

Experience with a Matrix Norm Estimator

Higham, Nicholas J.

1990

MIMS EPrint: **2006.160**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: <http://eprints.maths.manchester.ac.uk/>

And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

EXPERIENCE WITH A MATRIX NORM ESTIMATOR*

NICHOLAS J. HIGHAM†

Abstract. Fortran 77 codes for estimating the 1-norm of a real or complex matrix were presented by Higham in [*ACM Trans. Math. Software*, 14 (1988), pp. 381–396]. The codes have found use in various applications and have been adopted by two program libraries. Further observations about the norm estimation algorithm and experience in using it are reported here. In particular, an example is given where the algorithm requires nearly the maximum possible number of iterations.

Key words. matrix 1-norm, condition number estimation

AMS(MOS) subject classifications. primary 65F05, 65F35

1. Introduction. In [9] Higham presented two Fortran 77 codes for estimating the 1-norm of a real matrix or a complex matrix. The codes are in the NAG library at Mark 13 [13, Chap. F04], are being used for all the condition number estimation in LAPACK [3], and have been used with research or production codes by several workers [1], [2], [5], [6], [12]. Because of this wide use of the routines we feel it is worth reporting experience and insights accrued since the work in [9] was done. We describe here a new example of slow convergence of the norm estimation algorithm, point out that it is a special case of a more general iteration for estimating matrix norms, summarize the practical performance of the estimator, and describe several interesting applications.

First, we recall the original algorithm of Hager [7], on which the estimators in [9] are based. Our notation is as follows: e is the vector of all ones, e_j is the j th column of the identity matrix, and $\xi = \text{sign}(y)$ means $\xi_i = 1$ or -1 according as $y_i \geq 0$ or $y_i < 0$.

ALGORITHM 1. Given $B \in \mathbb{R}^{n \times n}$ this algorithm computes γ and $y = Bx$ such that $\gamma \leq \|B\|_1$ with $\|y\|_1/\|x\|_1 = \gamma$.

```
x = e/n
repeat
  y = Bx
  ξ = sign(y)
  z = BTξ
  if ||z||∞ ≤ zTx
    γ = ||y||1
    quit
  end
  x = ej, where |zj| = ||z||∞ (smallest such j)
end
```

An important feature of the algorithm is that it requires only a means for evaluating matrix-vector products Bx and $B^T\xi$ —it does not require explicit access to the matrix B . Of course, if B were available explicitly we could simply compute $\|B\|_1 = \max_j \sum_i |b_{ij}|$.

* Received by the editors October 19, 1989; accepted for publication (in revised form) December 5, 1989.

† Department of Mathematics, University of Manchester, Manchester, M13 9PL, United Kingdom (na.nhigham@na-net.stanford.edu).

To derive Algorithm 1, Hager regards $\|B\|_1$ as the maximum of $F(x) = \|Bx\|_1$ over the unit ball S in the 1-norm and, since F is nondifferentiable at points x where Bx has a zero component, he applies an optimization technique based on subgradients. A subgradient of F at x is any vector g such that

$$F(y) \geq F(x) + g^T(y - x) \quad \text{for all } y \in S.$$

See [7]–[9] for further details of the subgradient-based derivation. An alternative derivation relying on a simple heuristic argument is given in [9].

Hager notes that Algorithm 1 requires at most $n + 1$ iterations, since $F(x) = \|y\|_1 = z^T x$ increases strictly on each stage and so each of the vertices e_j is visited at most once. He proves that the final vector x is a local maximizer for F as long as $y = Bx$ has no zero components.

We noticed recently that Algorithm 1 is a special case of an iteration investigated in [14] for estimating the mixed subordinate norm

$$(1) \quad \|B\|_{\alpha, \beta} = \max_{x \neq 0} \frac{\|Bx\|_{\beta}}{\|x\|_{\alpha}}, \quad B \in \mathbb{R}^{m \times n},$$

where $\|\cdot\|_{\alpha}$ and $\|\cdot\|_{\beta}$ are arbitrary vector norms. The iteration is

$$(2) \quad x^{k+1} = h(z^k) = h(B^T g(Bx^k)),$$

where g and h are subgradients of the β -norm and the norm dual to the α -norm, respectively. With $\alpha = \beta = 1$ this iteration reduces to the one in Algorithm 1. In [14] it is proved that iteration (2) converges in a finite number of steps if one of the α - and β -norms is polyhedral (this class includes the 1- and ∞ -norms, but not the 2-norm), but no properties of the limit point are determined. In [15] the special case where $\alpha = \infty$, $\beta = 1$, and B is symmetric positive definite is examined in detail. Unfortunately, the results in [14] and [15] do not seem to provide any new insight into Algorithm 1.

Two key aspects of the behaviour of Algorithm 1 are as follows.

(1) The estimates produced by the algorithm are frequently exact ($\gamma = \|B\|_1$), usually “acceptable” ($\gamma \geq \|B\|_1/10$), and sometimes poor ($\gamma < \|B\|_1/10$). Several families of matrices are known for which $\gamma/\|B\|_1$ can be arbitrarily small [8], [9].

