption about the corresponding numerical methods or perturbations. A bad numerical method will result in a large backward error even for a well-conditioned system. For example, it is well known that Cramer's rule gives a large backward error [6, p. 15]. GE without pivoting may lead to a large growth of perturbations for general matrices as well. In this paper we do not want to dig into the details of the numerical methods. Instead we will make some simple but reasonable assumptions about the size of the perturbations. There are two major types of assumption: normwise and componentwise. Normwise analysis assumes $\|\Delta A\| \leq \epsilon\|E\|$ and $\|\Delta b\| \leq \epsilon\|f\|$, while componentwise analysis assumes $|\Delta A| \leq \epsilon E$ and $|\Delta b| \leq \epsilon f$, where E and f are assumed to have nonnegative entries. Different choices of E and f result in different error bounds. As stated in [6, p. 134], the most common choice of tolerance is $E = |A|$ and $f = |b|$. This choice is satisfied by QR factorization [6, p. 369], where $|\Delta A| \leq f(n)\epsilon G|A|$ and $|\Delta b| \leq f(n)\epsilon G|b|$. For LU factorization, $E = |L||U|$ should be used [6, p. 175]. Some special classes of matrices have LU factorization with $|L||U| = |A|$ or $|L||U| \leq 3|A|$ [6, p. 184]. In this paper, we will present a componentwise analysis by taking $E = |A|$ and $f = |b|$.

Different derived functions lead to different condition numbers. When we choose $g(x) = x$ we will obtain the traditional condition number. When we choose $g(x) = x_i$ we will obtain the component condition number. The derived function reflects the requirements of the application. For example, in the application of condition estimate for the linear system generated in a DAE solver, as in Example 3, the derived function is defined via the projection of the solution onto the space of the lower index variables. Thus we will refer to the corresponding error estimate as a *subspace error estimate.* Usually we define the derived function as a linear function of the solution x. Of course we could define a nonlinear derived function, but so far in our applications we have needed only the linear one. Thus we will write the derived function as $g(x) = Lx$, where $L : R^n \longrightarrow R^k$ is a linear function. We assume $\text{rank}(L) = k$.

Consider the perturbed linear system

$$(A + \Delta A)\tilde{x} = b + \Delta b, \tag{2.1}$$

where $|\Delta A| < \epsilon|A|$, $|\Delta b| < \epsilon|b|$. We have

$$A(x - \tilde{x}) = \Delta A\tilde{x} - \Delta b,$$

hence

$$x - \tilde{x} = A^{-1}(\Delta A\tilde{x} - \Delta b), \tag{2.2}$$

and

$$g(x) - g(\tilde{x}) = LA^{-1}(\Delta A\tilde{x} - \Delta b). \tag{2.3}$$

Thus we have the estimate

$$\begin{aligned} \frac{\|g(x) - g(\tilde{x})\|}{\|g(\tilde{x})\|} &\leq \frac{\||LA^{-1}|(|\Delta A||\tilde{x}| + |\Delta b|)\|}{\|L\tilde{x}\|} \\ &\leq \epsilon \frac{\||LA^{-1}|(|A||\tilde{x}| + |b|)\|}{\|L\tilde{x}\|}, \end{aligned} \tag{2.4}$$

and we obtain the condition number

$$\mathrm{cond}_L(A, \tilde{x}) = \frac{\||LA^{-1}|(|A||\tilde{x}| + |b|)\|}{\|L\tilde{x}\|}. \tag{2.5}$$

Supposing that $|b| \leq |A||x|$, and assuming that x is closely approximated by $\tilde{x}$, yields

$$\mathrm{cond}_L(A, \tilde{x}) \leq \frac{2\||LA^{-1}||A||\tilde{x}|\|}{\|L\tilde{x}\|}. \tag{2.6}$$

When we take $g(x) = x$, L is the identity operator, and this definition reduces to the condition number introduced by Skeel [15]. When we take $g(x) = x_i$, this definition reduces to the component condition number defined in [2]. Thus the relative error in the derived function is bounded by

$$\frac{\|g(x) - g(\tilde{x})\|}{\|g(\tilde{x})\|} \leq \mathrm{cond}_L(A, \tilde{x})\epsilon. \tag{2.7}$$

It is easy to generalize the properties of the standard condition number using this definition. We remind the reader that (2.6) and (2.7) are approximate in that they are based on the assumption that x is closely approximated by $\tilde{x}$.

