

## Accurate SVDs of weakly diagonally dominant M-matrices\*

James Demmel<sup>1</sup>, Plamen Koev<sup>2</sup>

<sup>1</sup> Computer Science Division and Mathematics Dept., University of California, Berkeley, CA 94720, USA; e-mail: demmel@cs.berkeley.edu

<sup>2</sup> Mathematics Dept., Massachusetts Institute of Technology, Cambridge, MA 02139, USA; e-mail: plamen@math.mit.edu

Received July 31, 2002 /

Published online April 27, 2004 – © Springer-Verlag 2004

**Summary.** We present a new  $O(n^3)$  algorithm which computes the SVD of a weakly diagonally dominant M-matrix to high relative accuracy. The algorithm takes as an input the offdiagonal entries of the matrix and its row sums.

*Mathematics Subject Classification (1991):* 65F15

### 1 Introduction

We consider a row weakly diagonally dominant M-matrix  $A$  for which we know the offdiagonal entries  $a_{ij} \leq 0$ ,  $i \neq j$  and the row sums

$$s_i = \sum_{j=1}^n a_{ij} \geq 0$$

*accurately* (i.e. with small relative error).

In this paper we present a new  $O(n^3)$  algorithm which, given  $a_{ij}$ ,  $i \neq j$  and  $s_i$ , computes the SVD of  $A = U \cdot \Sigma \cdot V^T$  to high relative accuracy, meaning [4, 5]:

---

\* This material is based in part upon work supported by the LLNL Memorandum Agreement No. B504962 under DOE Contract No. W-7405-ENG-48, DOE Grants No. DE-FG03-94ER25219, DE-FC03-98ER25351 and DE-FC02-01ER25478, NSF Grant No. ASC-9813362, and Cooperative Agreement No. ACI-9619020.

Correspondence to: P. Koev

- The error  $|\sigma_i - \hat{\sigma}_i|$  in each computed singular value  $\sigma_i$  is bounded by  $O(\epsilon)\sigma_i$ , where  $\epsilon$  is the machine precision, i.e. the relative error is small. In particular, zero singular values are computed exactly.
- The angle  $\theta(u_i, \hat{u}_i)$  between the true left singular vector  $u_i$ , corresponding to a simple singular value  $\sigma_i$ , and the computed singular vector  $\hat{u}_i$  is bounded by  $O(\epsilon)/\text{relgap}_i$ , where  $\text{relgap}_i = \min_{i \neq j} |\sigma_j - \sigma_i| / (\sigma_i + \sigma_j)$  is the relative gap between  $\sigma_i$  and the nearest other singular value. An analogous statement holds for the computed right singular vectors  $\hat{v}_i$ .
- The cost of the algorithm is  $O(n^3)$ , independent of the condition number  $\kappa(A) = \|A\| \cdot \|A^{-1}\|$ .

In other words, our algorithm can be arbitrarily faster than a conventional SVD algorithm that uses high enough precision (growing with  $\kappa(A)$ ) to get the right answer. Similarly, our algorithm can be arbitrarily more accurate than a conventional SVD algorithm in fixed precision.

We consider the standard model of floating point arithmetic, which only assumes that the relative error of any arithmetic operation is small

$$(1) \quad fl(x \odot y) = (x \odot y)(1 + \delta)$$

where  $\odot \in \{+, -, *, /\}$  and  $|\delta| < \epsilon$  for some fixed  $\epsilon$ , called *machine precision*. We also assume that no underflow or overflow occurs. This model implies that products, quotients, and sums of like-signed quantities can be computed accurately (i.e. with low relative error), but expressions involving subtractive cancellation may not be (for example, the sum of three numbers can provably not be computed accurately in this model [6]). We maintain the relative accuracy of all quantities throughout our algorithm by never performing any subtractions of quantities with the same sign.

We chose the offdiagonals and row sums as parameters to describe  $A$  because they reveal the weakly diagonally dominant M-matrix structure of  $A$ , and determine the entries of  $A$ , the entries of  $A^{-1}$  [1], the smallest eigenvalue of  $A$  [2] and the SVD of  $A$  to high relative accuracy (§5).

## 2 SVD algorithm

We compute the SVD of  $A$  in two steps. First, we compute the LDU decomposition of  $A$ , resulting from *Gaussian elimination with complete pivoting (GECP)*, by using a slight modification of the GTH algorithm as discussed in [9] and referenced in [1], obtaining each entry of  $L$ ,  $D$  and  $U$  to high relative accuracy. Then we use the algorithm of Demmel et al. [5] which produces an accurate SVD of any matrix for which an accurate LDU decomposition is available.

