A FAST RECURSIVE ALGORITHM FOR CONSTRUCTING MATRICES WITH PRESCRIBED EIGENVALUES AND SINGULAR VALUES

MOODY T. CHU*

Abstract. The Weyl-Horn theorem characterizes a relationship between the eigenvalues and the singular values of an arbitrary matrix. Based on that characterization, a fast recursive algorithm is developed to construct numerically a matrix with prescribed eigenvalues and singular values. Beside being theoretically interesting, the technique could be employed to create test matrices with desired spectral features. Numerical experiment shows this algorithm is quite efficient and robust.

Key words. eigenvalues, singular values, inverse problems, Weyl-Horn theorem, recursive algorithm

1. Introduction.

It is perhaps valid to state that eigenvalues and singular values are two of the most distinguishing characteristics in any given general square matrix. Eigenvalue analysis, for example, simplifies the representation of complicated systems, sheds light on the asymptotic behavior of differential equations, and helps to understand the performance of important numerical algorithms. Information on singular values, on the other hand, assumes a critical role whenever there is presence of roundoff error or inexact data. Many fundamental topics in linear algebra and important applications in practice are best understood when formulated in terms of the singular value decomposition.

Over the years, one of the most fruitful developments in numerical linear algebra, which serves as computational platforms for a variety of application problems, is that of efficient and stable algorithms for the computation of eigenvalues and singular values of a given matrix. This paper concerns an interesting inverse problem, i.e., instead of computing eigenvalues and singular values from a given matrix, we are interested in constructing a matrix that has prescribed eigenvalues and singular values. In this paper we propose a fast recursive algorithm to accomplish this construction. Our method, though simple and derived from an old theory, appears to be the first ever in dealing with this problem numerically.

For a Hermitian matrix A, the singular values of A are simply the absolute values of eigenvalues of A. For non-Hermitian matrices, some restrictions must be placed between eigenvalues and singular values. In-

*Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205. This research was supported in part by the National Science Foundation under grant DMS-9422280.

deed, the following key result is first proved by Weyl [12].

THEOREM 1.1. *Given any $n \times n$ matrix, A, let its eigenvalues $\lambda_1, \ldots, \lambda_n$ and singular values $\alpha_1, \ldots, \alpha_n$ be arranged in the order*

$$|\lambda_1| \geq \ldots \geq |\lambda_n| \quad \text{and} \quad \alpha_1 \geq \ldots \geq \alpha_n. \tag{1.1}$$

Then

$$\prod_{j=1}^{k} |\lambda_j| \leq \prod_{j=1}^{k} \alpha_j, \quad k = 1, \ldots, n-1, \tag{1.2}$$

$$\prod_{j=1}^{n} |\lambda_j| = \prod_{j=1}^{n} \alpha_j. \tag{1.3}$$

If $|\lambda_n| > 0$, then the conditions (1.2) and (1.3) may also be referred to as that the sequence $\{\log \alpha_i\}$ *majorizes* the sequence $\{\log |\lambda_i|\}$. Theory of majorization and its applications can be found in [6]. It turns out, due to Horn [3], that the above necessary conditions are also sufficient, i.e., (1.2) and (1.3) are the only relations between the eigenvalues and the singular values of any general matrix.

The result by Weyl and Horn then gives rise to an interesting inverse problem, i.e., to numerically construct a square matrix that possesses a prescribed set of eigenvalues and singular values. Such a construction might be useful in designing matrices with desired spectral specifications. Many important properties, such as the conditioning of a matrix, are determined by eigenvalues or singular values.

Our approach is based on the original proof by Horn that, in turn, is completed by the mathematical induction. Upon a careful study of his inductive argument, we realize that, with the aid of modern programming languages that allow a subprogram to invoke itself recursively, Horn's induction proof can be transformed into a recursive algorithm. In this way, we are able to construct numerically a matrix with prescribed eigenvalues and singular values, so long as the conditions (1.2) and (1.3) are satisfied. Our contribution in this paper is twofold: We significantly relax one key component in Horn's proof that, in return, enables us to implement a fast recursive algorithm.

This paper is organized as follows: We begin in Section 2 with a demonstration on how a 2×2 matrix can be handled explicitly, given its eigenvalues and singular values. This 2×2 construction plays a significant role in the subsequent steps since the original problem is eventually reduced to 2×2 blocks. We slightly extend Horn's result by showing that if the prescribed eigenvalues are complex conjugate to each other, then a real-valued 2×2 matrix with the prescribed eigenvalues and singular values does exist. To explain our recursive algorithm more plainly, we briefly review Horn's inductive proof in Section 3. In particular, we point out how and where the splitting can take place and, hence, the recursive algorithm can be applied. To know the splitting exactly is a crucial step in our recursive algorithm. The matrix under construction carries a specific structure. If we had followed Horn's proof precisely, then the resulting algorithm would have become extremely expensive. We discuss this issue in Section 4. In particular, we argue that the required structure of matrices in Horn's proof is entirely not necessary. This insight enables us to derive a very efficient algorithm. In Section 5 we give a symbolic example to demonstrate the flow of our recursive process. In Section 6 we report some interesting results from our numerical experiment. Specifically, we are able to construct nonsymmetric matrices that have the same eigenvalues and singular values as what those classical eigenvalue test matrices possess, including the most challenging Rosser matrix and the Wilkinson's matrices.

2. The 2×2 Case. The conditions in Theorem 1.1 for a 2×2 matrix A to have eigenvalues $\{\lambda_1, \lambda_2\}$ and singular values $\{\alpha_1, \alpha_2\}$ are

$$\left\{ \begin{array}{rcl} |\lambda_1| & \leq & \alpha_1, \\ |\lambda_1||\lambda_2| & = & \alpha_1\alpha_2. \end{array} \right. \tag{2.1}$$

It follows that $\alpha_2 \leq |\lambda_2| \leq |\lambda_1| \leq \alpha_1$ and $|\lambda_1|^2 + |\lambda_2|^2 \leq \alpha_1^2 + \alpha_2^2$. In this section, we explain in details how a 2×2 matrix can be constructed. Such a construction will serve as the building block in our recursive algorithm.

Horn's proof starts with the simple fact that the triangular matrix

$$A = \begin{bmatrix} \lambda_1 & \mu \\ 0 & \lambda_2 \end{bmatrix} \tag{2.2}$$

has singular value $\{\alpha_1, \alpha_2\}$ if and only if

$$\mu = \sqrt{\alpha_1^2 + \alpha_2^2 - |\lambda_1|^2 - |\lambda_2|^2}. \tag{2.3}$$

His proof for the higher dimensional case is then completed by induction which we will explain in the next section.

In the case when the eigenvalues or singular values are nearly equal to each other, the calculation of μ according to (2.3) is unstable when the magnitude of α_i or λ_i is large. Instead, we suggest to compute μ through the equivalent formula

$$\mu = \left\{ \begin{array}{rl} 0, & \text{if } |(\alpha_1 - \alpha_2)^2 - (|\lambda_1| - |\lambda_2|)^2| \leq \epsilon \\ \sqrt{|(\alpha_1 - \alpha_2)^2 - (|\lambda_1| - |\lambda_2|)^2|}, & \text{otherwise} \end{array} \right.$$

where ϵ is the machine accuracy.