For Example 3, we have

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{and} \quad A^{-1} = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ -\frac{1}{2h} & -\frac{1}{2h} & \frac{1}{2h} \end{bmatrix}.$$

Using (2.6), we have

$$\mathrm{cond}_L(A, x) \leq \frac{\sqrt{(|x_1| + |hx_3|)^2 + (|x_2| + |hx_3|)^2 + (|x_1| + |x_2|)^2}}{\sqrt{x_1^2 + x_2^2}}$$

in the 2-norm. The subspace condition number is $O(1)$ even in the case of inconsistent initial conditions for the index-2 DAE (the index-2 variable x_3 can be $O(\frac{1}{h})$ in this case because it is approximating an impulse). This corresponds well with DAE theory [12, p. 144].

3. Condition estimate. Just changing the definition of the condition number doesn't give us much benefit since in practice we may not be able to afford to compute A^{-1} or LA^{-1}. The natural question is, How can we efficiently compute this condition number? We will first give a method based on a scalar derived function, $Lx = l^T x$, where $l \in R^n$, and then extend the estimate for the case of a vector derived function.

3.1. Scalar derived function. For a scalar derived function $g(x) = l^T x$, we can efficiently compute the condition number by first computing the adjoint variable λ which solves

$$A^T \lambda = l \tag{3.1}$$

so that $\lambda^T = l^T A^{-1}$. Assuming that we have the LU or QR decomposition of A, this equation can be solved in $O(n^2)$ cost. Then the condition number becomes

$$\mathrm{cond}_l(A, x) = \frac{|\lambda^T|(|A||x| + |b|)}{|l^T x|}. \tag{3.2}$$

It is the condition number in a particular direction, so we will call it a *directional condition number*. When the direction is toward a single component, this becomes the component condition number.

3.2. Vector derived function. A direct extension of the above defined error estimate to the case of a vector derived function can be quite expensive to compute. Thus we will *estimate* a measure of the vector error by making use of a scalar derived function. To accomplish that, we introduce the small-sample statistical method for estimating the 2-norm (details can be found in [8]). In the following, the norm is the 2-norm.

For any vector $l \in R^n$, if z is selected uniformly and randomly from the unit sphere S_{n-1} in n dimensions, the expected value of $|l^T z|$ is given by

$$E(|l^T z|) = \|l\| E_n,$$

where $E_1 = 1$, $E_2 = \frac{2}{\pi}$, and for $n > 2$,

$$E_n = \frac{1 \cdot 3 \cdot 5 \cdots (n-2)}{2 \cdot 4 \cdot 6 \cdots (n-1)} \quad \text{for } n \text{ odd},$$

$$E_n = \frac{2}{\pi} \cdot \frac{2 \cdot 4 \cdot 6 \cdots (n-2)}{1 \cdot 3 \cdot 5 \cdots (n-1)} \quad \text{for } n \text{ even}.$$

E_n can be estimated by $\sqrt{\frac{2}{\pi(n-\frac{1}{2})}}$. Thus we use $\xi = \frac{|l^T z|}{E_n}$ to estimate $\|l\|$. The estimate satisfies

$$Pr\left(\frac{\|l\|}{w} \leq \xi \leq w\|l\|\right) \geq 1 - \frac{2}{\pi w} + O\left(\frac{1}{w^2}\right),$$

where $Pr()$ denotes the probability, and $w > 0$ is a real number. The bound does not depend on the vector l. In condition number estimation, usually we are interested in finding an estimate that is accurate to a factor of 10 ($w = 10$).

For a more accurate estimate, we can use more orthogonal random vectors. Suppose we have k orthogonal random vectors $z_1, z_2, \ldots, z_k$. Let

$$\xi_i = \frac{|l^T z_i|}{E_n}.$$

Then the estimate for $\|l\|$ is given by

$$\xi(k) = E_k\sqrt{\xi_1^2 + \cdots + \xi_k^2}. \tag{3.3}$$

Usually, at most two or three random vectors are required in practice. The corresponding probabilities satisfy [8]

$$Pr\left(\frac{\|l\|}{w} \le \xi(2) \le w\|l\|\right) \approx 1 - \frac{\pi}{4w^2},$$

$$Pr\left(\frac{\|l\|}{w} \le \xi(3) \le w\|l\|\right) \approx 1 - \frac{32}{3\pi^2 w^3}.$$

