Algorithm 694
A Collection of Test Matrices in MATLAB

NICHOLAS J. HIGHAM
University of Manchester

We present a collection of 45 parametrized test matrices. The matrices are mostly square, dense, nonrandom, and of arbitrary dimension. The collection includes matrices with known inverses or known eigenvalues, ill-conditioned or rank deficient matrices, and symmetric, positive definite, orthogonal, defective, involutary, and totally positive matrices. For each matrix we give a MATLAB M-file that generates it.

Categories and Subject Descriptors: G.1.3 [**Numerical Analysis**]: Numerical Linear Algebra

General Terms: Algorithms, Performance

Additional Key Words and Phrases: MATLAB, test matrix

1. INTRODUCTION

Numerical experiments are an indispensable part of research in numerical analysis. We do them for several reasons:

—to gain insight and understanding into an algorithm that is only partially understood theoretically;
—to verify the correctness of a theoretical analysis and to see if the analysis completely explains the practical behavior;
—to compare rival methods with regard to accuracy, speed, reliability, and so on; and
—to tune parameters in algorithms and codes and to test heuristics.

One of the difficulties in designing experiments is finding good test problems—ones which reveal extremes of behavior, cover a wide range of difficulty, are representative of practical problems, and (ideally) have known solutions. In many areas of numerical analysis good test problems have been identified, and several collections of such problems have been published. For

Part of this work was performed while the author was on leave in the Department of Computer Science at Cornell University.
Author's address: Department of Mathematics, University of Manchester, Manchester, M13 9PL, England.

example, collections are available in the areas of nonlinear optimization [28], linear programming [12, 26], ordinary differential equations [10], and partial differential equations [29].

Probably the most prolific devisers of test problems have been workers in matrix computations. Indeed, in the 1950s and 1960s it was common for a paper to be devoted to a particular test matrix: typically its inverse or eigenvalues would be obtained in closed form. An early survey of test matrices was given by Rutishauser [32]; most of the matrices he discusses come from continued fractions or moment problems. Two well-known books present collections of test matrices. Gregory and Karney [16] deal exclusively with the topic, while Westlake [36] gives an appendix of test matrices. In the two decades since these books appeared, several interesting matrices have been discovered (and in fact both books omit some worthy test matrices that were known at the time).

Our aim in this work has been to present an up-to-date, well documented and readily accessible collection of test matrices. With a few exceptions each of the 45 matrices satisfies the following requirements:

—It is a square matrix with one or more variable parameters, one of which is the dimension. Thus it is actually a parametrized family of matrices of arbitrary dimension.

—It is dense.

—It has some property which makes it of interest as a test matrix.

The first criterion is enforced because it is often desirable to explore the behavior of a numerical method as parameters such as the matrix dimension vary. The third criterion is somewhat subjective, and the matrices presented here represent the author's personal choice. Note that we have omitted plausible matrices that we thought not "sufficiently different" from others in the collection. Although all but one of our test matrices is usually real, those with an arbitrary parameter can be made complex by choosing a nonreal value for the parameter.

As well as their obvious application to research in matrix computations, the matrices presented here will be useful for constructing test problems in other areas, such as optimization (see, for example, Bartels and Joe [1]) and ordinary differential equations.

We mention three other collections of test matrices that complement ours. The Harwell–Boeing collection of sparse matrices, largely drawn from practical problems, is presented in Duff et al. [9]. Zielke [37] gives various parametrized rectangular matrices of fixed dimension with known generalized inverses. Demmel and McKenney [7] present a suite of Fortran 77 codes for generating random square and rectangular matrices with prescribed singular values, eigenvalues, band structure, and other properties. Our focus is primarily on nonrandom matrices but we include a class of random matrices (see `randsvd` in Section 3.1) which has some of the features of the Demmel and McKenney test set.

We present the matrices in the form of MATLAB M-files. (The reader unfamiliar with MATLAB should consult Coleman and Van Loan [5], Moler et al. [27], or Sigmon [33]). There are several reasons for this choice. One is that the matrices were collected, and have largely been used, in this form. Every time we come across an interesting matrix we write an M-file to generate it, and if the matrix turns out to satisfy the above requirements we add it to our library of M-files. When carefully written, M-files are self-documenting and so by writing an M-file we capture a matrix and its properties once and for all in a concise and easily manipulated form. (Compare this with maintaining a textual scrapbook with formulas, descriptions, references, and matrix instances.) Writing an M-file forces us to think of an appropriate and easily remembered name for the matrix. Where possible, we choose names eponymously since it is easier to remember, for example, "the Kahan matrix" than "Example 3.8." For portability reasons we restrict each M-file name to eight characters (since this is the limit in the MSDOS operating system, under which the PC version of MATLAB runs). We have written a routine (`matrix` in Section 3.2) that accesses the matrices by number rather than by name; this makes it easy to run experiments on the whole collection of matrices (with parameters other than the matrix dimension set to their default values.)

We do not give complete mathematical formulas for the elements of each matrix, since these are easily reconstructed from the MATLAB code (sometimes the formulas are given in comment lines). Nor do we give exhaustive descriptions of matrix properties, or proofs of these properties; instead we list a few key properties and give references where further details can be found.

This project stems from Higham [17] in which empirical observations made when using some of the matrices here led to several refinements to a matrix norm estimation algorithm. More recent experiments using this test collection (see Higham [19]) provided further insight into the algorithm.

