[14] G. L. Miller and J. H. Reif. Parallel tree contraction and its application. In *Proceedings of 16th Annual Symposium on the Foundations of Computer Science*, pages 478–489. IEEE, 1985.

[15] D. O'Leary and G. W. Stewart. Computing the eigenvalues and eigenvectors of symmetric arrowhead matrices. *J. Comp. Phys.*, 90:497–505, 1990.

[16] B. Parlett. *The Symmetric Eigenvalue Problem*. Prentice Hall, Englewood Cliffs, NJ, 1980.

[17] A. Sameh and D. Kuck. A parallel QR algorithm for symmetric tridiagonal matrices. *IEEE Trans. Computers*, C-26(2), 1977.

[18] D. Sorensen and P. Tang. On the orthogonality of eigenvectors computed by divide-and-conquer techniques. *SIAM J. Num. Anal.*, 28(6):1752–1775, 1991.

[19] P. Swarztrauber. A parallel algorithm for computing the eigenvalues of a symmetric tridiagonal matrix. *Math. Comp.*, 60, 1993. to appear.

[20] J. H. Wilkinson. *The Algebraic Eigenvalue Problem*. Oxford University Press, Oxford, 1965.

[21] Zhonggang Zeng. The acyclic eigenproblem can be reduced to the arrowhead one. preprint, Michigan State University, 1992.

[22] Hongyuan Zha. A two-way chasing scheme for reducing a symmetric arrowhead matrix to tridiagonal form. Scientific computing and computational mathematics, Stanford University, 1992.

Acknowledgement

The authors thank Richard Brualdi, who pointed out references [3, 5], as well as John Gilbert, who pointed out reference [14].

References

[1] A. Abdallah and Y. Hu. Parallel VLSI computing array implementation for signal subspace updating algorithm. *IEEE Trans. Acoustics, Speech, Signal Processing*, ASSP 37(5):742–748, 1989.

[2] P. Arbenz and G. Golub. QR-like algorithms for arrow matrices. *SIAM J. Mat. Anal. Appl.*, 13:655–658, 1992.

[3] T. Austin. The enumeration of point labeled chromatic graphs and trees. *Canad. J. Math.*, 12 pp. 535–545, 1960.

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

[5] N. Biggs. *Algebraic Graph Theory*. Cambridge University Press, 1974.

[6] R. P. Brent. The parallel evaluation of general arithmetic expressions. *Journal of the ACM*, 21(2):201–208, April 1974.

[7] J. J. M. Cuppen. A divide and conquer method for the symmetric tridiagonal eigenproblem. *Numer. Math.*, 36:177–195, 1981.

[8] P. Deift, J. Demmel, L.-C. Li, and C. Tomei. The bidiagonal singular values decomposition and Hamiltonian mechanics. *SIAM J. Num. Anal.*, 28(5):1463–1516, October 1991.

[9] J. Demmel and W. Kahan. Accurate singular values of bidiagonal matrices. *SIAM J. Sci. Stat. Comput.*, 11(5):873–912, September 1990.

[10] J. Dongarra and D. Sorensen. A fully parallel algorithm for the symmetric eigenproblem. *SIAM J. Sci. Stat. Comput.*, 8(2):139–154, March 1987.

[11] E. Jessup and I. Ipsen. Improving the accuracy of inverse iteration. *SIAM J. Sci. Stat. Comput.*, 13(2):550–572, 1992.

[12] E. Jessup and D. Sorensen. A parallel algorithm for computing the singular value decomposition of a matrix. Mathematics and Computer Science Division Report ANL/MCS-TM-102, Argonne National Laboratory, Argonne, IL, December 1987.

[13] W. Kahan. Accurate eigenvalues of a symmetric tridiagonal matrix. Computer Science Dept. Technical Report CS41, Stanford University, Stanford, CA, July 1966 (revised June 1968).