Horn's proof, while valid over the complex domain, has one shortcoming in that the matrix constructed is generally complex-valued even if the two eigenvalues are complex-conjugate. It appears difficult and sometimes impossible to convert the triangular matrix A in (2.2) by unitary similarity transformation into a real matrix. Suppose the given eigenvalues appear in complex-conjugate pairs, we often are interested in constructing a real-valued matrix. To our knowledge, we are not aware of any theory showing that such a real matrix exists. It is perhaps still an open problem. We are able to show, however, that such a construction for 2×2 matrices is possible.

Consider the matrix

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \tag{2.4}$$

where a, b, c, and d are real numbers to be determined. For the matrix A to have eigenvalues $\{\lambda_1, \lambda_2\}$ and singular values $\{\alpha_1, \alpha_2\}$, the entries of A must satisfy the following equations:

$$\left\{ \begin{array}{rcl} a+d & = & \lambda_1 + \lambda_2, \\ ad - bc & = & \lambda_1 \lambda_2, \\ a^2 + b^2 + c^2 + d^2 & = & \alpha_1^2 + \alpha_2^2, \\ (ad-bc)^2 & = & \alpha_1^2 \alpha_2^2. \end{array} \right. \tag{2.5}$$

Obviously, the last equation is equivalent to the second equation according to (2.1). Assume that $\nu := \lambda_1 + \lambda_2$ is a real number, i.e., assume that the given eigenvalues λ_1 and λ_2 are either real or complex-conjugate. Upon substitution by $d = \nu - a$, we obtain a reduced system of equations:

$$\left\{ \begin{array}{rcl} a(\nu - a) - bc & = & \lambda_1 \lambda_2, \\ a^2 + b^2 + c^2 + (\nu - a)^2 & = & \alpha_1^2 + \alpha_2^2. \end{array} \right. \tag{2.6}$$

It follows that

$$b = c \pm \sqrt{\alpha_1^2 + \alpha_2^2 - \lambda_1^2 - \lambda_2^2}. \tag{2.7}$$

Note that the quantity under the radical in the above is nonnegative. It now boils down to determining whether the equation

$$c^2 \pm c\sqrt{\alpha_1^2 + \alpha_2^2 - \lambda_1^2 - \lambda_2^2} + (a^2 - \nu a + \lambda_1 \lambda_2) = 0 \tag{2.8}$$

has real solutions a and c. Since all coefficients in (2.8) are real, a necessary condition for c to be real is that the discriminant is nonnegative, i.e., it requires to find a real value a so that

$$\alpha_1^2 + \alpha_2^2 - \lambda_1^2 - \lambda_2^2 - 4(a^2 - \nu a + \lambda_1 \lambda_2) \geq 0. \tag{2.9}$$

From (2.1) and our assumption, we know that the products $\lambda_1\lambda_2$ and $\alpha_1\alpha_2$ differ by at most a negative sign. Depending upon the sign in

$$\lambda_1\lambda_2 = \pm\alpha_1\alpha_2, \tag{2.10}$$

we thus may rewrite the left-hand side of (2.9) as

$$-4(a-\frac{1}{2}\nu)^2 + (\alpha_1 \mp \alpha_2)^2.$$

Obviously, if we choose

$$a = \frac{1}{2}\nu, \tag{2.11}$$

then (2.9) is satisfied. In that case, we may choose

$$c = \frac{\mp\sqrt{\alpha_1^2+\alpha_2^2-\lambda_1^2-\lambda_2^2} \pm (\alpha_1 \diamond \alpha_2)}{2} \tag{2.12}$$

where the first signs can be arbitrary and the last sign $\diamond$ is minus or plus depends upon the sign in (2.10).

In summary, we have shown the existence of a 2×2 matrix with prescribed eigenvalues and singular values, provided the conditions in (2.1) are met. More specifically, we have shown that the matrix can be real-valued, if the given eigenvalues are a complex-conjugate pair.

3. A Recursive Algorithm. Assuming that the two sets of numbers $\{\lambda_i\}$ and $\{\alpha_i\}$ are arranged in the order as in (1.1) and satisfy the conditions in (1.2) and (1.3), Horn proves the existence of a matrix with eigenvalues $\{\lambda_i\}$ and singular values $\{\alpha_i\}$ by treating the case of zero singular values separately followed by the mathematical induction. In this note we want to point out that his induction argument is in fact implementable as a recursive algorithm.

In order to explain the recursive algorithm more plainly, we briefly outline Horn's proof below.

Case 1. Suppose $\alpha_i > 0$ for all $i = 1, \ldots, n$. Then, from the condition (1.3), we know that $\lambda_i \neq 0$ for all i. Let

$$\left\{ \begin{array}{lcl} \sigma_1 & := & \alpha_1, \\ \sigma_i & := & \sigma_{i-1}\frac{\alpha_i}{|\lambda_i|}, \quad \text{for } i = 2, \ldots, n-1. \end{array} \right. \tag{3.1}$$

Assume the minimum value of these σ_i, denoted by

$$\sigma := \min_{1\leq i\leq n-1} \sigma_i, \tag{3.2}$$

occurs at the index j. By the condition (1.2), it follows that $|\lambda_1| \leq \sigma_i$ for all i. Hence, $|\lambda_1| \leq \sigma$. Define

$$\rho := \frac{|\lambda_1 \lambda_n|}{\sigma}. \tag{3.3}$$

By the definition of σ and the condition (1.3), we also have $\alpha_n \sigma \leq \alpha_n \alpha_1 \prod_{i=2}^{n-1} \frac{\alpha_i}{|\lambda_i|} = |\lambda_1 \lambda_n|$. It follows that

$$\alpha_n \leq \rho \leq |\lambda_n| \leq |\lambda_1| \leq \sigma \leq \alpha_1. \tag{3.4}$$

The relationship in (3.4) implies that the two conditions in (2.1) are satisfied by the two pairs of numbers $\{\lambda_1, \lambda_n\}$ and $\{\sigma, \rho\}$. From the discussion in the preceding section, there exist a number $\mu \in R$ and unitary matrices $U_0, V_0 \in C^{2\times 2}$ such that

$$U_0 \begin{bmatrix} \sigma & 0 \\ 0 & \rho \end{bmatrix} V_0^* = A_0 = \begin{bmatrix} \lambda_1 & \mu \\ 0 & \lambda_n \end{bmatrix}. \tag{3.5}$$