The best way to understand and appreciate these matrices is to experiment with them in MATLAB. For example, look at their eigenvalues, singular values, and inverse (`eig`, `svd`, and `inv`). It is also interesting to examine pictures of the matrices. We have written a routine (see Section 3.2) that displays four pictures of a matrix in the format

`mesh(A)`	`mesh(pinv(A))`
`semilogy(svd(A))`	`fv(A)`

MATLAB's `mesh` command plots a three-dimensional mesh surface, by regarding the entries of a matrix as specifying heights above a plane. `pinv(A)` is the pseudo-inverse A^+ of A, which is the usual inverse when A is square and nonsingular. `semilogy(svd(A))` plots the singular values of A (ordered in decreasing size) on a logarithmic scale; the singular values are denoted by crosses, which are joined by a dashed line to emphasise the shape of the distribution. `fv(A)` plots the field of values (also called the numerical range), which is the set $\{x^*Ax/x^*x\colon\ x \in C^n\}$; the eigenvalues of A are plotted as crosses. `fv` is not a built-in MATLAB function; see Section 3.2 for details of this routine. It is perhaps not widely appreciated how useful a tool the field of values is for visualizing a matrix. For an example of how the field of values

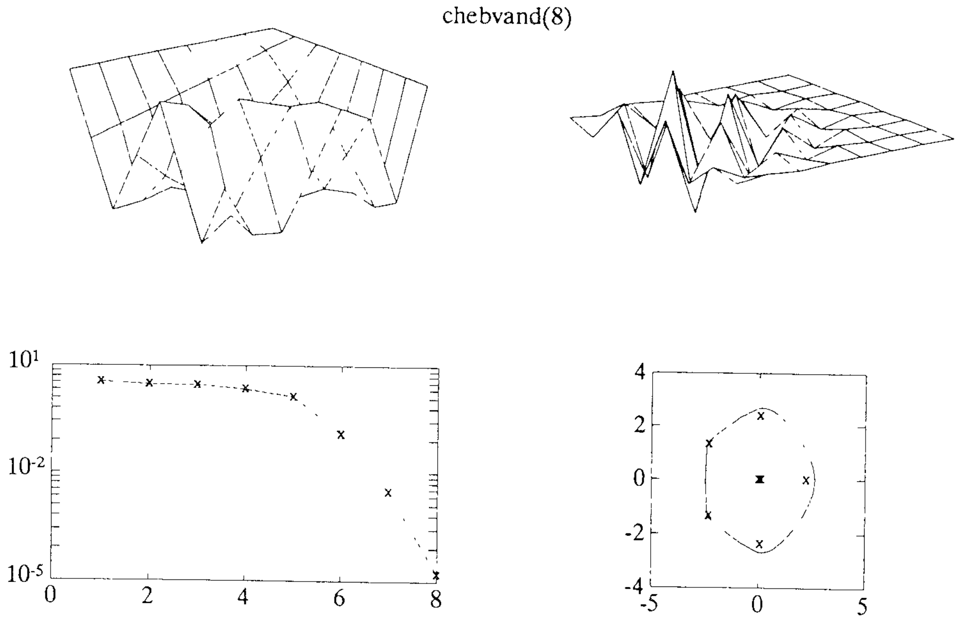

Figure 1

gives insight into the problem of finding a nearest normal matrix see Ruhe [31]. Figures 1–8 give examples of the pictures. A complete set of these pictures (for $n = 8$) is given in Higham [18] along with printouts of each matrix and its inverse for $n = 6$.

Our programming style is as follows. Each M-file `testmat` begins with comment lines that are displayed when the user types `help testmat`. Any further comments and references follow a blank line and so are not displayed by `help`. As far as possible, every routine sets default values for any parameters (other than the dimension) that are not specified, so that `A = testmat(n)` is usually valid. In general we have strived for conciseness, modularity, speed, and minimal use of temporary storage in our MATLAB codes. Hence, where possible, we have replaced `for` loops by matrix or vector constructs and have used calls to existing M-files. We check for errors in parameters in some, but not all, cases. A few of the test matrix routines do not properly handle the dimension $n = 1$ (for example, they halt with an error, or return an empty matrix). We decided not to add extra code for this case, since the routines are unlikely to be called with $n = 1$.

The matrices described here can be modified in various ways while still retaining some or all of their interesting properties. Among the many ways of constructing new test matrices from old are:

—similarity transformations $A \leftarrow X^{-1}AX$,

—orthogonal transformations $A \leftarrow UAV$, where $U^TU = V^TV = I$,

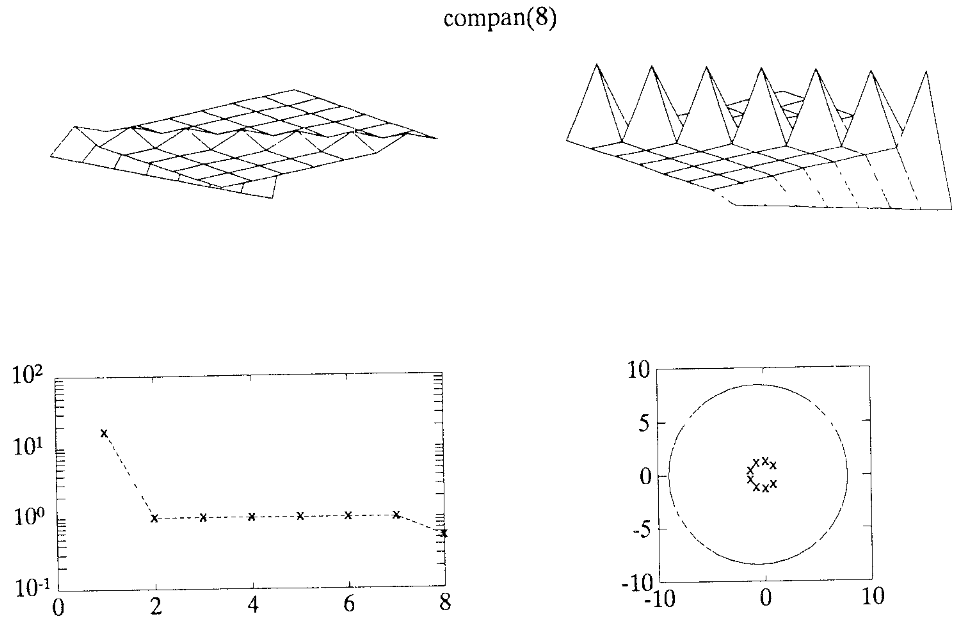

Figure 2

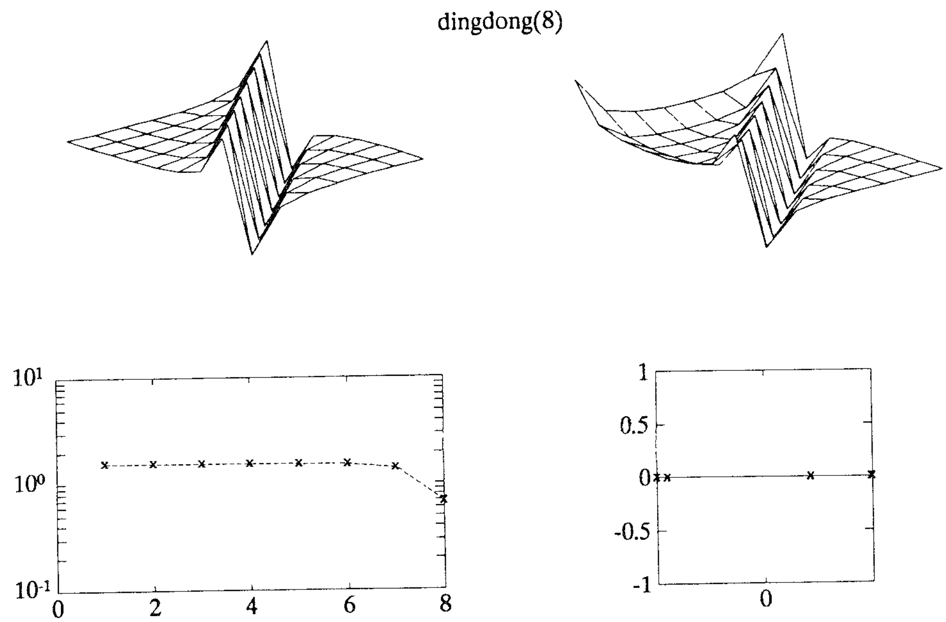

Figure 3

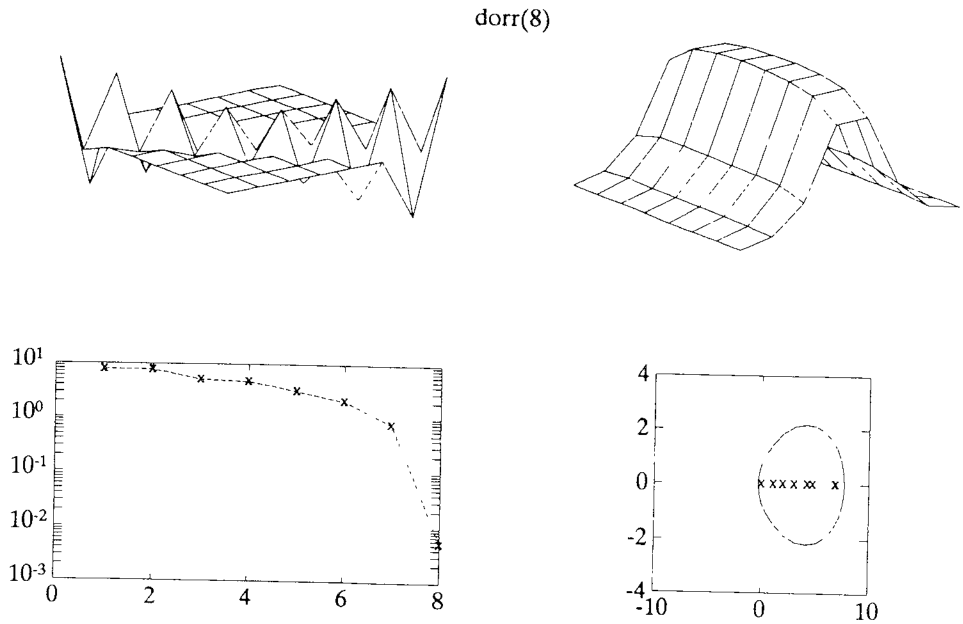

Figure 4

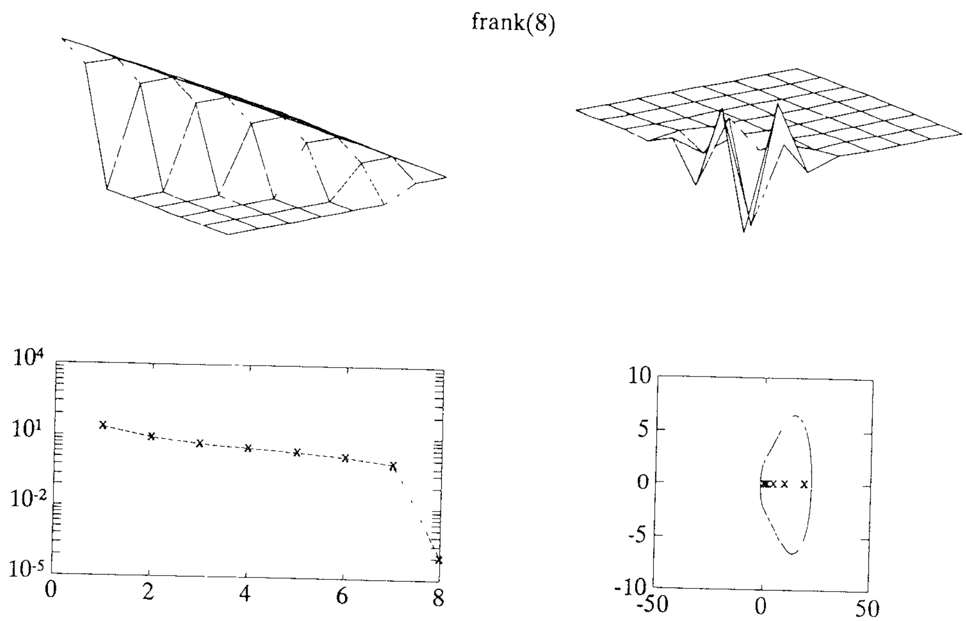

Figure 5

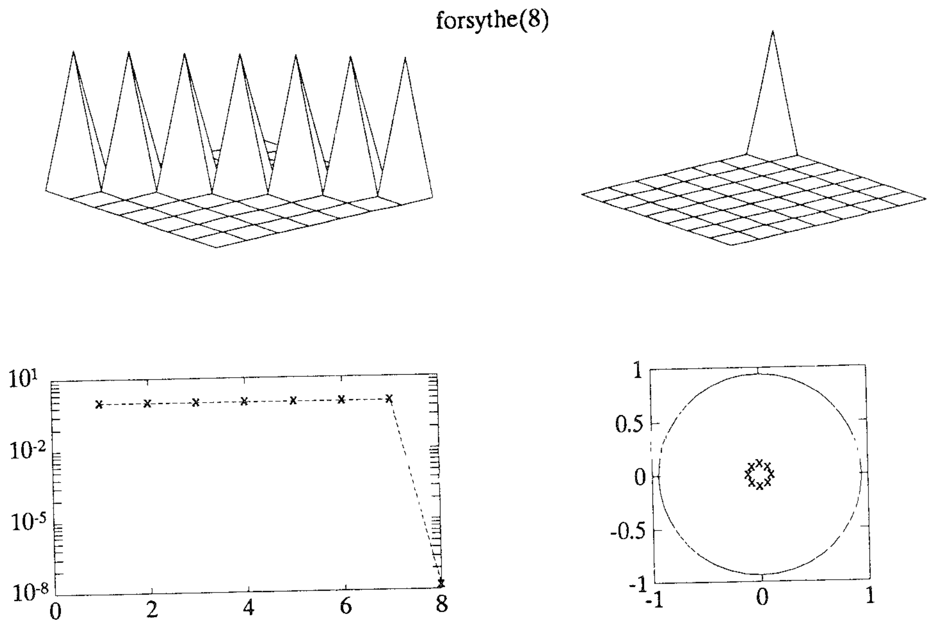

Figure 6

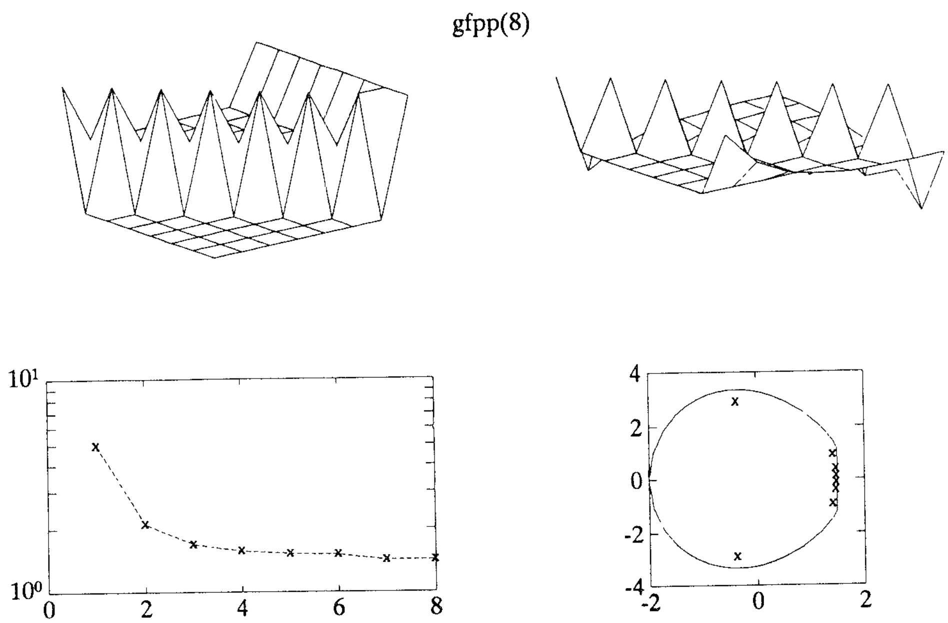

Figure 7

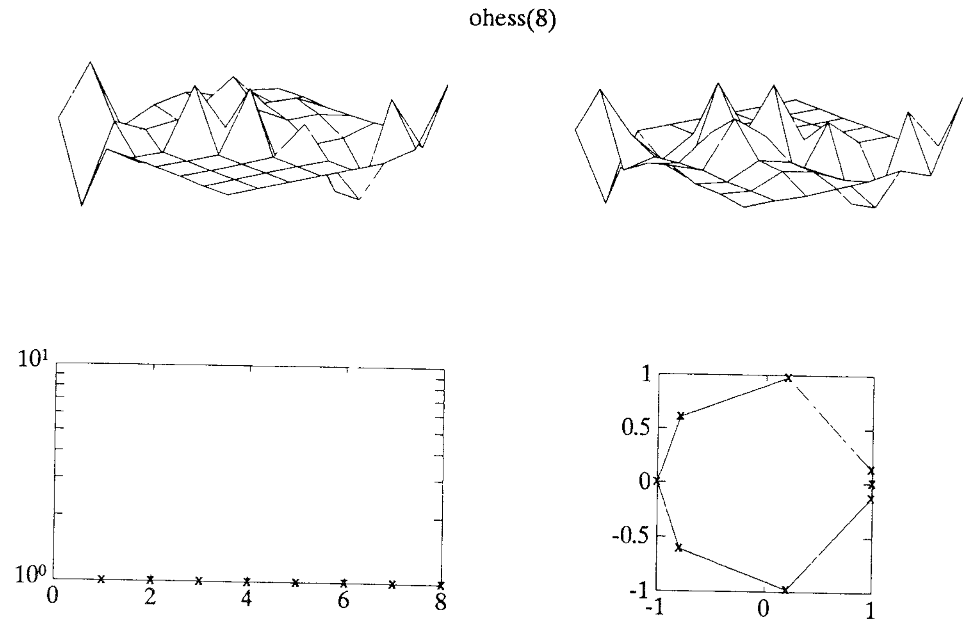

Figure 8

—Kronecker products $A \leftarrow A \otimes B$ or $B \otimes A$ (for which MATLAB has a routine `kron`), and

—powers $A \leftarrow A^k$.

For a discussion of these techniques, and others, see Gregory and Karney [16]. Techniques for obtaining a triangular, orthogonal, or symmetric positive definite matrix that is related to a given matrix include

—bandwidth reduction using orthogonal transformations (see `bandred` in Section 3.2) and

—LU, Cholesky, QR and polar decompositions.

For details of these techniques, see Golub and Van Loan [14].

Another way to generate a new matrix is to perturb an existing one. One approach is to add a random perturbation. Another is to round the matrix elements to a certain number of binary places; this can be done using the routine `chop` in Section 3.2.

The M-files in the test collection have been developed and tested using PC-MATLAB version 3.5a (March 1989) on '286'- and '386'-level PC machines. We anticipate no major problems in using the M-files with equivalent versions of MATLAB for other machines. The version number and date of the collection (both returned by `matrix(1)`) are version 1.2, May 30, 1990.

2. TEST MATRICES