6 Algorithms and Open Problems

In [8] a perturbation theorem for singular vectors of bidiagonal matrices is proven, which shows that the appropriate condition number for the i-th singular vector is the reciprocal of the relative difference between the i-th singular value and next closest one. It would be interesting to extend this to the biacyclic case.

Given the perturbation theory, it would be nice to compute the singular vectors as accurately as they deserve. A natural candidate is inverse iteration, but even in the simple case of symmetric tridiagonal matrices, open problems remain. In particular there is no absolute guarantee that the computed eigenvectors are orthogonal, although in practice the algorithm can be made quite robust [11].

In the "extreme" cases of tridiagonal and arrow matrices, we know how to compute the inertia in $O(\log n)$ time, using the so-called parallel-prefix algorithm in the tridiagonal case [17, 19], and more simply in the arrow case. The stability in the tridiagonal case is unknown, but in practice it appears to be stable. We can extend this to the general symmetric acyclic case in two ways. First, the tree describing the expression whose final value is d_i has at most n leaves. From [6] we know any such expression tree can be evaluated in at most $4 \log_2 n$ parallel steps, although stability may be lost. Another approach, which includes parallel prefix and the algorithm in [15] as special cases, is based on [14]. The idea is to simply evaluate the tree greedily, summing k leaves of a single node in $O(\log_2 k)$ steps wherever possible, and collapsing a chain of k nodes into a single node via parallel prefix in $O(\log_2 k)$ steps wherever possible. If we could understand the numerical stability of parallel prefix, we could probably analyze this more general scheme as well.

Divide and conquer [7, 10, 18, 12] has been widely used for the tridiagonal eigenproblem and bidiagonal singular value decomposition. This can be straightforwardly extended to the acyclic case. In terms of the tree, just remove the root by a "rank two tearing", solve the independent child subtrees recursively and in parallel, and merge the results by solving the secular equation [21]. Any node can be the root, and to be efficient it is important that no subtree be large. In the tridiagonal case, the rank-two tearing corresponds to zeroing out two adjacent offdiagonal entries; note this is slightly different from the algorithm in the literature which uses rank-one tearing, although the secular equation to be solved is very similar. Also in the tridiagonal case, there are always two subtrees of nearly equal size. In a general tree one can only make sure that no subtree has more than half the nodes of the original tree (this is easily done in $O(n)$ time via depth first search).

QR does not appear to extend beyond the tridiagonal case. The case of arrow matrices was analyzed in [2], where it was shown that no QR algorithm could exist. A simpler proof arises from noting that two steps of LL^T is equivalent to one step of QR in the positive definite case, and so the question is whether the sparsity pattern of $T_0 = LL^T$ is the same as that of $T_1 = L^T L$; this is easily seen to include only tridiagonal T_0 among all symmetric acyclic matrices.

Finally, we conjecture that the set of symmetric acyclic matrices is the complete set of symmetric matrices whose eigenvalues can be computed with tiny componentwise relative backward error independent of the values of the matrix entries.

The proof depends strongly on there not being any fill-in and on each offdiagonal entry being computable by a single division. Since these properties hold if and only if the graph $G'(T)$ is symmetric acyclic, we strongly suspect that this is the only class of matrices whose eigenvalues can always be computed with tiny componentwise relative backward error.

We now apply Theorem 2 to compute singular values of biacyclic matrices to high relative accuracy. So suppose B is a matrix whose graph $G(B)$ is acyclic. Consider the symmetric matrix

$$A = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix}$$

whose positive eigenvalues are the singular values of B. It is also immediate that the graph $G'(A) = G(B)$. Therefore B is biacyclic if and only if A is symmetric acyclic, so we can apply the above algorithm to compute B's singular values to high relative accuracy.