The trick in computing an accurate LDU decomposition is in never subtracting quantities of the same sign in the process of GECP. The weakly

diagonally dominant M-matrix structure of  $A$  is preserved during pivoting (which, if it occurs, is diagonal) and Schur complementation [3, 7]. Therefore it suffices to show how to perform one step of GECP accurately and proceed by induction.

Let  $A'$  be obtained from  $A$  after one step of GECP. The row sums ( $s'_i$ ) and offdiagonal entries ( $a'_{ij}$ ,  $i \neq j$ ) of  $A'$  can be computed to high relative accuracy as follows. First, we compute the diagonal entries of  $A$  accurately as sums of positive quantities  $a_{ii} = s_i - \sum_{j \neq i} a_{ij}$ . Then we compute the offdiagonal entries of  $A'$ , which remain nonpositive

$$(2) \quad a'_{ij} = a_{ij} - \frac{a_{i1}a_{1j}}{a_{11}}.$$

Finally, we update the row sums of  $A'$ , which remain nonnegative

$$(3) \quad s'_i = s_i - \frac{a_{i1}}{a_{11}}s_1.$$

No subtractive cancellation can occur in computing (2) and (3) (since  $s_k > 0$ ,  $a_{kl} < 0$ ,  $k \neq l$  and  $a_{11} > 0$ ), therefore  $s'_i$  and  $a'_{ij}$  are computed to high relative accuracy. Now we can proceed by induction and conclude that every entry in the LDU decomposition of  $A$  resulting from GECP is computed to high relative accuracy.

**Algorithm 1** Given the offdiagonal entries  $a_{ij}$ ,  $i \neq j$  and row sums  $s_i = \sum_{j=1}^n a_{ij}$ , the following algorithm computes the SVD of a weakly diagonally dominant M-matrix  $A$  to high relative accuracy:

```

P = I_n ... the permutation matrix, initially I_n
for k = 1 : n
  for i = k : n do aii = si - ∑j≠i aij
  Pivot the largest aii to (k, k) position
  Update A, P and s accordingly if needed
  for i = k + 1 : n
    if akk ≠ 0
      aik = aik/akk
      si = si - aiksk
      for j = k + 1 : n
        if i ≠ j then aij = aij - aikakj
L and DU are stored in the lower/upper triangular
part of A respectively
Call the SVD algorithm from [5] passing L, D, U □

```

### 3 Cost

The cost of Algorithm 1 is  $O(n^3)$  arithmetic operations ( $n^3$  for computing the LDU decomposition and  $O(n^3)$  for the SVD algorithm of Demmel from [5]).

## 4 Error analysis

Because of (2) and (3), the relative error in the row sums will not exceed the relative error in the offdiagonal entries at every step of GECP. Then the error analysis in [9] goes over verbatim. The only difference between our algorithm and the GTH algorithm in [9] is that in our algorithm the diagonal entries that need to be computed at the  $k$ th step of Gaussian elimination are the sum of  $n - k + 1$  quantities—the off diagonal entries and the row sums, whereas in [9] the row sums are zero and the diagonal entries are the sums of  $n - k$  quantities. The analysis in [9] however treats the computation of the diagonal entries as being a result of  $n - k + 1$  arithmetic operations. Thus we can conclude that each entry of  $L$ ,  $D$  and  $U$  has relative error no larger than  $3n^3\epsilon$  [9].

## 5 Perturbation theory

It is easy to see from the above error analysis that small relative changes in the initial data (offdiagonals and row sums) will cause only small relative changes in the entries of  $L$  and  $U$ . These in turn cause only small relative changes in the singular values [5]. Therefore the SVD is a well-conditioned problem, independent of the values of the singular values.

## 6 Numerical experiments

To test our algorithm, we randomly generated a number of very ill-conditioned matrices as follows

1. We chose a  $20 \times 20$  matrix  $A$  with uniform random offdiagonal entries between  $-1$  and  $0$ ;
2. We chose 20 random row sums  $s_i$  of the form  $r \cdot 10^k$ , where  $r$  is a uniform random number in  $[0, 1]$  and  $k$  is a uniform random integer in  $[-40, -20]$ ;
3. We multiplied the  $i$ th row of  $A$  and  $s_i$  by another random number of the form  $r \cdot 10^j$ , where  $r$  is again uniform random in  $[0, 1]$ , and  $j$  is a uniform random integer in  $[-100, 100]$ .

The reason for Step 3 above is that without it, we will have one tiny singular value and the rest large, which is not much of a test. The resulting matrices had singular values, computed using our algorithm as implemented in MATLAB [8], ranging over about 110 orders of magnitude. We present a typical example in Figure 1.