We will use this tool to construct a subspace error estimate for the linear system. To estimate $\frac{\|L(x-\tilde{x})\|}{\|L\tilde{x}\|}$, where L is a linear function from R^n to R^k, we select a vector z uniformly and randomly from the unit sphere S_{k-1}. Let $g_z(x) = z^T Lx$. Then $|g_z(x) - g_z(\tilde{x})| = |z^T L(x-\tilde{x})|$. Defining $K_1 = \frac{|z^T L(x-\tilde{x})|}{E_k\|L\tilde{x}\|}$, we have

$$Pr\left(\frac{1}{w}\frac{\|L(x-\tilde{x})\|}{\|L\tilde{x}\|} \le K_1 \le w\frac{\|L(x-\tilde{x})\|}{\|L\tilde{x}\|}\right) \approx 1 - \frac{2}{\pi w}.$$

Taking λ to solve the adjoint equation

$$A^T\lambda = L^T z, \tag{3.4}$$

we have from (2.3),

$$|z^T L(x-\tilde{x})| \le \epsilon|\lambda|^T(|A||\tilde{x}| + |b|).$$

We define

$$e_1 = \frac{|\lambda|^T(|A||\tilde{x}| + |b|)}{E_k\|L\tilde{x}\|},$$

where λ solves (3.4). We have $K_1 \le e_1\epsilon$. The condition estimate is given by e_1. The relative error is estimated by $e_1\epsilon$. When $L = I$, this differs from the traditional relative error bound by a factor of E_k. Note that K_1 approximates the relative error with a high probability, and $e_1\epsilon$ is an upper bound for K_1. Thus $e_1\epsilon$ is usually larger than the relative error.

Numerical experiments show that this estimate, using one random vector, gives a good result for most cases. But for some random vectors, it may produce a large error. In this situation, using more random orthogonal vectors improves the result. To keep the computational cost low, we use at most two or three random orthogonal vectors. Given orthogonal vectors $z_i \in R^k$, define

$$K_2 = \frac{E_2\sqrt{(z_1^T L(x-\tilde{x}))^2 + (z_2^T L(x-\tilde{x}))^2}}{E_k\|L\tilde{x}\|},$$

$$K_3 = \frac{E_3\sqrt{(z_1^T L(x-\tilde{x}))^2 + (z_2^T L(x-\tilde{x}))^2 + (z_3^T L(x-\tilde{x}))^2}}{E_k\|L\tilde{x}\|}.$$

Then

$$Pr\left(\frac{1}{w}\frac{\|L(x-\tilde{x})\|}{\|L\tilde{x}\|} \le K_2 \le w\frac{\|L(x-\tilde{x})\|}{\|L\tilde{x}\|}\right) \approx 1 - \frac{\pi}{4w^2},$$

$$Pr\left(\frac{1}{w}\frac{\|L(x-\tilde{x})\|}{\|L\tilde{x}\|} \le K_3 \le w\frac{\|L(x-\tilde{x})\|}{\|L\tilde{x}\|}\right) \approx 1 - \frac{32}{3\pi^2 w^3}.$$

For a condition estimate, we usually require the magnitude of the estimate to be within a ratio of 10. Letting $w = 10$, the probability of an acceptable estimate for K_1 is 93.6%, while for K_2 it is 99.2% and for K_3 it is 99.9%.

Let λ_i solve

$$A^T \lambda_i = L^T z_i.$$

Defining

$$v_i = |\lambda_i|^T (|A||\tilde{x}| + |b|),$$

we obtain

$$e_2 = \frac{E_2\sqrt{(v_1^2 + v_2^2)}}{E_k\|L\tilde{x}\|} \tag{3.5}$$

and

$$e_3 = \frac{E_3\sqrt{(v_1^2 + v_2^2 + v_3^2)}}{E_k\|L\tilde{x}\|}. \tag{3.6}$$

Thus $K_2 \le e_2\epsilon$, $K_3 \le e_3\epsilon$. e_1, e_2, and e_3 are the corresponding condition estimates.

This method is especially useful for obtaining a subspace condition estimate. Let L be a projection from R^n to R^k. The above method gives a relative error estimate for the subspace of the solution under the projection.

To summarize, the algorithm for the subspace error estimate is given as follows. We suggest using three random vectors for the estimate.

Subspace Error Estimate Algorithm. *Suppose we have an LU or QR decomposition of A and the numerical solution $\tilde{x}$. The condition number is estimated as follows:*

Step 1. *Determine the subspace or the components for which one wants to estimate the error. Let k be the dimension of the subspace and L be the projection from R^n to the subspace.*

Step 2. *Randomly choose three orthogonal vectors z_1, z_2, z_3 from the unit sphere S_{k-1}. Solve* (3.4) *for the corresponding λ_1, λ_2, λ_3.*

Step 3. *Compute*

$$v_i = |\lambda_i^T|(|A||\tilde{x}| + |b|).$$

Then the subspace condition estimate is given by

$$e_3 = \frac{E_3\sqrt{(v_1^2 + v_2^2 + v_3^2)}}{E_k\|L\tilde{x}\|}, \tag{3.7}$$

and the subspace relative error estimate is given by $e_3\epsilon$.