Furthermore, it can be verified that each of the two sets of numbers,

$$\left\{ \begin{array}{ccccccc} \sigma & \geq & |\lambda_2| & \geq & \dots & \geq & |\lambda_j| \\ \alpha_1 & \geq & \alpha_2 & \geq & \dots & \geq & \alpha_j \end{array} \right. \tag{3.6}$$

and

$$\left\{ \begin{array}{ccccccc} |\lambda_{j+1}| & \geq & \dots & \geq & |\lambda_{n-1}| & \geq & \rho \\ \alpha_{j+1} & \geq & \dots & \geq & \alpha_{n-1} & \geq & \alpha_n, \end{array} \right. \tag{3.7}$$

satisfies the Weyl-Horn conditions (1.2) and (1.3) by itself. Upon applying the induction hypothesis, there exist unitary matrices U_1, $V_1 \in C^{j\times j}$, U_2, $V_2 \in C^{(n-j)\times(n-j)}$, and *triangular* matrices A_1, A_2 such that

$$U_1 \begin{bmatrix} \alpha_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & & 0 \\ \vdots & \ddots & & \\ 0 & 0 & \dots & \alpha_j \end{bmatrix} V_1^* = A_1 = \begin{bmatrix} \sigma & \times & \times & \dots & \times \\ 0 & \lambda_2 & & & \times \\ & & & & \\ \vdots & & & \ddots & \\ 0 & 0 & & & \lambda_j \end{bmatrix},$$

(3.8)

$$U_2 \begin{bmatrix} \alpha_{j+1} & 0 & \dots & 0 \\ 0 & \alpha_{j+2} & & 0 \\ \vdots & & \ddots & \\ 0 & 0 & \dots & \alpha_n \end{bmatrix} V_2^* = A_2 = \begin{bmatrix} \lambda_{j+1} & \times & \dots & \times & \times \\ 0 & \lambda_{j+2} & & & \times \\ \vdots & & \ddots & & \vdots \\ & & & \lambda_{n-1} & \times \\ 0 & 0 & \dots & 0 & \rho \end{bmatrix}.$$

In the above, the symbol $\times$ stands for some appropriate numbers. Horn claims that the block diagonal matrix

$$\begin{bmatrix} A_1 & \bigcirc \\ \bigcirc & A_2 \end{bmatrix}$$

is unitarily equivalent to the triangular matrix

$$\begin{bmatrix}
\lambda_2 & \times & \dots & \times & \times & & & & & \\
0 & & & & \times & & & & & \\
\vdots & & \ddots & & \vdots & & & \bigcirc & & \\
 & & & \lambda_j & \times & & & & & \\
0 & & \dots & 0 & \sigma & 0 & & & & \\
0 & 0 & \dots & 0 & 0 & \rho & \times & \times & \dots & \times \\
 & & & & & & \lambda_{j+1} & & & \times \\
 & & \bigcirc & & & & & & & \\
 & & & & & \vdots & & & \ddots & \\
 & & \dots & & & 0 & 0 & & & \lambda_{n-1}
\end{bmatrix} \tag{3.9}$$

which, by applying (3.5), is unitarily equivalent to the triangular matrix

$$\begin{bmatrix}
\lambda_2 & \times & \dots & \times & \times & & & & & \\
0 & & & & \times & & & & & \\
\vdots & & \ddots & & \vdots & & & \bigcirc & & \\
 & & & \lambda_j & \times & & & & & \\
0 & & \dots & 0 & \lambda_1 & \mu & & & & \\
0 & 0 & \dots & 0 & 0 & \lambda_n & \times & \times & \dots & \times \\
 & & & & & & \lambda_{j+1} & & & \times \\
 & & \bigcirc & & & & & & & \\
 & & & & & \vdots & & & \ddots & \\
 & & \dots & & & 0 & 0 & & & \lambda_{n-1}
\end{bmatrix}.$$

Case 2. Suppose $\alpha_1 \geq \ldots \geq \alpha_k > \alpha_{k+1} = \ldots \alpha_n = 0$ for some k. It follows from the condition (1.2) that $\lambda_{k+1} = \ldots = \lambda_n = 0$. Let m be the largest integer for which $\lambda_m \neq 0$ but $\lambda_i = 0$ for all $i > m$. Clearly, $m \leq k$. Define

$$\beta := \frac{\prod_{i=1}^{m} |\lambda_i|}{\prod_{i=1}^{m-1} \alpha_i}. \tag{3.10}$$

It follows from (1.2) that $0 < \beta \leq \alpha_m$. Note that this set of numbers,

$$\left\{ \begin{array}{ccccccc} |\lambda_1| & \geq & \dots & & & \geq & |\lambda_m| \\ \alpha_1 & \geq & \dots & \geq & \alpha_{m-1} & \geq & \beta, \end{array} \right. \tag{3.11}$$

satisfies the Weyl-Horn conditions again. By induction hypothesis, therefore, there exist unitary matrices $U_3, V_3 \in C^{m\times m}$ and an upper triangular matrix A_3 such that

$$U_3 \begin{bmatrix} \alpha_1 & 0 & \dots & 0 & 0 \\ 0 & \alpha_2 & & & 0 \\ \vdots & \ddots & & & \\ 0 & & & \alpha_{m-1} & 0 \\ 0 & 0 & \dots & 0 & \beta \end{bmatrix} V_3^* = A_3 = \begin{bmatrix} \lambda_1 & \times & \times & \dots & \times \\ 0 & \lambda_2 & & & \times \\ & & & & \\ \vdots & & & \ddots & \\ 0 & 0 & & & \lambda_m \end{bmatrix}.$$

Define

$$(3.12)\quad A = \left[\begin{array}{cccc|ccccc|ccc} & V_3^* A_3 V_3 & & & & & \bigcirc & & & & \bigcirc & \\ \hline 0 & \dots & 0 & \gamma & 0 & 0 & \dots & & 0 & 0 & \dots & 0 \\ 0 & & 0 & 0 & \alpha_{m+1} & 0 & & & & & & \\ 0 & & & & 0 & & \ddots & & & & & \\ \vdots & & & & \vdots & & \ddots & & \vdots & & & \\ 0 & \dots & & 0 & 0 & & & \alpha_k & 0 & 0 & \dots & 0 \\ \hline & \bigcirc & & & & & \bigcirc & & & & \bigcirc & \end{array}\right]$$

with

$$(3.13)\quad \gamma := \sqrt{\alpha_m^2 - \beta^2}.$$

Then it is readily seen that A has eigenvalues $\{\lambda_1, \dots, \lambda_m, 0, \dots, 0\}$ and singular values $\{\alpha_1, \dots, \alpha_k, 0, \dots, 0\}$.

In both cases above, we see that the original inverse problem is reduced to inverse problems of smaller sizes that are guaranteed to be solvable according to the induction hypothesis. In the first case, the problem of size n is broken down two subproblems of sizes j and $n-j$, respectively. After finding the solutions A_1 and A_2 to the smaller problems, the two subproblems are then affixed together by working on a 2×2 submatrix. The 2×2 problem has an explicit solution from the discussion in the preceding section. More significantly, the eigenvalues and the singular values for the two subproblems are given explicitly by (3.6) and (3.7), respectively. Therefore, by repeating the argument, each of the two subproblems can further be downsized. In this way, the original problem is eventually solved by first *dividing* the problem into subproblems of blocks 2×2 or 1×1, and then by *conquering* these small blocks to build up the the original size. This divide and conquer process is similar to that occurred in the radix-2 fast Fourier transform. In the second case, the zero eigenvalues are first taken out of consideration. The remaining data (3.11) constitute a new subproblem that can be solved by the divide and conquer process discussed in the first case. Finally, the off-diagonal element γ defined in (3.13) and the structure of A in (3.12) take the original singular values into account.