One other algorithm is worth mentioning. If A is symmetric positive definite and symmetric acyclic, then its Cholesky factor L is biacyclic (provided we do the elimination in the same postorder as the algorithm Gt), has the "lower half" of the sparsity pattern of A. It may occasionally be more accurate to compute A's eigenvalues by first computing L, computing its singular values by bisection, and then squaring the singular values to get A's eigenvalues [4]. This is the case, for example, for the tridiagonal matrix with 2's on the diagonal and 1's on the offdiagonal.

5 Examples

We give various examples of acyclic sparsity patterns, beginning with acyclic $G(S)$. Given any acyclic sparsity pattern, others can be generated either by permuting rows and/or columns, or by adding more zeros. Since all square biacyclic matrices have monomial (or zero) determinants, this means we can permute them to be upper triangular. In addition to bidiagonal matrices, some other examples are

$$\begin{bmatrix} x & & & & x \\ & x & & & x \\ & & x & & x \\ & & & x & x \\ & & & & x \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x & & & x & & \\ & x & & & x & \\ & & x & & & x \\ & & & x & & x \\ & & & & x & x \\ & & & & & x \end{bmatrix} .$$

To get symmetric acyclic matrices A, one can always take an acyclic B and set $A = \begin{bmatrix} 0 & B \\ B^T & 0 \end{bmatrix} - \lambda I$. Some other examples are

$$\begin{bmatrix} x & & & & x \\ & x & & & x \\ & & x & & x \\ & & & x & x \\ x & x & x & x & x \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} x & & & x & & \\ & x & & & x & \\ & & x & & & x \\ x & & & x & & x \\ & x & & & x & x \\ & & x & x & x & x \end{bmatrix} .$$

We see that each entry of T is used just once as follows. $T_{i\,i}$ is only used when visiting node i, and $T_{i\,j}$ is used only once, when visiting i if j is a child of i or when visiting j if i is a child of j in the postorder traversal tree.

Now denote the d computed when visiting node i by d_i. The floating point operations performed while visiting node i are then

$$d_i = fl\left(T_{i\,i} - x - \sum_{\substack{\text{all children} \\ j \text{ of } i}} \frac{T_{i\,j}^2}{d_j} \right) \tag{4.1}$$

To analyze this formula, we will let subscripted ε's denote independent quantities bounded in absolute value by ε_M. We will also make standard approximations like $(1+\varepsilon_1)^{\pm 1}(1+\varepsilon_2)^{\pm 1} = 1+2\varepsilon_3$.

Since we do not know the number of terms or the order of the sum in equation (4.1), we will make the worst case assumption that there are $v \leq n-1$ terms where v is the maximum degree of any node in the graph $G'(S)$. This leads to

$$d_i = (1+(v+1)\varepsilon_{ia})T_{ii} - (1+(v+1)\varepsilon_{ib})x - \sum_{\substack{\text{all children} \\ j \text{ of } i}} (1+(v+3)\varepsilon_{ij})\frac{T_{ij}^2}{d_j} \tag{4.2}$$

or

$$\frac{d_i}{1+(v+1)\varepsilon_{ia}} = T_{ii} - x + (2v+2)\varepsilon_{ic}x - \sum_{\substack{\text{all children} \\ j \text{ of } i}} \frac{((1+(v+2)\varepsilon_{ij'})T_{ij})^2}{d_j} \tag{4.3}$$

Let ε_{ja} be the roundoff error corresponding to ε_{ia} committed when computing d_j. Then

$$\frac{d_i}{1+(v+1)\varepsilon_{ia}} = T_{ii} - x + (2v+2)\varepsilon_{ic}x - \sum_{\substack{\text{all children} \\ j \text{ of } i}} \frac{((1+(1.5v+2.5)\varepsilon_{ij''})T_{ij})^2}{d_j/(1+(v+1)\varepsilon_{ja})} \tag{4.4}$$

or, finally,

$$d_i' = T_{ii} - x + (2v+2)\varepsilon_{ic}x - \sum_{\substack{\text{all children} \\ j \text{ of } i}} \frac{((1+(1.5v+2.5)\varepsilon_{ij''})T_{ij})^2}{d_j'} \tag{4.5}$$

where $d_i' = d_i/(1+\varepsilon_{ia})$. Equation (4.5) tells us that the d_i' are the exact diagonal entries of D in $P(T+\delta T - xI)P^T = LDL^T$. Since they obviously have the same signs as the d_i, this proves Theorem 2.