3.3. Examples. Here we demonstrate how the proposed method resolves the problems in Examples 1–3 of section 1.

Example 4.

$$A = \begin{bmatrix} 1 & 0 \\ 0 & \delta \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}.$$

The solution is $\tilde{x} = (b_1, b_2/\delta)$. Recall that, for this example, the solution has high relative accuracy for any right-hand side, assuming a relative perturbation. To compute the error estimate, let the random vector be z, where $\|z\|_2 = 1$. Solving the adjoint equation (3.4) yields $\lambda = (z_1, z_2/\delta)^T$. Then the relative error is estimated using e_1 by

$$\frac{2(|b_1||z_1| + |b_2||z_2|/|\delta|)}{E_2\sqrt{b_1^2 + b_2^2/\delta^2}}\epsilon \leq \frac{2\epsilon}{E_2}.$$

Regardless of the random vector chosen, this method always yields a small condition number. Of course, e_2 will yield the exact condition number since the problem has just two dimensions and we choose orthogonal random vectors.

Example 5.

$$A = \begin{bmatrix} 1 & 1+\delta \\ 1-\delta & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1+\delta+\delta^2 \\ 1 \end{bmatrix}.$$

Suppose our goal is an accurate x_2. Then we let $g(x) = x_2$. Since $g(x)$ is a scalar function, we do not need a random vector here. Solving the adjoint equation (3.1), we have $\lambda = \frac{1}{\delta^2}[-(1-\delta), 1]^T \approx \frac{1}{\delta^2}[-1, 1]^T$. The relative error in x_2 is estimated by

$$\frac{|\lambda^T|(|A||\tilde{x}| + |b|)\epsilon}{|\tilde{x}_2|} \approx \frac{4}{|\delta|^3}\epsilon.$$

With a good numerical method like GE with partial pivoting (GEPP) or QR, ϵ is just a multiple of the relative machine precision ϵ_{mach} for this two-dimensional problem. For Matlab we get $\epsilon_{mach} \approx 10^{-16}$. We can see from our estimate that when $\epsilon = 10^{-5}$, the solution for x_2 will have a relative error of 0.1 (the computed result yields an error of 0.112). When $\epsilon = 10^{-4}$, the estimate predicts four digits of accuracy in x_2. Thus, the estimate accurately predicts the results obtained by Matlab (described in section 1), while the standard condition number underestimates the error.

Example 6.

$$A = \begin{bmatrix} 1 & 0 & -h \\ 0 & 1 & -h \\ 1 & 1 & 0 \end{bmatrix}, \quad b = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}.$$

For the subspace condition number, we choose a random vector $z = [r_1,\ r_2,\ 0]^T$ of norm 1. Solving the adjoint equation (3.4) yields $\lambda = [\frac{1}{2}(r_1 - r_2),\ \frac{1}{2}(r_2 - r_1),\ \frac{1}{2}(r_1 + r_2)]^T$. The condition is estimated using one random vector and (2.6) to be

$$\frac{|\lambda_1|(|x_1|+|hx_3|+|x_1-hx_3|)+|\lambda_2|(|x_2|+|hx_3|+|x_2-hx_3|)+|\lambda_3|(|x_1|+|x_2|+|x_1+x_2|)}{E_3\sqrt{x_1^2+x_2^2}}$$
$$\leq \frac{(|\lambda_1|+|\lambda_2|+|\lambda_3|)(|x_1|+|x_2|+|hx_3|)}{E_3\sqrt{x_1^2+x_2^2}}.$$

Thus the condition estimated is $O(1)$, as we would expect from DAE theory [12, p. 144]) for the condition of the low-index subspace.

4. Numerical results. The numerical experiments were performed in Matlab on a Linux computer. We chopped the data for a round-off error of 10^{-8} to avoid any possibility that differences in the conclusions could be caused by different machine precisions.