In an environment that allows a subprogram to invoke itself recursively, we can exploit this feature by providing a routine that does exactly one step of the divide-and-conquer procedure described above. As a simple example, in MATLAB syntax, this recursive algorithm can be conveniently expressed as the program in Table 3.1. More details on the structure of the matrix under construction will be discussed in the next section. At this moment, note that the function **svd_eig** calls itself which results in further splitting. In the final return, the program produces a matrix A that has eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$ and singular values $\{\alpha_1, \ldots, \alpha_n\}$. Once A is constructed, any similarity transformation QAQ^* by an unitary matrix Q will maintain the same eigenvalues and singular values.

It should be pointed out that in the attached program we choose not to calculate the unitary matrices mentioned in the discussion. If desired, these matrices could be computed in a very efficiently way as we shall see in the next section.

4. The Matrix Structure. It is necessary to make one important remark concerning the structure of the matrix we intend to construct. In his inductive proof, Horn assumes that both the intermediate matrices A_1 and A_2 are upper triangular matrices and that the diagonal entries are arranged in a certain order. See (3.8) and (3.9). Horn has never been specific on how these assumptions can be satisfied, though their validity can be seen from the Schur decomposition theorem. In practice, however, it turns out that these assumptions cannot be achieved simply by permutations. It, in fact, involves a quite complicated procedure to rearrange the diagonal entries by unitary similarity transformations while maintaining the upper triangular structure. This procedure of rearrangement is one order more expensive than the divide and conquer algorithm described above.

One important contribution in our study is that the triangular structure is entirely unnecessary. The matrix A produced from our algorithm is generally not triangular. Unlike Horn's proof, we do not require to first rearrange the diagonal entries and then perform the unitarily equivalent transformation to the middle 2×2 block. Instead, we modify the first and the last rows and columns of the corresponding block diagonal

```
function [A]=svd_eig(alpha,lambda);
n = length(alpha);
if n == 1                              % The 1 by 1 case
   A = [lambda(1)];
elseif n == 2                     % The 2 by 2 case
   [U,V,A] = two_by_two(alpha,lambda);
else                                   % Check zero singular values
   tol = n*alpha(1)*eps;
   k = sum(alpha > tol); m = sum(abs(lambda) > tol);
   if k == n                           % Case 1
      j = 1; s = alpha(1); temp = s;
      for i = 2:n-1
         temp = temp*alpha(i)/abs(lambda(i));
         if temp < s, j = i; s = temp; end
      end
      rho = abs(lambda(1)*lambda(n))/s;
      [U0,V0,A0] = two_by_two([s;rho],[lambda(1);lambda(n)]);
      [A1] = svd_eig(alpha(1:j),[s;lambda(2:j)]);
      [A2] = svd_eig(alpha(j+1:n),[lambda(j+1:n-1);rho]);
      A = [A1,zeros(j,n-j);zeros(n-j,j),A2];
      Temp = A;
         A(1,:)=U0(1,1)*Temp(1,:)+U0(1,2)*Temp(n,:);
         A(n,:)=U0(2,1)*Temp(1,:)+U0(2,2)*Temp(n,:);
      Temp = A;
         A(:,1)=V0(1,1)*Temp(:,1)+V0(1,2)*Temp(:,n);
         A(:,n)=V0(2,1)*Temp(:,1)+V0(2,2)*Temp(:,n);
   else                                % Case 2
      beta = prod(abs(lambda(1:m)))/prod(alpha(1:m-1));
      [U3,V3,A3] = svd_eig([alpha(1:m-1);beta],lambda(1:m));
      A = zeros(n); A(1:m,1:m) = V3'*A3*V3;
      for i = m+1:k, A(i,i+1) = alpha(i); end
      A(m,m+1) = sqrt(abs(alpha(m)^2-beta^2));
   end
end
```

TABLE 3.1
A MATLAB program for the recursive algorithm.

matrix $\begin{bmatrix} A_1 & \bigcirc \\ \bigcirc & A_2 \end{bmatrix}$ according to entries in the 2×2 unitary equivalence transformation matrices U_0 and V_0. More precisely, let the entries of U_0 and V_0 be denoted by $U_0 = [u_{0,st}]$ and $V_0 = [v_{0,st}]$, respectively. Define

$$U := \begin{bmatrix} u_{0,11} & 0 & u_{0,12} \\ 0 & I_{n-1} & 0 \\ u_{0,21} & 0 & u_{0,22} \end{bmatrix} \begin{bmatrix} U_1 & 0 \\ 0 & U_2 \end{bmatrix}, \tag{4.1}$$

$$V := \begin{bmatrix} v_{0,11} & 0 & v_{0,12} \\ 0 & I_{n-1} & 0 \\ v_{0,21} & 0 & v_{0,22} \end{bmatrix} \begin{bmatrix} V_1 & 0 \\ 0 & V_2 \end{bmatrix}, \tag{4.2}$$

where I_{n-1} stands for the identity matrix of size $n-1$. We affix the two matrices A_1 and A_2 together by defining

$$A := U \begin{bmatrix} \alpha_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & & 0 \\ \vdots & & \ddots & \\ 0 & & & \alpha_n \end{bmatrix} V^*. \tag{4.3}$$

It is easy to see that A has the structure

$$A = \begin{bmatrix} \lambda_1 & \otimes & \dots & \otimes & \otimes & * & * & & & \mu \\ \otimes & & & & \times & 0 & 0 & & & * \\ \vdots & \ddots & & & \vdots & & & \bigcirc & & \\ & & & \lambda_{j-1} & \times & & & & & \\ \otimes & \times & \dots & \times & \lambda_j & & & & & * \\ * & 0 & \dots & 0 & 0 & \lambda_{j+1} & \times & \times & \dots & \otimes \\ * & & & & 0 & \times & \lambda_{j+1} & & & \otimes \\ & & \bigcirc & & & & & & & \\ & & & & & \vdots & & & \ddots & \\ 0 & * & \dots & * & * & \otimes & \otimes & & & \lambda_n \end{bmatrix}. \tag{4.4}$$

In the above, the symbol $\times$ represents unchanged, original entries from A_1 or A_2. The symbol $\otimes$ denotes entries of A_1 or A_2 that are modified by scalar multiplications. Note that if an entry at $\otimes$ was zero to begin with, then it remains zero after the scaling. Finally, we use $*$ to represent possible new entries that were originally zero. It is important to observe that the zero pattern of $*$ in the first(last) row is exactly the same as that of $\otimes$ in the last(first) row. A similarly observation holds for columns. Note also that the four corners of A follow from (3.5).