(2) The algorithm almost always converges after at most four iterations [7]–[9].

Based on these, and other observations, the following changes to Algorithm 1 were made in [9]. We will refer to the modified algorithm as Algorithm EST.

Definition of estimate. To overcome most of the poor estimates, γ is redefined as

$$(3) \quad \gamma = \max \left\{ \|y\|_1, \frac{\|w\|_1}{\|v\|_1} \right\}, \quad \text{where } w = Bv, \quad v_i = (-1)^{i+1} \left(1 + \frac{i-1}{n-1} \right).$$

The vector v is considered likely to “pick out” any large elements of B in those cases where such elements fail to propagate through to y .

Convergence test. The algorithm is limited to a minimum of two and a maximum of five iterations. Also, convergence is declared after computing ξ if the new ξ is the same as the previous one (which signals that convergence will be obtained on the current iteration) or if the new $\|y\|_1$ is no larger than the previous one. The latter event can happen only in finite precision arithmetic and indicates that a vertex e_j is being revisited—the onset of “cycling.”

which implies

$$\|T_n(\alpha)v\|_1 \geq \| |T_n(\alpha)|e \|_1 = n^2 - n\alpha,$$

and hence

$$\frac{\|T_n(\alpha)v\|_1/\|v\|_1}{\|T_n(\alpha)\|_1} \geq \frac{(n^2 - n\alpha)/(\frac{3}{2}n)}{2n - 2 - \alpha} \geq \frac{1}{3}.$$

For $\alpha > 1$ it does not seem possible to obtain a concise lower bound for $\|T_n(\alpha)v\|_1$, but a rough analysis indicates that the extra estimate cannot differ greatly from $\|T_n(\alpha)\|_1$ (and this is supported by numerical tests). In conclusion, Algorithm EST produces an acceptable estimate of $\|T_n(\alpha)\|_1$ for all n and α . We are not aware of any matrices for which the limit of five iterations in Algorithm EST is responsible for the algorithm producing a poor estimate.

3. Worst-case norm estimates. A class of matrices $B(\theta)$ is given in [9] for which $\gamma/\|B(\theta)\|_1$ can be arbitrarily small for Algorithm EST. In practice, rounding errors sustained in constructing $B(\theta)$ usually make the computed matrix one for which a good estimate is returned. Apart from the matrices $B(\theta)$ we have found exceedingly few matrices for which the estimate γ is less than $\frac{1}{3}\|B\|_1$. Here we list a few of the worst cases. Where $A = B^{-1}$ is specified we used Algorithm EST to estimate $\|A^{-1}\|_1$, which involves solving linear systems with A and A^T as coefficient matrices. $QR(B)$ denotes the triangular QR factor of B . The matrices are specified using the notation of [10]; they are of orders 16, 8, 50, and 100, respectively.

$$\begin{aligned} A &= \text{vand}(\text{seqa}(-1, 1, 16)), & \gamma/\|B\|_1 &= 0.199, \\ A &= \text{vand}(\text{seqa}(-1, 1, 8)), & \gamma/\|B\|_1 &= 0.288, \\ B &= QR(\text{chebspec}(50)), & \gamma/\|B\|_1 &= 0.326, \\ A &= \text{vand}(100), & \gamma/\|B\|_1 &= 0.207, \quad \|B\|_1 = 6.66 \times 10^{49}. \end{aligned}$$

In the last example Algorithm EST detected cycling. This matrix A is so violently ill-conditioned that the computed quantities in Algorithm EST may have no correct significant digits (the same applies to the “exact” $\|A^{-1}\|_1$ with which we compare γ !). These results were obtained using our PC-MATLAB implementation of Algorithm EST. PC-MATLAB uses IEEE standard arithmetic with machine epsilon $2^{-52} \approx 10^{-16}$. Different results might be obtained in another computing environment. To indicate the sensitivity of these worst-case estimates to rounding errors we mention that when A in the last example was scaled to $A/3$ Algorithm EST produced an exact estimate!

4. Applications. The most obvious application of Algorithm EST is to the estimation of the norm condition number $\kappa_p(A) = \|A\|_p\|A^{-1}\|_p$, $p = 1, \infty$. We summarize three other applications in which the algorithm has been found to be useful.

(1) In [12] a hybrid algorithm is developed for computing the polar decomposition. Algorithm EST is used to decide when to switch from one iteration to another by testing the convergence criterion $\|X_k^T X_k - I\|_1 < 1$, where $X_k \in \mathbb{R}^{n \times n}$. By using Algorithm EST formation of the matrix product $X_k^T X_k$ is avoided.