Table I provides a summary of the properties of the test matrices. The column headings have the following meanings:

Table I. Summary of the Properties of the Test Matrices

Matrix	Inverse	Ill-cond	Rank	Symm	Pos Def	Orth	Eig
augment		√	√				
cauchy	√	√		√	√		
chebspec			√				√
chebvand		√				√	
chow			√				√
circul				√	√		√
clement	√		√	√			√
compan	√		√				√
condex		√					
cycol			√				
dingdong				√			√
dorr		√					
dramadah		√					
fiedler	√			√			√
forsythe	√	√					√
frank		√					√
gallery	√	√	√	√	√		√
gear			√				√
gfpp	√	√					
hadamard	√					√	√
hanowa							√
hilb	√	√		√	√		
invol	√	√					√
ipjfact		√		√			
jordan	√	√	√				√
kahan	√	√	√				
kms	√	√		√	√		
krylov		√					
lauchli		√					
lehmer	√			√	√		
lotkin	√	√					√
minij	√			√	√		√
moler	√	√		√	√		
ohess	√					√	√
orthog	√					√	√
pascal	√	√		√	√		√
pei	√	√		√	√		√
rando							
randsvd	√	√		√	√	√	
riemann							√
tridiag	√	√	√	√	√		√
triw	√	√					
vand	√	√					
wathen				√	√		√
wilk		√		√	√		√

Inverse: the inverse of the matrix is known explicitly.
Ill-cond: the matrix is ill-conditioned for some values of the parameters.
Rank: the matrix is rank-deficient for some values of the parameters (we exclude "trivial" examples such as vand which is singular if its vector argument contains repeated points).
Symm: the matrix is symmetric for some values of the parameters.

Pos Def: the matrix is symmetric positive definite for some values of the parameters.
Orth: the matrix is orthogonal, or a diagonal scaling of an orthogonal matrix, for some values of the parameters.
Eig: something is known about the eigensystem, ranging from bounds or qualitative knowledge of the eigenvalues to explicit formulas for some or all eigenvalues and eigenvectors.