We compare our error estimate with Skeel's condition estimate (1.6), the standard condition number, and the condition estimate provided by Matlab for randomly generated data. We first generate the random matrix A. Then a real x is generated randomly, and b is determined by $b = Ax$. We chop the data of A and b to get a relative error of 10^{-8}. Then we solve $A\tilde{x} = b$ for $\tilde{x}$. We compare the estimates and the actual relative error $\frac{\|x-\tilde{x}\|_2}{\|\tilde{x}\|_2}$. Skeel's condition number and the standard condition number have been computed accurately without approximation of $|A^{-1}|$. For the relative error of x, our definition reduces to Skeel's definition. But the statistical estimate used in our method is different from the estimate used in the suggested implementation of Skeel's method. Our estimate uses the small sample statistical method and the adjoint equation to estimate $\||A^{-1}|(|A\tilde{x}| + |b|)\|$ for the whole space, or $\||LA^{-1}|(|A\tilde{x}| + |b|)\|$ for some subspace, using several random orthogonal vectors, while the suggested implementation of Skeel's method approximates the matrix $|A^{-1}|$ directly [6, section 14.5]. The latter is much more complicated and expensive and is limited to matrices of a particular structure. For the three orthogonal random vectors on the unit sphere, we first generate three random vectors r_1, r_2, r_3 uniformly in $R^k([-1,1]) = \{x \in R^k | x_i \in [-1,1]\}$ and then make them orthogonal by setting

$$z_1 = \frac{r_1}{\|r_1\|}, \quad z_2 = \frac{r_2 - z_1^T r_2 z_1}{\|r_2 - z_1^T r_2 z_1\|}, \quad z_3 = \frac{r_3 - z_1^T r_3 z_1 - z_2^T r_3 z_2}{\|r_3 - z_1^T r_3 z_1 - z_2^T r_3 z_2\|}.$$

Note that although this is not exactly uniform on the unit sphere, it is cheaper to generate, and from our practice we feel it works quite well.

4.1. Scalar function g. Our first numerical test is to estimate the relative error for a scalar function. Here we let $g(x) = \frac{1}{n}\sum_{i=1}^{n} x_i$. Since $g(x)$ is a scalar function, we can use the directional condition estimate. Other definitions do not provide a good estimate because they have not been designed to deal with this type of derived function. The corresponding results are shown in Figures 4.1 and 4.2 and Table 4.1. We show both the overestimate ratio $\frac{estimate}{real\ error}$ and the underestimate ratio $\frac{real\ error}{estimate}$. It can be seen that the standard condition definition and Skeel's definition result in a much greater overestimate than our method.

4.2. Vector error estimate. Next we compared the relative error $\frac{\|x-\tilde{x}\|}{\|\tilde{x}\|}$ with the estimates for 10,000 randomly generated dense matrices A and vectors x of dimension 100. The results are shown in Figures 4.3 and 4.4. The underestimates and overestimates for our method are displayed in Figure 4.3. Figure 4.4 shows the overestimate ratio for Skeel's definition, the standard condition number, and Matlab's estimator. Table 4.2 compares the mean and max value of those ratios for each method. From the results, we can see that there is a potential for a substantial overestimate for all the definitions and estimators. Our estimator is, with high probability, within a factor of 10 of the standard condition estimate, as shown in Figure 4.5. If we take an overestimate larger than 100 as a bad estimate, in 10,000 random tests, our method generates 142 bad estimates (1.42%), Skeel's condition number generates 195 bad estimates (1.95%), the standard condition number generates 405 bad estimates (4.05%), and the Matlab estimator generates 2124 bad estimates (21.24%).

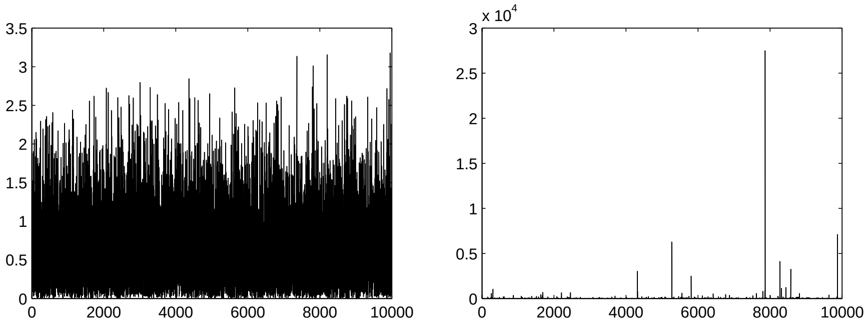

FIG. 4.1. *The plot on the left shows $\frac{real\ error}{our\ estimate}$, the amount by which our method underestimates the error of the mean function $g(x)$, for* 10,000 *randomly generated dense matrices A and vectors x of dimension* 100. *The plot on the right shows the amount of overestimate $\frac{our\ estimate}{real\ error}$.*