(2) When a square linear system $Ax = b$ is subject to perturbations ΔA and Δb satisfying inequalities $|\Delta A| \leq \epsilon E$, $|\Delta b| \leq \epsilon f$, a bound for the change in x can be derived that involves the condition number

$$\kappa_{E,f}(A, b) = \frac{\| |A^{-1}|E|x| + |A^{-1}|f \|_\infty}{\|x\|_\infty}.$$

The following idea from [1] shows how $\kappa_{E,f}(A, b)$ can be estimated with the aid of Algorithm EST. Defining $z = (E|x| + f)/\|x\|_\infty$ and $Z = \text{diag}(z)$ we have

$$\begin{aligned}\kappa_{E,f}(A, b) &= \| |A^{-1}|z \|_\infty = \| |A^{-1}|Ze \|_\infty \\ &= \| |A^{-1}Z|e \|_\infty = \| |A^{-1}Z| \|_\infty \\ &= \| A^{-1}Z \|_\infty = \|B\|_1,\end{aligned}$$

where $B = (A^{-1}Z)^T$. To apply Algorithm EST we just need to evaluate products $Bx = Z(A^{-T}x)$ and $B^T\xi = A^{-1}(Z\xi)$, which is easily done given the ability to solve linear systems involving A and A^T . LAPACK [3] uses Algorithm EST to provide an estimate of $\kappa_{E,f}(A, b)$ in routines for the solution of $Ax = b$ by Gaussian elimination with iterative refinement.

(3) The componentwise perturbation analysis mentioned in (2) can be extended to least squares problems [2], [4], [11] (i.e., to rectangular A and E), and a bound is obtained that contains the terms

$$\frac{\| |A^+|(E|x| + f) \|_\infty}{\|x\|_\infty} \quad \text{and} \quad \frac{\| |(A^T A)^{-1}|E^T|r \|_\infty}{\|x\|_\infty},$$

where A^+ is the pseudo-inverse of A and $r = b - Ax$. As in (2), each of these terms can be reduced to the form $\|B\|_1$, where it is straightforward in the context of solving a least squares problem to evaluate Bx and $B^T\xi$.

Acknowledgments. I thank Des Higham for pointing out an oversight in my original analysis of the $T_n(\alpha)$ example. The associate editor Robert Skeel also offered useful comments.

REFERENCES

- [1] M. ARIOLI, J.W. DEMMEL, AND I.S. DUFF, *Solving sparse linear systems with sparse backward error*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 165–190.
- [2] M. ARIOLI, I.S. DUFF, AND P.P.M. DE RIJK, *On the augmented system approach to sparse least-squares problems*, Numer. Math., 55 (1989), pp. 667–684.
- [3] C.H. BISCHOF, J.W. DEMMEL, J.J. DONGARRA, J.J. DU CROZ, A. GREENBAUM, S.J. HAMMARLING, AND D.C. SORENSEN, *Provisional contents*, LAPACK Working Note #5, Report ANL-88-38, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1988.
- [4] A. BJÖRCK, *Componentwise backward errors and condition estimates for linear least squares problems*, Manuscript, Department of Mathematics, Linköping University, Sweden, March 1988.
- [5] R.W. BRANKIN AND I. GLADWELL, *Codes for almost block diagonal systems*, Internat. J. Computers and Mathematics with Applications, 19 (1990), pp. 1–6.
- [6] T.F. COLEMAN AND L.A. HULBERT, *A direct active set algorithm for large sparse quadratic programs with simple bounds*, Tech. Report TR 88-926, Department of Computer Science, Cornell University, Ithaca, NY, 1988.
- [7] W.W. HAGER, *Condition estimates*, SIAM J. Sci. Statist. Comput., 5 (1984), pp. 311–316.
- [8] N.J. HIGHAM, *A survey of condition number estimation for triangular matrices*, SIAM Rev., 29 (1987), pp. 575–596.
- [9] ———, *FORTTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation (Algorithm 674)*, ACM Trans. Math. Software, 14 (1988), pp. 381–396.
- [10] ———, *A collection of test matrices in MATLAB*, Tech. Report 89-1025, Department of Computer Science, Cornell University, Ithaca, NY, 1989.
- [11] ———, *Computing error bounds for regression problems*, in Proc. Joint Summer Research Conference on Statistical Analysis of Measurement Error Models and Applications, P.

- Brown and W.A. Fuller, eds., Contemporary Mathematics, American Mathematical Society, Providence, RI, 1990.
- [12] N.J. HIGHAM AND R.S. SCHREIBER, *Fast polar decomposition of an arbitrary matrix*, Tech. Report 88-942, Department of Computer Science, Cornell University, Ithaca, NY, 1988; SIAM J. Sci. Statist. Comput., this issue, pp. 648–655.
- [13] NAG, NAG *Fortran Library Manual*, Mark 13, NAG Ltd., Oxford, U.K., 1988.
- [14] PHAM DINH TAO, *Convergence of a subgradient method for computing the bound norm of matrices*, Linear Algebra Appl., 62 (1984), pp. 163–182. (In French.)
- [15] ———, *Some methods for computing the maximum of quadratic form on the unit ball of the maximum norm*, Numer. Math., 45 (1984), pp. 377–401. (In French.)