Figure 2: Computing count(T, x)

```
call Count(i, x, d, s) where i is any node 1 ≤ i ≤ n
return count(T, x) = s

procedure Count(i, x, d, s)
    /* i and x are input parameters, d and s are output parameters */
    d = T_ii − x
    s = 0
    for all children j of i do
        call Count(j, x, d', s')
        d = d − T_ij^2 / d'
        s = s + s'
    end for
    if d < 0, then s = s + 1
    return d and s
end procedure
```

$$|\delta T_{ii}| \le (2v + 2)\varepsilon_M |x| .$$

Here $v \le n - 1$ is the maximum degree of any node in the graph of T. In other words, the computed count(T, x) *is the exact value of* count$(T + \delta T, x)$ *where δT is bounded as above.*

This is essentially identical to the standard error analysis of Sturm sequence evaluation for symmetric tridiagonal matrices [9, Sec. 6] [13] (this is stronger than the result in [20, p. 303]).

Our algorithm simply performs symmetric Gaussian elimination on $T - xI$: $P(T - xI)P^T = LDL^T$ where P is a permutation matrix, L is unit lower triangular and D is diagonal. Then by Sylvester's Inertia Theorem [16], count(T, x) is simply the number of negative diagonal entries of D. The order of elimination is the same as a postorder traversal of the nodes of the acyclic graph. Since leaves, which have degree 1, are eliminated first, there is no fill-in during the elimination, and all offdiagonal entries L_{ij} of L can be computed by simply dividing $L_{ij} = T_{ij}/D_{jj}$.

We assume the graph $G'(S)$ is connected, since otherwise the matrix can be reordered to be block diagonal (one diagonal block per connected component of $G'(S)$), and the inertia of each diagonal block can be computed separately. The algorithm Count(i, x, d, s) in Figure 2 assumes the matrix is stored in graph form. It does a postorder traversal of the acyclic graph $G'(S)$, and may be called starting at any node $1 \le i \le n$. In addition to i, x is an input parameter. The variables d and s are output parameters; on return s is the desired value of count(T, x).

To prove Theorem 2, we will exploit the symmetric acyclicity of T to show that each computed quantity and original entry of T is used (directly) just once during the entire computation, and then use this to "push" the rounding error back to the original data.

term in the determinant corresponds to a choice of s entries of M located in disjoint rows and columns, and each such choice of s entries selects a perfect match in $G(M)$.

Now suppose a square submatrix M of A has at least two terms in its determinant. These correspond to two different perfect matchings. Take the symmetric difference of the edges in these matchings. This symmetric difference forms a cycle, which we get by following edges of the two matchings in alternation. Thus $G(M)$ contains a cycle, and so must $G(A)$ since it includes $G(M)$.

Now suppose $G(A)$ contains a cycle. Assume without loss of generality that it is a simple cycle, i.e. it is connected and visits each node once. Let M be the corresponding square submatrix. This cycle determines two perfect matchings in $G(M)$, consisting of alternate edges of the cycle. This means $\det(M)$ has at least two terms. □