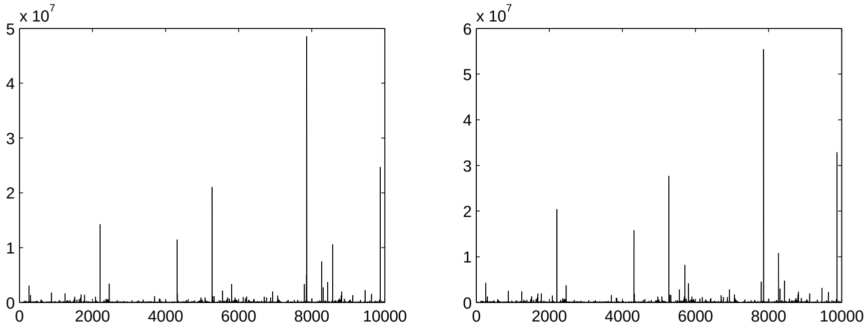

FIG. 4.2. *Overestimate of error of the mean function by Skeel's definition (left) and by the standard condition estimate (right) for* 10,000 *randomly generated dense matrices A and vectors x of dimension* 100. *(Note that, since Skeel's definition and the standard condition estimate are not designed for the computation of the condition of a scalar derived function, for these definitions we are using the estimate of the full vector.)*

TABLE 4.1

Comparison of ratios of overestimate and underestimate of error of the mean function using different condition estimates for dense matrices. For our method, the maximum of the overestimate and the underestimate is shown.

	Our method	Skeel	Standard	Matlab
MEAN	12.48	3.58×10^4	4.65×10^4	1.16×10^5
MAX	2.75×10^4	4.85×10^7	5.54×10^7	1.22×10^8

4.3. Ill-conditioned matrices. Another group of experiments was done for the (ill-conditioned) Hilbert matrix of dimension 10, where $a_{ij} = \frac{1}{i+j}$. The results are shown in Figures 4.6 and 4.7 for 10,000 randomly generated vectors x. Here we can see that all the methods can give a substantial overestimate to the actual error. Our method yields a result which is comparable to Skeel's estimate and to the standard condition estimate. For the number of overestimates by a factor of more than 100, in 10,000 random tests our method generated 259 (2.59%), Skeel's condition number generated 628 (6.28%), the standard condition number generated 4056 (40.56%), and Matlab's estimator generated 7394 (73.94%).

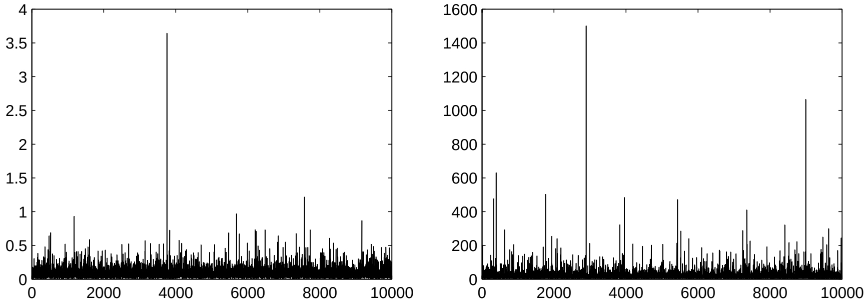

Fig. 4.3. *Underestimate of vector error* $\frac{real\ error}{our\ estimate}$ *(left) and overestimate of vector error* $\frac{our\ estimate}{real\ error}$ *(right) by our method for* 10,000 *randomly generated dense matrices A and vectors x of dimension* 100*.*

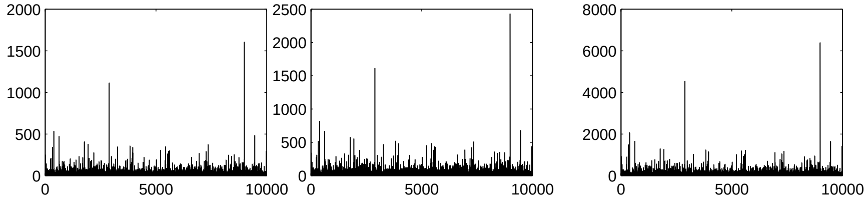

Fig. 4.4. *Overestimate of vector error by Skeel's condition estimate (left), by the standard condition estimate (middle), and by Matlab's condition estimate (right) for* 10,000 *randomly generated dense matrices A and vectors x of dimension* 100*.*

Table 4.2
Comparison of ratios of overestimate and underestimate of vector error using different condition estimates for dense matrices.