Below we summarize further interesting properties possessed by some of the matrices. Recall that A is a Hankel matrix if the anti-diagonals are constant ($a_{ij} = r_{i+j}$), idempotent if $A^2 = A$, nilpotent if $A^k = 0$ for some k, involutary if $A^2 = I$, totally positive (nonnegative) if the determinant of every submatrix is positive (nonnegative), and a Toeplitz matrix if the diagonals are constant ($a_{ij} = r_{j-i}$). See Horn and Johnson [21] for further details of these matrix properties.

Some more properties of the matrices are as follows:

defective:	chebspec, gallery, gear, jordan
Hankel:	dingdong, hilb, ipjfact
Hessenberg:	chow, frank, ohess, randsvd
idempotent:	invol
involutary:	invol, orthog, pascal
nilpotent:	chebspec, gallery
rectangular:	chebvand, cycol, kahan, krylov, lauchli, rando, randsvd, triw, vand
Toeplitz:	chow, dramadah, kms
totally positive or totally nonnegative:	hilb, lehmer
tridiagonal:	clement, dorr, gallery, randsvd, tridiag, wilk
inverse of a tridiagonal matrix:	kms, lehmer, minij
triangular:	dramadah, jordan, kahan, pascal, triw

3. M-FILE SUMMARY

In this section we give a brief description of the M-files in the test matrix collection. The M-files that generate the test matrices are described in Section 3.1. In Section 3.2 we describe utility routines that are called by some of the test matrix M-files, as well as a few extra routines of interest for viewing and modifying the test matrices.

The comments in the M-files constitute the main documentation: they provide full details of the matrices and their properties, a description of the input and output parameters of each routine, and further references in addition to the selected ones given here.