To prove that Property 2 implies Property 1, we will show the contrapositive. So assume $G(A)$ contains a cycle, and let M be an s by s submatrix whose determinant has at least 2 terms. This means we may choose all the entries of M to be nonzero but such that M is exactly singular. Thus its singular values include at least one which is exactly zero. Scale M so that its entry of smallest absolute value is 1, and let $\sigma = \|M\|_2 \ge 1$. Now let $A(M,\eta)$ denote the matrix with sparsity S, submatrix M, and other nonzero entries equal to η. Then $A(M,0)$ will have at least $\min(m,n) - s + 1$ zero singular values, $\min(m,n) - s$ from the zero rows and columns outside M, and 1 from the singularity of M. By standard perturbation theory $A(M,\eta)$ will have at least $\min(m,n) - s + 1$ singular values no larger than $mn\eta$. Now change a smallest entry of M from 1 to $1+x$ to get M_x; thus x is also the relative change in this entry. Then $|\det(M_x)| \ge x$, and so $\sigma_{min}(M_x) \ge |x|/(\sigma+|x|)^{s-1}$. This means $\sigma_s(A(M_x,\eta)) \ge |x|/(\sigma+|x|)^{s+1} - mn\eta$, whereas $\sigma_s(A(M,\eta)) \le mn\eta$. Thus

$$\frac{\sigma_s(A(M_x,\eta))}{\sigma_s(A(M,\eta))} \ge \frac{\frac{x}{(\sigma+|x|)^{s+1}} - mn\eta}{mn\eta} = \frac{x}{mn\eta(\sigma+|x|)^{s+1}} - 1 \; .$$

If Property 2 held, then this last quantity would be bounded in absolute value by $1+|x|$ no matter how small η were, which is impossible. This completes the proof that Property 2 implies Property 1, and so also completes the proof of Theorem 1.

4 A bisection algorithm for computing eigenva tiny backward error

Let ε_M denote the machine precision. We will assume the usual model of floating point error, $fl(a \otimes b) = (a \otimes b)(1+\delta)$ with $|\delta| \le \varepsilon_M$, and assume neither underflow nor overflow occur. (Of course, a practical algorithm would need to account for overflow. This can be done analogously to the way overflow is accounted for in standard tridiagonal bisection [13].)

In this section we will show how to compute the eigenvalues of a symmetric acyclic matrix T with tiny componentwise relative backward error. Our main result is

Theorem 2 *The algorithm in Figure 2 computes count(T,x), the number of eigenvalues of T less than x, with a backward error δT with the following properties:*

$|\delta T_{ij}| \le (1.5v+2.5)\varepsilon_M |T_{ij}|$ *when* $i \neq j$.

It is known that the singular values of A are the same as the positive eigenvalues of the pencil

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} - \lambda I$$

which are in turn the same as the positive eigenvalues of the equivalent symmetric definite pencil

$$\begin{bmatrix} R & 0 \\ 0 & C \end{bmatrix} \left(\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} - \lambda \cdot I \right) \begin{bmatrix} R & 0 \\ 0 & C \end{bmatrix} = \begin{bmatrix} 0 & E \\ E^T & 0 \end{bmatrix} - \lambda \cdot \begin{bmatrix} R^2 & 0 \\ 0 & C^2 \end{bmatrix} \equiv F - \lambda D^2$$

Now suppose we perturb A by changing nonzero entry A_{ij} to βA_{ij}, resulting in the perturbed matrix A'. Apply the algorithm in Lemma 1 to compute a new R' and C'. We assume without loss of generality that A_{ij} appears in numerators of R_i (otherwise consider A^T). By Lemma 1 either $R'_k = R_k$ or $R'_k = \beta R_k$, and either $C'_k = C_k$ or $C'_k = \beta^{-1} C_k$. Note we may multiply R by any nonzero γ and divide C by γ without changing the fact that $RAC = E$, so we divide R by $|\beta|^{1/2}$ and multiply C by $|\beta|^{1/2}$. We obtain R' and C' matrices each of whose entries differs from the corresponding entry of R and C by factors of $|\beta|^{\pm 1/2}$. In particular, this implies

$$|\beta|^{-1} \leq \frac{x^T R^2 x}{x^T R'^2 x} \leq |\beta| \quad \text{and} \quad |\beta|^{-1} \leq \frac{x^T C^2 x}{x^T C'^2 x} \leq |\beta|$$