	Our method	Skeel	Standard	Matlab
MEAN	21	25	33	83
MAX	1500	1604	2432	6389

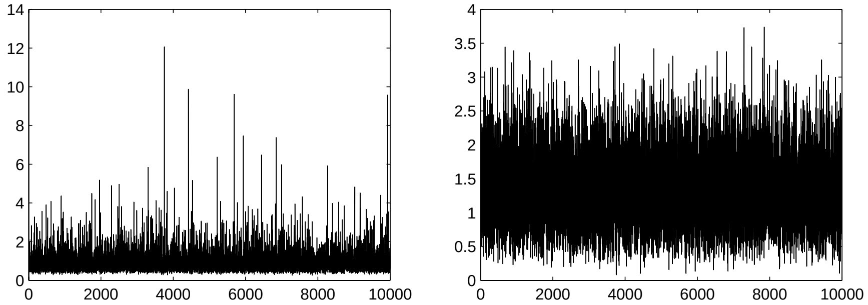

Fig. 4.5. *Underestimate of standard condition number* $\frac{standard\ condition\ number}{our\ estimate}$ *(left) and overestimate of standard condition number* $\frac{our\ estimate}{standard\ condition\ number}$ *(right) by our method for* 10,000 *randomly generated dense matrices A and vectors x of dimension* 100*.*

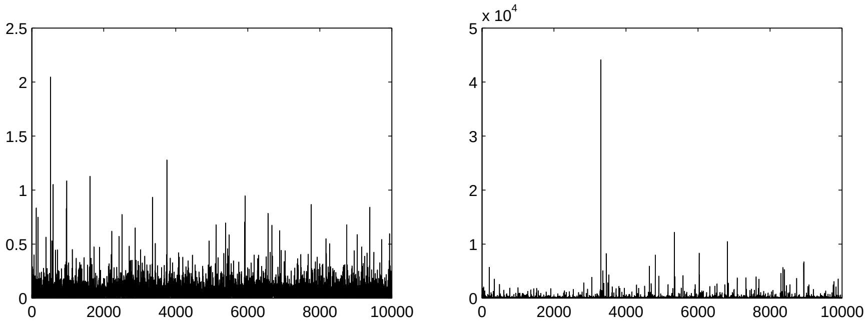

FIG. 4.6. *Underestimate (left) and overestimate (right) of the vector error by our method for the Hilbert matrix of dimension* 10 *with* 10,000 *randomly generated vectors x.*

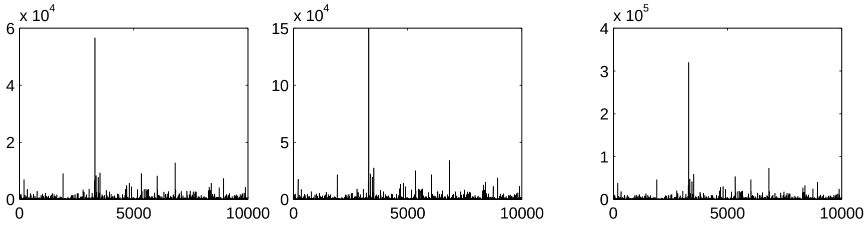

FIG. 4.7. *Overestimate of the vector error by Skeel's condition estimate (left), the standard condition estimate (middle), and Matlab's condition estimate (right) for the Hilbert matrix of dimension* 10 *with* 10,000 *randomly generated vectors x.*

TABLE 4.3
Comparison of condition numbers for Example 3 *in section* 1.

Stepsize h	Ours (full space)	Skeel's	Standard	Ours (subspace)
10^{-6}	2.04×10^6	1.41×10^6	1.5×10^6	3.30
10^{-8}	2.13×10^8	1.41×10^8	1.5×10^8	3.30
10^{-12}	2.40×10^{12}	1.41×10^{12}	1.5×10^{12}	3.30

4.4. DAE examples. We take Example 3 in section 1 as our first DAE example. We choose different stepsizes $h = 10^{-6}$, 10^{-8}, 10^{-12} and random right-hand sides b. The corresponding condition numbers are listed in Table 4.3. With the stepsize decreasing, the condition number for the full solution space grows as $O(\frac{1}{h})$ for all these definitions. But for the subspace of only the first two components, the subspace condition number remains at 3.30. This indicates that this subspace is well-conditioned, although the system is ill-conditioned in the full solution space.

Another DAE example comes from an application in mechanics. It is of interest for the computation of the elliptic Fekete points [13]. The problem is of the form

$$M\frac{dy}{dt} = f(y(t)), \quad y(0) = y_0, \quad y'(0) = y'_0, \tag{4.1}$$

with $y, f \in R^{2N}$ and $0 \leq t \leq t_{end}$. Here, $t_{end} = 1000$, $N = 20$, and M is the mass matrix given by

$$M = \begin{pmatrix} I_{6N} & 0 \\ 0 & 0 \end{pmatrix},$$

Table 4.4
Comparison of condition numbers for the Fekete problem.