Clearly, the matrix A has singular values $\{\alpha_1, \ldots, \alpha_n\}$. We need to show that A has eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$. If the procedure works, formulas (4.1) and (4.2) also suggest an efficient way to compute the unitary matrices if so desired.

We now examine the structure of A more closely and explain why our algorithm works. In contrast to (3.8) where A_1 and A_2 are upper triangular matrices, we shall assume in our induction hypothesis that the unitary matrices U_i and V_i have produced A_i, $i = 1, 2$, with the following properties:

(P1) Diagonal entries of A_1 and A_2 are in the order

$$\sigma, \lambda_2, \ldots, \lambda_j,$$

and

$$\lambda_{j+1}, \ldots, \lambda_{n-1}, \rho,$$

respectively.

(P2) Each A_i is similar through permutations to a lower triangular matrix whose diagonal entries constitute the same set as the diagonal entries of A_i. (Thus, each A_i has precisely its own diagonal entries as its eigenvalues.) More specifically, each A_i has the following structure:

1. The first row and the last row have the same zero pattern except that the lower-left corner is always zero.
2. The first column and the last column have the same zero pattern except that the lower-left corner is always zero.

If we can show that the affixed matrix A, defined in (4.4), has exactly the same properties, the the induction principle implies that our algorithm is indeed a variation of Horn's original proof. Our implementation of making it operative then becomes quite remarkable.

The general proof can be best argued through the graph theory. We suggest the book [7] as a general reference. Instead of providing the heavy machinery involved, we provide a short-cut proof below. Consider the digraph G_i corresponding to $A_i = [a_{i,st}]$, $i = 1, 2$, where the directed edge from vertex s to vertex t is marked by the entry $a_{i,st}$. Then the property (P2) is equivalent to the following statement:

(P2') The graph G_i contains no cycles of length greater than 1.

1'. Whenever there is an edge outgoing from the first vertex, there is an edge outgoing from the last vertex, and vise versa.

2'. Whenever there is an edge ingoing into the first vertex, there is an edge ingoing into the last vertex, and vise versa.

Let P_i, $i = 1, 2$, be the permutation matrix such that $P_i^T A_i P_i$ is lower triangular. Let $P := \begin{bmatrix} P_1 & \bigcirc \\ \bigcirc & P_2 \end{bmatrix}$. Label the vertices of A as $\lambda_1, \ldots, \lambda_n$ where we identify λ_1 with the first vertex σ in A_1 and λ_n with the last vertex ρ in A_2. Since simultaneous permutations of rows and columns introduce isomorphic graphs, we conclude that $P^T A P$ is of the form

$$\left[\begin{array}{cccccccccccccccccccc}
\lambda^* & 0 & \ldots & 0 & 0 & & & & & & & & & & & & & & & \\
\times & \lambda^* & & & 0 & & & & & & & & & & & & & & & \\
\vdots & & \ddots & & \vdots & & & \bigcirc & & & & & \bigcirc & & & & & \bigcirc & & \\
\times & & & & 0 & & & & & & & & & & & & & & & \\
\times & \times & \ldots & \times & \lambda^* & & & & & & & & & & & & & & & \\
+_1 & +_1 & \ldots & +_1 & +_1 & \lambda_1 & 0 & \ldots & & 0 & \oplus_n & \oplus_n & \ldots & \oplus_n & \oplus_n & & & & & \\
\times & \times & \ldots & \times & \times & -_1 & \lambda^* & & & 0 & 0 & 0 & & 0 & \ominus_1 & & & & & \\
\vdots & & \ldots & & \vdots & \vdots & & \ddots & & \vdots & \vdots & \vdots & & \vdots & \vdots & & & \bigcirc & & \\
\times & \times & & & \times & -_1 & \times & & & 0 & 0 & 0 & & 0 & \ominus_1 & & & & & \\
\times & \times & \ldots & & \times & -_1 & \times & \ldots & \times & \lambda^* & 0 & 0 & \ldots & 0 & \ominus_1 & & & & & \\
0 & 0 & \ldots & & 0 & & & & & & \lambda_* & 0 & \ldots & 0 & 0 & & & & & \\
0 & 0 & \ldots & & 0 & & & & & & \times & \lambda_* & & & 0 & & & & & \\
\vdots & & \ldots & & \vdots & & & \bigcirc & & & \vdots & & \ddots & & \vdots & & & \bigcirc & & \\
0 & 0 & \ldots & & 0 & & & & & & \times & \times & & \lambda_* & 0 & & & & & \\
\oplus_1 & \oplus_1 & \ldots & \oplus_1 & \oplus_1 & & & & & & +_n & +_n & +_n & +_n & \lambda_n & & & & & \\
& & & & & \ominus_n & 0 & \ldots & 0 & 0 & \times & \times & \ldots & \times & -_n & \lambda_* & 0 & \ldots & 0 & 0 \\
& & & & & \ominus_n & 0 & \ldots & 0 & 0 & \times & \times & \ldots & \times & -_n & \times & \lambda_* & & & 0 \\
& & \bigcirc & & & \vdots & \vdots & & \vdots & \vdots & \vdots & \vdots & & & \vdots & \vdots & & \ddots & & \vdots \\
& & & & & \ominus_n & 0 & & 0 & 0 & \times & \times & & \times & -_n & \times & \times & & \lambda_* & 0 \\
& & & & & \ominus_n & 0 & & 0 & 0 & \times & \times & \ldots & \times & -_n & \times & \times & & \times & \lambda_*
\end{array}\right].$$

To facilitate the understanding, we have adopted some special symbols in the above to mark where the new edges connecting graphs G_1 and G_2 to form the graph for A are added:

- λ^* represents generically any value from the list $\{\lambda_2, \ldots, \lambda_j\}$. Likewise, λ_* represents any value from the list $\{\lambda_{j+1}, \ldots, \lambda_{n-1}\}$.
- $+_1$ indicates a possible (and the only possible) edge outgoing from the vertex 1. Consequently, according to (P2.1'), a corresponding new edge outgoing from the vertex n could occur only at $\oplus_1$. Similar meaning holds for $+_n$ and $\oplus_n$.
- $-_1$ indicates a possible (and the only possible) edge ingoing into the vertex 1. Consequently, according to (P2.2'), a corresponding new edge ingoing into the vertex n could occur only at $\ominus_1$. Similar meaning holds for $-_n$ and $\ominus_n$.

It should be clear from the structure of P^TAP that the matrix A indeed also enjoys properties (P1) and (P2). Our proof therefore is complete. Specifically, we have proved that the rearrangement as is required in Horn's proof is not necessary. We have proved that modifying the "border" of the matrix $\begin{bmatrix} A_1 & \bigcirc \\ \bigcirc & A_2 \end{bmatrix}$ as is done in our program will result in a matrix with eigenvalues $\{\lambda_1, \ldots, \lambda_n\}$ and singular values $\{\alpha_1, \ldots, \alpha_n\}$.

5. A Symbolic Example. The recursive process and the structure of the resulting matrices are quite complicated. It perhaps will be helpful if we illustrate the idea by tracing symbolically one possible scenario of our algorithm in details.