for any nonzero vector x. Let $D' = \mathrm{diag}(R', C')$ as we above defined $D = \mathrm{diag}(R, C)$. Then

$$|\beta|^{-1} \leq \frac{y^T D^2 y}{y^T D'^2 y} \leq |\beta|$$

for any nonzero vector y. We may now apply [4, Lemma 2] to conclude that

$$\sigma_k(A) = \min_{\mathbf{S}^k} \max_{\substack{x \in \mathbf{S}^k \\ \|x\|_2 = 1}} \frac{x^T F x}{x^T D^2 x}$$

and

$$\sigma_k(A') = \min_{\mathbf{S}^k} \max_{\substack{x \in \mathbf{S}^k \\ \|x\|_2 = 1}} \frac{x^T F x}{x^T D'^2 x} \ ,$$

where the minima are over all $k + \max(n, m)$ dimensional subspaces $\mathbf{S}^k$, can differ by no more than a factor of β. This proves that Property 1 implies Property 2.

Lemma 2 *Let A have sparsity pattern S, and let all its nonzero entries be independent indeterminates. Then $G(S)$ is acyclic if and only if all minors of A are either 0 or monomials.*

PROOF. We begin by noting that to each term in the determinant of an s by s square matrix M corresponds a unique perfect matching in graph $G(M)$. This is because each

Figure 1: Computing R and C

```
if q is row node r_i then
    if r_i is the root then
        R_i = 1
    else
        suppose c_j is the parent of q
        R_i = 1/(A_ij C_j)
    end
else (q must be column node c_j) then
    if c_j is the root then
        C_j = 1
    else
        suppose r_i is the parent of q
        C_j = 1/(A_ij R_i)
    end
end if
```

PROOF. Since $G(S)$ is acyclic, it is a forest of trees. We may consider each tree independently. We traverse each tree via depth first search, and execute the program in Figure 1 when first visiting node q.

The depth first search visits each node once. Since the graph is bipartite, row nodes and column nodes alternate, so the parent of a row node is a column node and vice versa. Since each node is visited once, the above program is executed once for each edge in the tree, i.e. once for each nonzero entry A_{ij}, corresponding to the edge connecting nodes r_i and c_j. Thus each R_i and C_j is set exactly once. Since the i, j entry of RAC is $R_i A_{ij} C_j$, we see immediately from the way R_i and C_j are defined that this quantity is 1 if $A_{ij} \neq 0$ (and 0 otherwise). Since each A_{ij} is used once during the graph traversal, each R_i and C_j must be be a quotient of monomials. If A_{ij} is first used in R_i, then the formulas in the above program and the fact the row and column nodes alternate mean that A_{ij} will only appear in denominators of entries of R and numerators of entries of C. Alternatively, if A_{ij} is first used in C_j, then A_{ij} will only appear in denominators of entries of C and numerators of entries of R. ☐

The rest of the proof that Property 1 implies Property 2 mimics that of [4, Thm 1]. Let E be the matrix of ones and zeros with sparsity S, so that $RAC = E$. Write $R = S_R|R|$ where $|R|$ is the matrix of absolute values of R, and S_R is a diagonal matrix with $|S_R| = I$. Similarly write $C = S_C|C|$. Then

$$A = R^{-1} E C^{-1} = S_R^{-1} |R|^{-1} E |C|^{-1} S_C^{-1} = S_R^{-1} |A| S_C^{-1}$$

so that A is related to $|A|$ by pre- and postmultiplication by diagonal orthogonal matrices. In particular, A and $|A|$ have the same singular values. We will henceforth assume without loss of generality that A is nonnegative and so R and C are also nonnegative.

for the singular values of biacyclic matrices, and section 3 proves it. Section 4 shows how to compute eigenvalues of symmetric acyclic matrices with tiny componentwise relative backward error, and applies this to compute the singular values of biacyclic matrices to high relative accuracy. Section 5 give some examples of matrices with acyclic sparsity patterns. Section 6 discusses algorithms and open problems.