In this summary we use MATLAB-style notation for submatrices and index ranges. Thus $A(p: q, r: s)$ denotes the submatrix of A comprising the intersection of rows p to q and columns r to s, and '$k = 1: n$' is equivalent to

'$k = 1, 2, \dots, n$'. We also use MATLAB notation for three special $n \times n$ matrices: `eye(n)` is the identity matrix, `ones(n)` is a matrix of ones, and `rand(n)` is a matrix of random numbers.

Whenever an M-file has n as an input parameter it generates an $n \times n$ matrix (`lauchli` is the sole exception). Routines `cycol`, `kahan`, `rando`, `randsvd` and `triw` allow n to be a 2-vector (n_1, n_2), in which case the matrix returned is $n_1 \times n_2$.

3.1 Summary of Test Matrix M-files

`augment(A)` is the square matrix

$$\begin{bmatrix} I_m & A \\ A^* & 0 \end{bmatrix}$$

of dimension $m + n$, where A is $m \times n$. It is the symmetric and indefinite coefficient matrix of the augmented system associated with a least squares problem minimize $\| Ax - b \|_2$. As a special case, `augment(n)` is the same as `augment(rand(`p, q`))` where $n = p + q$ and $p = \lfloor (n+1)/2 \rfloor$, that is, a random augmented matrix of dimension n is produced.

$C =$ `cauchy(`x, y`)`, where x and y are n-vectors, is the $n \times n$ matrix with $C(i, j) = 1/(x(i) + y(j))$. Explicit formulas are known for $\det(C)$ and the elements of C^{-1}. Reference: Knuth (1973, p. 36).

$C =$ `chebspec(`n, k`)` is a Chebyshev spectral differentiation matrix. For $k = 0$ ('no boundary conditions'), C is nilpotent with $C^n = 0$. For $k = 1$, C is nonsingular and well-conditioned, and its eigenvalues have negative real parts.

$C =$ `chebvand(`p`)`, where p is a vector, is the (primal) Chebyshev Vandermonde matrix based on the points p, i.e., $C(i, j) = T_{i-1}(p(j))$, where T_{i-1} is the Chebyshev polynomial of degree $i - 1$. `chebvand(`m, p`)` is a rectangular version of `chebvand(`p`)` with m rows.

$A =$ `chow(`n, α, δ`)` is a Toeplitz lower Hessenberg matrix $A = H(\alpha) + \delta I_n$, where $H(i, j) = \alpha^{i-j+1}$. $H(\alpha)$ has $p = \lfloor (n+1)/2 \rfloor$ zero eigenvalues, the rest being $4\alpha \cos(k\pi/(n+2))^2$, $k = 1{:}\ n - p$. Reference: Chow (1969).

$C =$ `circul(`v`)` is the circulant matrix whose first row is v. The eigensystem of C is known explicitly. Reference: Davis (1977).

`clement(`n, k`)` is a tridiagonal matrix with zero diagonal entries and eigenvalues equal to plus and minus the numbers $n - 1, n - 3, n - 5, \dots,$ (1 or 0). For $k = 0$ the matrix is unsymmetric, while for $k = 1$ it is symmetric. Reference: Clement (1959).

`compan(`p`)`, where p is an $(n + 1)$-vector, is the companion matrix whose first row is $-p(2{:}\ n + 1)/p(1)$.

`condex(`n, k, θ`)` is a 'counter-example' to the LINPACK condition estimator $(k = 1{:}3)$ or to a matrix norm estimator of Higham $(k = 4)$. References: Cline and Rew (1983), Higham (1988).

A = cycol(n, k) is a matrix of the form $A = B(1:n, 1:n)$ where $B = [C\ C\ C\ \ldots]$ and C = rand(n, k). Thus A's columns repeat cyclically, and A has rank at most k (which need not divide n).

A = dingdong(n) is the symmetric Hankel matrix with $A(i, j) = 0.5/(n - i - j + 1.5)$. The eigenvalues of A cluster around $\pi/2$ and $-\pi/2$.

dorr(n, θ) is a diagonally dominant, tridiagonal M-matrix which is ill-conditioned for small values of the parameter $\theta \geq 0$. Reference: Dorr (1971).

An anti-Hadamard matrix A is a matrix with elements 0 or 1 for which $\mu(A) = \| A^{-1} \|_F$ is maximal. A = dramadah(n, k) is a (0, 1) matrix for which $\mu(A)$ is relatively large, although not necessarily maximal. The available types are Toeplitz A ($k = 1$) with $|\det(A)| = 1$ and $\mu(A) > c(1.75)^n$, where c is a constant, and Toeplitz upper triangular A ($k = 2$). The inverses of both types have integer entries. Reference: Graham and Sloane (1984).

A = fiedler(c), where c is an n-vector, is the symmetric matrix with $A(i, j) = |c(i) - c(j)|$.

forsythe(n, α, λ) is equal to jordan(n, λ) except it has α in the $(n, 1)$ position. It has the characteristic polynomial $\det(A - tI) = (\lambda - t)^n - (-1)^n\alpha$.

F = frank(n, k) is the Frank matrix. It is upper Hessenberg with determinant 1. If $k = 1$, the elements are reflected about the anti-diagonal $(1, n)$ – $(n, 1)$. The eigenvalues of F may be obtained in terms of the zeros of the Hermite polynomials. They are positive and occur in reciprocal pairs. Thus if n is odd, 1 is an eigenvalue. F has $\lfloor n/2 \rfloor$ ill-conditioned eigenvalues—the smaller ones. Reference: Frank (1958).

gallery(n) is an $n \times n$ matrix with some special property. The values of n available are $n = 3$ (badly conditioned), $n = 4$ (the Wilson matrix—symmetric positive definite with integer inverse), $n = 5$ (an interesting eigenvalue problem: defective and nilpotent), $n = 8$ (the Rosser matrix, a classic symmetric eigenvalue problem), and $n = 21$ (Wilkinson's tridiagonal W_{21}^{+}, another eigenvalue problem).