Consider the case $n = 6$. For clarity, we shall denote the splitting index j, the corresponding σ, and ρ (See (3.2) and (3.3)) at the ℓ-th step by j_ℓ, σ_ℓ and ρ_ℓ, respectively. Let each non-empty box in the diagram below represents a matrix to be constructed with values at the top row and the bottom row in each box as the corresponding eigenvalues and singular values, respectively. For reference, we also mark down on the right side of the down-pointing arrow the eigenvalues and the singular values of the associated 2×2 matrix that will be used to affix matrices together.

Suppose, for a certain set of eigenvalues $\lambda_1, \ldots, \lambda_6$ and singular values $\alpha_1, \ldots, \alpha_6$, that our algorithm results in splittings at $j_1 = 5$, $j_2 = 2$, and $j_3 = 1$. The data available for construction are as follows:

$$\boxed{\left\{\begin{array}{cccccc} \lambda_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 & \lambda_6 \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \alpha_6 \end{array}\right.}$$

$$j_1 = 5 \quad \Downarrow \quad \left\{\begin{array}{cc} \lambda_1 & \lambda_6 \\ \sigma_1 & \rho_1 \end{array}\right.$$

$$\boxed{\left\{\begin{array}{ccccc} \sigma_1 & \lambda_2 & \lambda_3 & \lambda_4 & \lambda_5 \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 \end{array}\right.} \quad \boxed{\left\{\begin{array}{c} \rho_1 \\ \alpha_6 \end{array}\right.}$$

$$j_2 = 2 \quad \Downarrow \quad \left\{\begin{array}{cc} \sigma_1 & \lambda_5 \\ \sigma_2 & \rho_2 \end{array}\right.$$

$$\boxed{\left\{\begin{array}{cc} \sigma_2 & \lambda_2 \\ \alpha_1 & \alpha_2 \end{array}\right.} \quad \boxed{\left\{\begin{array}{ccc} \lambda_3 & \lambda_4 & \rho_2 \\ \alpha_3 & \alpha_4 & \alpha_5 \end{array}\right.} \quad \boxed{\phantom{\begin{array}{c} x \\ x \end{array}}}$$

$$j_3 = 1 \quad \Downarrow \quad \left\{\begin{array}{cc} \lambda_3 & \rho_2 \\ \sigma_3 & \rho_3 \end{array}\right.$$

$$\boxed{\phantom{\begin{array}{c} x \\ x \end{array}}} \quad \boxed{\left\{\begin{array}{c} \sigma_3 \\ \alpha_3 \end{array}\right.} \quad \boxed{\left\{\begin{array}{cc} \lambda_4 & \rho_3 \\ \alpha_4 & \alpha_5 \end{array}\right.} \quad \boxed{\phantom{\begin{array}{c} x \\ x \end{array}}}$$

Our algorithm then builds up the 6×6 matrix step by step according to the order depicted in the following diagram. To stress how the 2×2 affixing matrix changes the entries, we use the same symbols as in (4.4) to indicate how an entry is modified.

$$\boxed{\begin{bmatrix} \lambda_1 & \otimes & 0 & 0 & \otimes & * \\ 0 & \lambda_2 & 0 & 0 & 0 & 0 \\ \otimes & 0 & \lambda_3 & 0 & \times & * \\ \otimes & 0 & \times & \lambda_4 & \times & * \\ 0 & \times & 0 & 0 & \lambda_5 & 0 \\ 0 & * & 0 & 0 & * & \lambda_6 \end{bmatrix}}$$

$$j_1 = 5 \quad \Uparrow \quad \begin{bmatrix} \sigma_1 & \times & 0 & 0 & \times & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 & 0 \\ \times & 0 & \lambda_3 & 0 & \times & 0 \\ \times & 0 & \times & \lambda_4 & \times & 0 \\ 0 & \times & 0 & 0 & \lambda_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & \rho_1 \end{bmatrix}$$

$$\boxed{\begin{bmatrix} \sigma_1 & \otimes & 0 & 0 & * \\ 0 & \lambda_2 & 0 & 0 & 0 \\ * & 0 & \lambda_3 & 0 & \otimes \\ * & 0 & \times & \lambda_4 & \otimes \\ 0 & * & 0 & 0 & \lambda_5 \end{bmatrix}} \quad \boxed{\begin{bmatrix} \rho_1 \end{bmatrix}}$$

$$j_2 = 2 \quad \Uparrow \quad \begin{bmatrix} \sigma_2 & \times & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 \\ 0 & 0 & \lambda_3 & 0 & \times \\ 0 & 0 & \times & \lambda_4 & \times \\ 0 & 0 & 0 & 0 & \rho_2 \end{bmatrix}$$

$$\boxed{\begin{bmatrix} \sigma_2 & * \\ 0 & \lambda_2 \end{bmatrix}} \quad \boxed{\begin{bmatrix} \lambda_3 & 0 & * \\ * & \lambda_4 & \otimes \\ 0 & 0 & \rho_2 \end{bmatrix}} \quad \boxed{\phantom{\begin{bmatrix} 0 \end{bmatrix}}}$$

$$j_3 = 1 \quad \Uparrow \quad \begin{bmatrix} \sigma_3 & 0 & 0 \\ 0 & \lambda_4 & \times \\ 0 & 0 & \rho_3 \end{bmatrix}$$

$$\boxed{\phantom{\begin{bmatrix} 0 \end{bmatrix}}} \quad \boxed{\begin{bmatrix} \sigma_3 \end{bmatrix}} \quad \boxed{\begin{bmatrix} \lambda_4 & * \\ 0 & \rho_3 \end{bmatrix}} \quad \boxed{\phantom{\begin{bmatrix} 0 \end{bmatrix}}}$$

Note that each matrix in the box, i.e., either A_1 or A_2 constructed in each step, is not necessarily triangular. A closer examination of each matrix, however, shows they possess the properties (P1) and (P2) described before. In particular, each A_i can be transformed into blocks of triangular matrices by permutations (Though this is still different from what Horn assumes in his proof.) For example, if we permute a $(1,6)$ permutation, i.e., if we swap the first and the sixth rows and columns of the top 6×6 matrix, we obtain the matrix

$$\begin{bmatrix} \lambda_6 & \times & 0 & 0 & \times & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 & 0 \\ \times & 0 & \lambda_3 & 0 & \times & \times \\ \times & 0 & \times & \lambda_4 & \times & \times \\ 0 & \times & 0 & 0 & \lambda_5 & 0 \\ \times & \times & 0 & 0 & \times & \lambda_1 \end{bmatrix}$$

whose 5×5 principal submatrix has exactly the same structure as that of the 5×5 matrix A_1 constructed in the previous step. Note that properties (P2.1) and (P2.2) ensure that no other structure along the "border" will be changed. The $\times$ at the $(1,5)$ position seems troublesome. However, if we further perform the $(1,5)$ permutation, we obtain the matrix

$$\begin{bmatrix} \lambda_5 & \times & 0 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 & 0 & 0 \\ \times & 0 & \lambda_3 & 0 & \times & \times \\ \times & 0 & \times & \lambda_4 & \times & \times \\ \times & \times & 0 & 0 & \lambda_6 & 0 \\ \times & \times & 0 & 0 & \times & \lambda_1 \end{bmatrix}$$

whose block triangular structure should be obvious by now.