We also computed the singular values using 200-digit arithmetic in Mathematica [10], and every time got the same result to 14 digits. These correct singular values are shown as  $\times$ 's in Figure 1. Note that we generated the

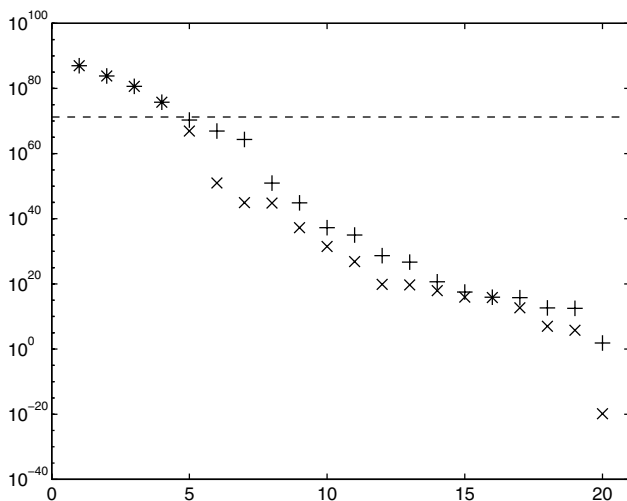


Fig. 1. The singular values of a  $20 \times 20$  M-matrix,  $\times$  =correct,  $+$  =usual

random numbers in MATLAB, output them as decimal numbers to a file, read them into Mathematica, converted them to 200 decimal digit big floats, and then formed the explicit matrix of which we computed the SVD. In other words, we more than likely made some errors in the 16th place just in forming the matrix in Mathematica. But the tiniest singular values still agreed with the MATLAB results to 14 places, confirming our claims that the problem is well-conditioned, even for computing the tiniest singular values.

In contrast the traditional algorithm only computed the singular values greater than  $\sigma_1 \epsilon$  with some relative accuracy (the singular values from the traditional algorithm are shown as  $+$ 's in Figure 1, and the value of  $\sigma_1 \epsilon$  is indicated by a horizontal dotted line). We could see from our tests (and is clear from the example above) the traditional algorithm tries to respect grading [5, §4] in the sense that the singular values that are less than  $\sigma_1 \epsilon$  are not clustered around  $\sigma_1 \epsilon$ , but are orders of magnitude smaller. Still, the tiny singular values computed by the traditional algorithm differ from the true singular values by orders of magnitude.

To verify the perturbation theory we perturbed the offdiagonal entries and the row sums of our test matrices with random noise in the 10th digit. Then computed the SVD of the thus perturbed matrices using Algorithm 1 and in Mathematica using 200 digit arithmetic. The singular values computed using each of these methods agreed with the singular values of the unperturbed matrices to 9 digits in each case confirming the fact that small relative perturbations in the initial data only cause small relative perturbations in the SVD.

## References

1. Alfa, A. S., Xue, J., Ye, Q.: Entrywise perturbation theory for diagonally dominant  $M$ -matrices with applications. *Numer. Math.* **90**, 401–414 (2002)
2. Alfa, A. S., Xue, J., Ye, Q.: Accurate computation of the smallest eigenvalue of a diagonally dominant  $M$ -matrix. *Math. Comp.* **71**(237), 217–236 (electronic), (2002)
3. Demmel, J.: *Applied Numerical Linear Algebra*, SIAM, 1997
4. Demmel, J.: Accurate SVDs of structured matrices. *SIAM J. Mat. Anal. Appl.* **21**(2), 562–580, (1999)
5. Demmel, J., Gu, M., Eisenstat, S., Slapničar, I., Veselić, K., Drmač, Z.: Computing the singular value decomposition with high relative accuracy. *Lin. Alg. Appl.* **299**(1–3), 21–80 (1999)
6. Demmel, J., Koev, P.: Necessary and sufficient conditions for accurate and efficient rational function evaluation and factorizations of rational matrices. In *Structured matrices in mathematics, computer science, and engineering. II* (Boulder, CO, 1999), pages 117–143. Amer. Math. Soc., Providence, RI, 2001
7. Golub, G., Van Loan, C.: *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, 3rd edition, 1996
8. The MathWorks, Inc., Natick, MA. *MATLAB Reference Guide*, 1992
9. O’Cinneide, C.: Relative-error for the LU decomposition via GTH algorithm. *Numer. Math.* **73**, 507–519 (1996)
10. Wolfram, S.: *Mathematica: A system for doing mathematics by computer*. Addison-Wesley, 1988