2 Statement of Perturbation Theorem for Singul

In this section we define two properties of sparsity patterns of matrices, one about graph theory and one about perturbation theory. Our main result, which we prove in the next section, is that these properties are equivalent.

Let A be an m by n matrix with a fixed sparsity pattern S.

Property 1 . *$G(S)$ is acyclic.*

Property 2 . *Given sparsity pattern S, let A be any matrix with this sparsity, and A_{ij} any nonzero entry. Let β be any nonzero constant. Let $A' = A$ except for $A'_{ij} = \beta A_{ij}$. Then for all singular values $\sigma_k(A')$*

$$\min(|\beta|, |1/\beta|)\sigma_k(A) \le \sigma_k(A') \le \max(|\beta|, |1/\beta|)\sigma_k(A)$$

If p entries of A are simultaneously perturbed by possibly different factors β, all of which satisfy $|\beta - 1| \le \epsilon \ll 1$, Property 2 can be applied p times to show no singular value can change by a factor outside the interval from $(1-|\epsilon|)^p = 1 - p|\epsilon| - O(\epsilon^2)$ to $(1+|\epsilon|)^p = 1 + p|\epsilon| + O(\epsilon^2)$. Since the maximum number of nonzeros is $m+n-1$, the relative perturbation in any singular value is bounded by $(m+n-1)|\epsilon| + O(\epsilon^2)$.

Our main result is

Theorem 1 *Properties 1 and 2 of a sparsity pattern S are equivalent.*

One could ask if a weaker perturbation property than Property 2 might hold for even more sparsity patterns than biacyclic ones. In particular, we could consider restricting the condition so that β must be close to 1 for some relative perturbation bound to hold. One can still show that even asking for this restricted perturbation property limits us to biacyclic matrices.

3 Proof of Perturbation Theorem for Singular V

First we will prove that Property 1 implies Property 2, and then the converse.

Lemma 1 *Let A have sparsity pattern S, and suppose $G(S)$ is acyclic. Then there are diagonal matrices R and C such that each entry of RAC is either 0 or 1. Each diagonal entry R_i of R or C_j of C is a quotient of monomials in the entries of A. In each monomial each distinct factor A_{ij} which appears has unit exponent. Each A_{ij} can appear only in numerators of entries of R and denominators of entries of C, or vice versa, in denominators of entries of R and numerators of entries of C.*

is simple: a sparsity pattern has this property if and only if its associated bipartite graph is acyclic.

We define this undirected graph as follows. Let S be a sparsity pattern for m by n matrices; in other words, S is a list of the entries permitted to be nonzero. Let $G(S)$ be a bipartite graph with one group of nodes $\{r_1, \ldots, r_m\}$ representing the m rows and one group $\{c_1, \ldots, c_n\}$ representing the n columns. There is an edge between r_i and c_j if and only if A_{ij} is permitted to be nonzero. We will sometimes write $G(A)$ instead of $G(S)$, where S is the sparsity pattern of A. We will call a matrix A and its sparsity pattern S *biacyclic* if the graph $G(S)$ is acyclic.

We also present another equivalent perturbation property of biacyclic matrices which is quite strong: multiplying any single matrix entry by any factor $\beta \neq 0$ cannot increase or decrease any singular value by more than a factor of β.