6. Numerical Experiment. The algorithm described above can be used to generate test matrices with desired spectral properties. As it is written now, the algorithm will produce real matrices if all eigenvalues are real. However, when complex eigenvalues are present, the resulting matrix will be complex-valued, even if they appear in complex conjugate pairs.

Example 1. A classical challenge for eigenvalue algorithms is the 8×8 Rosser matrix R with integer elements

$$R = \begin{bmatrix} 611 & 196 & -192 & 407 & -8 & -52 & -49 & 29 \\ 196 & 899 & 113 & -192 & -71 & -43 & -8 & -44 \\ -192 & 113 & 899 & 196 & 61 & 49 & 8 & 52 \\ 407 & -192 & 196 & 611 & 8 & 44 & 59 & -23 \\ -8 & -71 & 61 & 8 & 411 & -599 & 208 & 208 \\ -52 & -43 & 49 & 44 & -599 & 411 & 208 & 208 \\ -49 & -8 & 8 & 59 & 208 & 208 & 99 & -911 \\ 29 & -44 & 52 & -23 & 208 & 208 & -911 & 99 \end{bmatrix}.$$

The matrix is symmetric, but has a double eigenvalue, three nearly equal eigenvalues, a zero eigenvalue, two dominant eigenvalues of opposite sign and a small nonzero eigenvalues. From MATLAB, the *computed* eigenvalues and singular values of R are

$$\lambda = \begin{bmatrix} -1.020049018429997e{+}03 \\ 1.020049018429997e{+}03 \\ 1.020000000000000e{+}03 \\ 1.019901951359278e{+}03 \\ 1.000000000000001e{+}03 \\ 9.999999999999998e{+}02 \\ 9.804864072152601e{-}02 \\ 4.851119506099622e{-}13 \end{bmatrix}, \alpha = \begin{bmatrix} 1.020049018429997e{+}03 \\ 1.020049018429996e{+}03 \\ 1.020000000000000e{+}03 \\ 1.019901951359279e{+}03 \\ 1.000000000000000e{+}03 \\ 9.999999999999998e{+}02 \\ 9.804864072162672e{-}02 \\ 1.054603342667098e{-}14 \end{bmatrix},$$

respectively. We notice that the MATLAB estimates the rank of R to be 7 due to the nearly zero singular value. The machine zero corresponding to a matrix A of size n is defined to be $n\epsilon\|A\|_2$ where the floating point relative accuracy ϵ is approximately 2.2204×10^{-16}.

Using our recursive algorithm for the above λ and α, we obtain the following matrix A which, for the convenience of running text, is displayed in only 5 digits:

$$A = \begin{bmatrix} 1.0200e{+}03 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.0200e{+}03 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.0200e{+}03 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.0199e{+}03 & 0 & 0 & 1.4668e{-}09 & 0 \\ 0 & 0 & 0 & 0 & 1.0000e{+}03 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.0000e{+}03 & 0 & 0 \\ 0 & 0 & 0 & -1.5257e{-}05 & 0 & 0 & 9.8049e{-}02 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1.4045e{-}07 & 0 \end{bmatrix}.$$

The computed eigenvalues and singular values of A are

$$\hat{\lambda} = \begin{bmatrix} -1.020049018429997e{+}03 \\ 1.020049018429997e{+}03 \\ 1.020000000000000e{+}03 \\ 1.019901951359278e{+}03 \\ 1.000000000000001e{+}03 \\ 9.999999999999998e{+}02 \\ 9.80486407215721e{-}02 \\ 0 \end{bmatrix}, \hat{\alpha} = \begin{bmatrix} 1.020049018429997e{+}03 \\ 1.020049018429997e{+}03 \\ 1.020000000000000e{+}03 \\ 1.019901951359279e{+}03 \\ 1.000000000000001e{+}03 \\ 9.999999999999998e{+}02 \\ 9.804864072162672e{-}02 \\ 0 \end{bmatrix},$$

respectively. It is interesting to note that the resulting A is not symmetric, yet the eigenvalues and singular values of A agree with those of R up to the machine accuracy.

Example 2. Wilkinson's matrices are symmetric and tridiagonal with pairs of nearly, but not exactly, equal eigenvalues. This is another

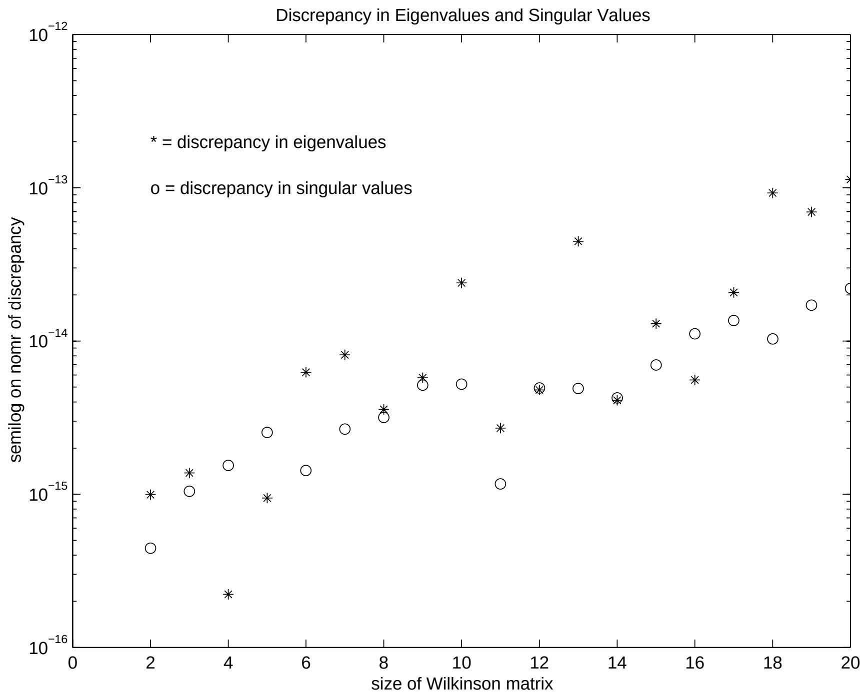

FIG. 6.1. *L_2 norm of discrepancy in eigenvalues and singular values.*

well known class of eigenvalue test matrices. We use the eigenvalues and singular values of these matrices from size 2 to 21 as the test data. In Figure 6.1 we report the discrepancy in eigenvalues and singular values between our constructed matrices and Wilkinson's matrices. Again, the discrepancy is in the range of machine accuracy. It is interesting to represent these 21×21 matrices by 3-D mesh surfaces in Figure 6.2. It is also interesting to note that the matrices constructed by our algorithm are nearly but not symmetric.