Stepsize h	Ours (full space)	Skeel's	Standard	Ours (subspace)
10^{-6}	7.07×10^6	1.00×10^6	2.5×10^{11}	24.37
10^{-8}	5.93×10^8	1.00×10^8	3.44×10^{15}	24.37
10^{-12}	1.15×10^9	1.00×10^{12}	1.42×10^{27}	24.37

where I_{6N} is the identity matrix of dimension $6N$. The details of this problem can be found in [13] and also on the website http://hilbert.dm.uniba.it/∼testset/descrip.htm. Since we are concerned only with the linear system generated in the solution process, we extract the linear system for different stepsizes $h = 10^{-6}$, 10^{-8}, 10^{-12} and randomly generate the right-hand sides b. The subspace with the first 120 components is what we are concerned with here. The numerical results are shown in Table 4.4. The condition number of the full solution space grows when the stepsize decreases, while the condition number for the subspace remains the same at 24.37. This subspace condition number shows that the solution to the linear system can be computed safely for the first 120 components.

5. Conclusion. In this paper we proposed a new definition of condition number and a new method for error and condition estimation based on the adjoint equation and the small-sample statistical method. This new definition can produce a subspace error estimate, which is useful in some applications. For a vector measure of the error, the new definition, estimated as outlined by the small-sample statistical method, has low ($3n^2$) cost (assuming direct solution of dense linear systems where the matrix has already been factorized) and probability of 99.9% for the accuracy of the error estimate to be within a factor of 10. The method easily allows for the use of different derived functions (measures of the error) that may be relevant for different problems.

Acknowledgments. The authors would like to thank Charles Kenney, Shiv Chandrasekaran, and the referees for their insightful comments.

REFERENCES

[1] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whitney, *ScaLAPACK Users' Guide*, SIAM, Philadelphia, 1997.

[2] S. Chandrasekaran and I. C. F. Ipsen, *On the sensitivity of solution components in linear systems of equations*, SIAM J. Matrix Anal. Appl., 16 (1995), pp. 93–112.

[3] A. K. Cline, C. B. Moler, G. W. Stewart, and J. H. Wilkinson, *An estimate for the condition number of a matrix*, SIAM J. Numer. Anal., 16 (1979), pp. 368–375.

[4] J. Demmel, B. Diament, and G. Malajovich, *On the complexity of computing error bounds*, Found. Comput. Math., 1 (2001), pp. 101–125.

[5] J. D. Dixon, *Estimating extremal eigenvalues and condition numbers of matrices*, SIAM J. Numer. Anal., 20 (1983), pp. 812–814.

[6] N. J. Higham, *Accuracy and Stability of Numerical Algorithms,* 2nd ed., SIAM, Philadelphia, 2002.

[7] W. Kahan, *Numerical linear algebra*, Canad. Math. Bull., 9 (1966), pp. 757–801.

[8] C. S. Kenney and A. J. Laub, *Small-sample statistical condition estimates for general matrix functions*, SIAM J. Sci. Comput., 15 (1994), pp. 36–61.

[9] C. S. Kenney, A. J. Laub, and M. S. Reese, *Statistical condition estimation for linear systems*, SIAM J. Sci. Comput., 19 (1998), pp. 566–583.

[10] J. Kuczyński and H. Woźniakowski, *Estimating the largest eigenvalue by the power and Lanczos algorithms with a random start*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 1094–1122.

[11] J. Kuczyński and H. Woźniakowski, *Probabilistic bounds on the extremal eigenvalues and condition number by the Lanczos algorithm*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 672–691.

[12] K. E. Brenan, S. L. Campbell, and L. R. Petzold, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, Classics in Applied Mathematics 14, SIAM, Philadelphia, 1996.

[13] P. M. Pardalos, *An open global optimization problem on the unit sphere*, J. Global Optim., 6 (1995), p. 213.

[14] J. R. Rice, *A theory of condition*, SIAM J. Numer Anal., 3 (1966), pp. 187–310.

[15] R. D. Skeel, *Scaling for numerical stability in Gaussian elimination*, J. ACM, 26 (1979), pp. 494–526.

[16] J. H. Wilkinson, *Rounding Errors in Algebraic Processes*, Prentice-Hall, Englewood Cliffs, NJ, 1963.