A = gear(n, i, j) has ones on the sub- and super-diagonals, sign(i) in the $(1, |i|)$ position, sign(j) in the $(n, n + 1 - |j|)$ position, and zeros everywhere else. All eigenvalues are of the form $2\cos(a)$ and the eigenvectors are of the form $(\sin(w + a), \sin(w + 2a), \ldots, \sin(w + na))^T$. A is singular, can have double and triple eigenvalues, and can be defective. Reference: Gear (1969).

gfpp(T) is a matrix of order n for which Gaussian elimination with partial pivoting yields a growth factor 2^{n-1}. The parameter T is an arbitrary nonsingular upper triangular matrix of order $n - 1$. gfpp(T, c) sets all the multipliers to c ($0 \leq c \leq 1$) and gives growth factor $(1 + c)^{n-1}$. gfpp(n, c) (a special case) is the same as gfpp(eye($n - 1$), c) and generates the well-known example of Wilkinson. Reference: Higham and Higham (1989).

hadamard(n) is a Hadamard matrix of order n, that is, a matrix H with elements 1 or -1 such that $HH^T = nI$. An $n \times n$ Hadamard matrix with $n > 2$ exists only if n is divisible by 4. This M-file handles only the cases where n, $n/12$ or $n/20$ is a power of 2.

hanowa(n, d) is defined only for even $n = 2m$; it is the block 2×2 matrix

$$\begin{bmatrix} dI_m & -\operatorname{diag}(1:m) \\ \operatorname{diag}(1:m) & dI_m \end{bmatrix}.$$

It has complex eigenvalues $\lambda_k = d \pm ki$, $k = 1:m$.

hilb(n) is the Hilbert matrix, with elements $1/(i+j-1)$. hilb(n) is symmetric positive definite, totally positive, and a Hankel matrix.

invol(n) is an involutory and ill-conditioned matrix.

$A =$ ipjfact(n, k) is the Hankel matrix with $A(i,j) = (i+j)!$ (for $k = 0$) or $A(i,j) = 1/(i+j)!$ (for $k = 1$).

jordan(n, λ) is a Jordan block with eigenvalue λ.

kahan(n, θ) is an upper trapezoidal matrix involving a parameter θ, which has some interesting properties regarding estimation of condition and rank. Reference: Kahan (1966).

$A =$ kms(n, ρ) is the Kac–Murdock–Szego Toeplitz matrix with $A(i,j) = \rho^{|i-j|}$ (for real ρ). If ρ is complex, then the same formula holds except that elements below the diagonal are conjugated. A is positive definite if and only if $0 < |\rho| < 1$. A^{-1} is tridiagonal.

krylov(A, x, j) is the Krylov matrix $[x, Ax, A^2x, \ldots, A^{j-1}x]$, where A is an $n \times n$ matrix and x is an n-vector.

lauchli(n, μ) comprises μI_n augmented with a leading row of ones. It is a well-known example in least squares and other problems that indicates the dangers of forming A^TA. Reference: Lauchli (1961).

$A =$ lehmer(n) is the symmetric positive definite matrix with $A(i,j) = i/j$ for $j \geq i$. A is totally nonnegative and $n \leq \kappa_2(A) \leq 4n^2$. A^{-1} is tridiagonal.

lotkin(n) is the Hilbert matrix with its first row altered to all ones. It is unsymmetric, ill-conditioned, and has many negative eigenvalues of small magnitude. The inverse has integer entries and is known explicitly. Reference: Lotkin (1955).

$A =$ minij(n) is the symmetric positive definite matrix with $A(i,j) = \min(i,j)$. A^{-1} is tridiagonal.

moler(n, α) is the symmetric positive definite matrix U^TU where $U =$ triw(n, $-1, \alpha$). It has one small eigenvalue.

$H =$ ohess(n) is a real, random, orthogonal upper Hessenberg matrix. Alternatively, $H =$ ohess(x), where x is an arbitrary real n-vector ($n > 1$), constructs H nonrandomly using the elements of x as parameters.

orthog(n, k) is an orthogonal matrix or a diagonal scaling of an orthogonal matrix. There are six choices, selected by k.

pascal(n) is the Pascal matrix: a symmetric positive definite matrix with integer entries, made up from Pascal's triangle. Its inverse has integer entries. pascal(n, 1) is the involutary lower triangular Cholesky factor (up to signs of columns) of the Pascal matrix. pascal(n, 2) is a transposed and permuted version of pascal(n, 1) which is a cube root of the identity.

pei(n, α), where α is a scalar, is the symmetric matrix αI_n + ones(n). The matrix is singular for $\alpha = 0, -n$.

rando(n, k) is a random matrix whose elements are chosen with equal probability from $\{0,1\}$, $\{-1,1\}$, or $\{-1,0,1\}$, depending on k.

A = randsvd(n, κ, *mode*, *kl*, *ku*) is a (*kl*, *ku*)-banded random matrix with $\kappa_2(A) = \kappa$. The singular values are from one of five distributions selected by *mode*: one large singular value, one small singular value, geometrically distributed singular values, arithmetically distributed singular values, and random singular values with uniformly distributed logarithm. As a special case, if $\kappa < 0$ then a random, full, symmetric positive definite matrix is produced with $\kappa_2(A) = -\kappa$ and eigenvalues distributed according to *mode* (*kl* and *ku*, if present, are ignored).

A = riemann(n) is a matrix such that the Riemann hypothesis is true if and only if $\det(A) = O(n! n^{-1/2+\varepsilon})$ for every $\varepsilon > 0$. Bounds are known on the eigenvalues of A. Reference: Roesler (1986).

tridiag(x, y, z) is the tridiagonal matrix with subdiagonal x, diagonal y, and superdiagonal z. Alternatively tridiag(n, c, d, e), yields the Toeplitz tridiagonal matrix whose subdiagonal, diagonal, and superdiagonal elements are given by the scalars c, d and e, respectively.

triw(n, α, k) is the upper triangular matrix with ones on the diagonal and α on the first $k \geq 0$ superdiagonals.

vand(p) is a Vandermonde matrix based on the points in the vector p, i.e., $V(i,j) = p(j)^{i-1}$. vand(m, p) is a rectangular version of vand(p) with m rows.