Example 3. The divide-and-conquer feature in our algorithm brings on fast computation. A closer examination of our algorithm suggests the overall cost should be in the order of $O(n^2)$. Using eigenvalues and singular values of 200 random matrices from size 2 to 200 as test data, we gauge the total cost by measuring the flops. The result is plotted in log-log scale in Figure 6.3. The linearity of that curve as well as its slops confirms that the cost is approximately $O(n^2)$.

7. Conclusion. We have modified Horn's original proof on the relationship between eigenvalues and singular values of a matrix. With the aid of programming languages that allow a subprogram to invoke itself recursively, we have developed a fast algorithm for reconstruct-

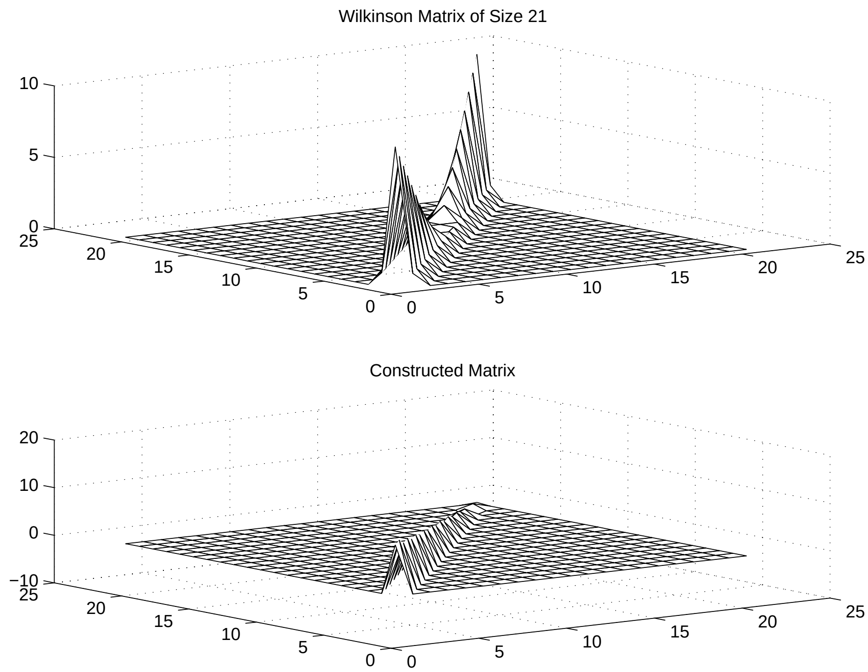

FIG. 6.2. *3-D mesh representation of* 21×21 *matrices*

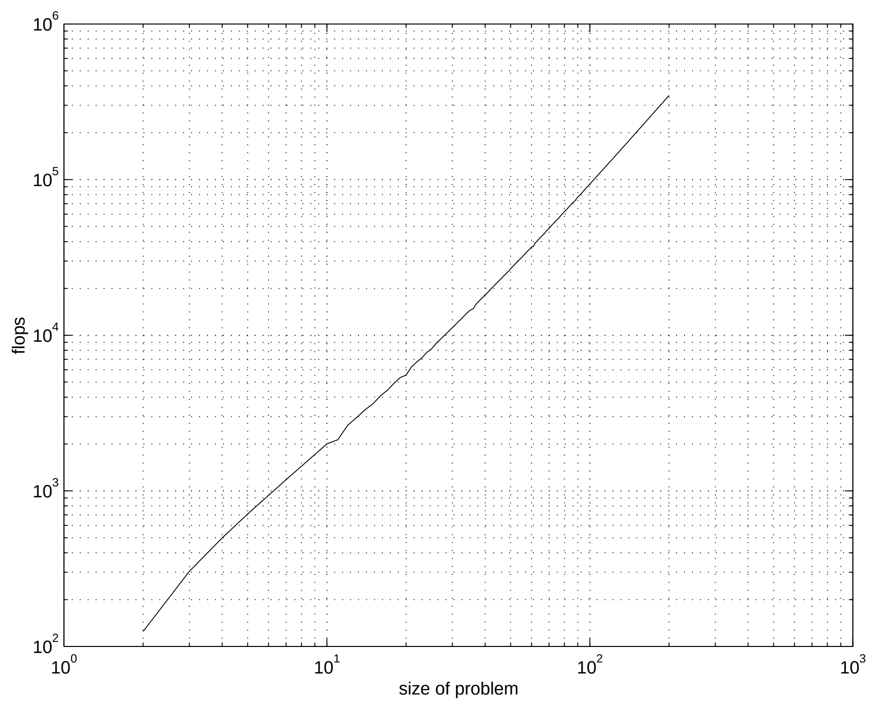

FIG. 6.3. *log-log plot of computational flops versus problem sizes*

ing matrices with prescribed eigenvalues and singular values. The cost of construction is approximately $O(n^2)$. The matrix being constructed usually is not symmetric and is complex-valued, if complex eigenvalues are present. Numerical experiment on some very challenging problems suggests that our method is quite robust.

REFERENCES

[1] M. T. Chu, Constructing a Hermitian matrix from its diagonal entries and eigenvalues, SIAM J. Matrix Anal. Appl., 16(1995), 207-217.

[2] M. T. Chu, On constructing matrices with prescribed singular values and diagonal elements, North Carolina State University, preprint, 1998.

[3] A. Horn, On the eigenvalues of a matrix with prescribed singular values, Proc. Amer. J. Math., 76(1954), 4-7.

[4] A. Horn, Doubly stochastic matrices and the diagonal of a rotation matrix, Amer. J. Math., 76(1956), 620-630.

[5] R. A. Horn and C. R. Johnson, Matrix Analysis, Cambridge University Press, New York, 1991.

[6] A. W. Marshall and I Olkin, Inequalities: Theory of Majorization and Its Applications, Vol. 143, in Mathematics in Sciences and Engineering, Academic Press, New York, New York, 1979.

[7] O. Melnikov, R. Tyshkevich, V. Yemelichev, and V. Sarvanov, Lectures on Graph Theory, Wissenschaftsverlag. Mannheim, 1994.

[8] L. Mirsky, Inequalities and existence theorems in the theory of matrices, J. Math. Anal. Appl., 9(1964), 99-118.

[9] F. Y. Sing, Some results on matrices with prescribed diagonal elements and singular values, Canad. Math. Bull., 19(1976), 89-92.

[10] R. C. Thompson, Singular values, diagonal elements, and convexity, SIAM J. Appl. Math., 32(1977), 39-63.

[11] H. F. Miranda and R. C. Thompson, Group majorization, the convex hulls of sets of matrices, and the diagonal element — singular value inequalities, Linear Alg. Appl., 199(1994), 131-141.

[12] H. Weyl, Inequalities between two kinds of eigenvalues of a linear transformation, Proc. Nat. Acad. Sci. U.S.A., 35(1949), 408-411.

[13] H. Zha and Z. Zhang, A note on constructing a symmetric matrix with specified diagonal entries and eigenvalues, BIT, 35(1995), 448-452.