Sparsity patterns with this property have at most $n+m-1$ nonzero entries. There are a great many such sparsity patterns. Let us consider only m by n sparsity patterns S which cannot be permuted into block diagonal form (this means $G(S)$ is connected). Then the number of different such sparsity patterns is equal to the number of spanning trees on connected bipartite graphs with $m+n$ vertices; this number is $m^{n-1} n^{m-1}$ [5, p. 38] [3]. If we only wish to count sparsity patterns which cannot be made identical by reordering the rows and columns, a very simple lower bound on the number of such equivalence classes is $m^{n-1} n^{m-1}/(n!m!)$. In the square case $n = m$, Stirling's formula lets us approximate this lower bound by $e^{2n}/(2\pi n^3)$, which grows quickly.

Since we know the singular values of these biacyclic matrices are determined to high relative accuracy by the data, it makes sense to try to compute them this accurately. We present a bisection algorithm which does this. The same algorithm can compute the eigenvalues of arbitrary "symmetric acyclic" matrices with tiny componentwise relative error. We define symmetric acyclicity of a symmetric matrix as follows. Given a sparsity pattern S of an n by n symmetric matrix, we define an undirected graph $G'(S)$ by taking n nodes, and connecting node i with node $j \neq i$ if and only if the (i, j) entry is nonzero. The matrix A and its symmetric sparsity pattern S are called *symmetric acyclic* if the graph $G'(S)$ is acyclic. (We will sometimes write $G'(A)$ instead of $G'(S)$ where S is the sparsity pattern of A.) The algorithm evaluates the inertia of such a matrix by doing symmetric Gaussian elimination, with the order of elimination determined by a postorder traversal of $G'(S)$.

In summary, the well-known attractive properties of bidiagonal matrices B and symmetric tridiagonal matrices T, that the singular values of B can be computed to high relative accuracy and the eigenvalues of T computed with tiny componentwise relative backward error, have been extended to biacyclic and symmetric acyclic matrices. In the case of computing singular values, we have shown that this extension is complete: no other sparsity patterns have this property. We conjecture that the set of symmetric acyclic matrices is also the complete set of symmetric matrices whose eigenvalues can be computed with tiny componentwise relative backward error independent of the values of the matrix entries.

Other algorithms for the special case of "arrow" matrices are discussed in [1, 2, 15, 22]. This work generalizes the adaptations of bisection to arrow matrices, and is almost certainly more stable than the QR-based schemes.

The rest of this paper is organized as follows. Section 2 states the perturbation theorem

On computing accurate singular values and eigenvalues of matrices with acyclic graphs

James W. Demmel *
Computer Science Division and Department of Mathematics
University of California
Berkeley, California 94720

William Gragg †
Department of Mathematics
Naval Postgraduate School
Monterey, California 93943

Abstract

It is known that small relative perturbations in the entries of a bidiagonal matrix only cause small relative perturbations in its singular values, independent of the values of the matrix entries. In this paper we show that a matrix has this property if and only if its associated bipartite graph is acyclic. We also show how to compute the singular values of such a matrix to high relative accuracy. The same algorithm can compute eigenvalues of symmetric matrices with acyclic graphs with tiny componentwise relative backward error. This class includes tridiagonal matrices, arrow matrices, and exponentially many others.

1 Introduction

In [9] it was shown that small relative perturbations in the entries of a bidiagonal matrix B only cause small relative perturbations in its singular values. This is true independent of the values of the nonzero entries of B. This property justifies trying to compute the singular values of B to high relative accuracy, and is essential to the error analyses of the corresponding algorithms [9].

Since this attractive property of bidiagonal matrices is independent of the values of the nonzero entries, it is really just a function of the sparsity pattern of bidiagonal matrices. In this paper we completely characterize those sparsity patterns with the property that independent of the values of the nonzero entries, small relative perturbations of the matrix entries only cause small relative perturbations of the singular values. The characterization

*The author was supported by NSF grants ASC-9005933 and CCR-9196022, and DARPA grant DAAL03-91-C-0047 via a subcontract from the University of Tennessee. This work was performed during a visit to the Institute for Mathematics and its Applications at the University of Minnesota.

†The author also acknowledges the Institute for Mathematics and its Applications at the University of Minnesota.