A = wathen(n_x, n_y) is a random finite element matrix of dimension $3n_x n_y + 2n_x + 2n_y + 1$. A is precisely the "consistent mass matrix" for a regular $n_x \times n_y$ grid of 8-node (serendipity) elements in two space dimensions. A is symmetric positive definite and if D is the diagonal part of A then $0.25 \leq \lambda_i(D^{-1}A) \leq 4.5$ for all i. Reference: Wathen (1987).

[A, b] = wilk(n) is a matrix or system devised or discussed by Wilkinson. The values of n available are $n = 3$ (upper triangular system $Ux = b$ illustrating inaccurate solution), $n = 4$ (lower triangular system $Lx = b$, ill-conditioned), $n = 5$ (a scalar multiple of a submatrix of hilb(6), symmetric positive definite), and $n = 21$ (W_{21}^{+}, a tridiagonal eigenvalue problem).

3.2 Utility M-Files

B = bandred(A, kl, ku) is a matrix orthogonally equivalent to A with lower bandwidth kl and upper bandwidth ku (i.e., $B(i,j) = 0$ if $i > j + kl$ or $j > i + ku$). The reduction is performed using Householder transformations. This routine is called by randsvd.

chop(X, t) is the matrix obtained by rounding the elements of X to t significant binary places.

comp is a comparison matrix. comp(A) (often denoted by $M(A)$ in the literature) is A with each diagonal element replaced by its absolute value, and each off-diagonal element replaced by minus its absolute value. comp(A, 1) is the same except that each off-diagonal element is replaced by minus the absolute value of the largest element in absolute value in its row. However, if A is triangular comp(A, 1) is too.

fv(A, nk, θ_{max}) evaluates and plots the field of values (or numerical range) of the nk largest leading principal submatrices of A, using θ_{max} equally spaced angles in the complex plane. The eigenvalues of A are displayed as 'x'.

If $[v, \beta]$ = house(x) then $H = I - \beta v v^*$ is a Householder matrix such that $Hx = -\operatorname{sign}(x(1)) \| x \|_2 e_1$.

matrix(k, n) is the $n \times n$ instance of the matrix number k in the collection (including the matrices invhilb and magic provided with MATLAB), with all other parameters set to their default. Only those matrices which take an arbitrary dimension n are included (thus gallery is omitted, for example). matrix(k) is a string containing the name of the kth matrix. matrix(0) is the number of matrices, i.e., the upper limit for k. matrix(-1) returns the version number and date of the collection.

qmult(A) is QA where Q is a random real orthogonal matrix from the Haar distribution, of dimension the number of rows in A. This routine is called by randsvd. Reference: Stewart (1980).

rq(A, x) is the Rayleigh quotient of A and x, $x^*Ax/(x^*x)$. This routine is called by fv.

see(A) displays mesh(A), mesh(A^+), semilogy(svd(A)), and fv(A) in four subplot windows.

seqa(a, b, n) produces a row vector comprising n equally spaced numbers starting at a and finishing at b. If n is omitted then 10 points are generated.

seqcheb(n, k) produces a row vector of n points related to Chebyshev polynomials, T_n. For $k = 1$ the points are the zeros of T_n and for $k = 2$ they are the extrema of T_{n-1}.

seqm(a, b, n) produces a row vector comprising n logarithmically equally spaced numbers, starting at $a \neq 0$ and finishing at $b \neq 0$. If $ab < 0$ and $n > 2$ then complex results are produced.

show(X) displays X in '`format +` ' form, with '+', '−' and blank representing positive, negative and zero elements, respectively.

skew(A) is the skew-symmetric (Hermitian) part of A, $(A - A^*)/2$. It is the nearest skew-symmetric (Hermitian) matrix to A in both the 2- and the Frobenius norms.

sparse(A) plots the sparsity pattern of a matrix A, showing an 'x' for every nonzero element. sparse(A, tol) plots the elements bigger than tol in absolute value.

S = sparsify(A, p) is A with elements randomly set to zero ($S = S^*$ if $A = A^*$, i.e., symmetry is preserved). Each element has probability p of being zeroed. Thus on average $100p$ percent of the elements of A will be zeroed.

sub(A, i, j) is the principal submatrix $A(i:j, i:j)$. sub(A, i) is the leading principal submatrix of order i, $A(1:i, 1:i)$, if $i > 0$, and the trailing principal submatrix of order $|i|$ if $i < 0$.

symm(A) is the symmetric (Hermitian) part of A, $(A + A^*)/2$. It is the nearest symmetric (Hermitian) matrix to A in both the 2- and the Frobenius norms.

ACKNOWLEDGMENTS

I thank George Cybenko for help with the kms matrix, Nick Trefethen for help with chebspec, Peter Graves-Morris for suggesting ipjfact, Andy Wathen for providing wathen.m, Steve Vavasis for suggesting condex, and the referee for suggesting improvements to see.m. Thanks also to Gene Golub for pointing out Rutishauser (1966) and to Des Higham for his comments and suggestions (one of which is rando). Finally, I applaud the authors of MATLAB for producing such a wonderful instrument for teaching and research.

REFERENCES

1. BARTELS, R., AND JOE, B. On generating discrete linear l_∞ test problems. *SIAM J. Sci. Stat. Comput. 10* (1989), 550–561.
2. CHOW, T. S. A class of Hessenberg matrices with known eigenvalues and inverses. *SIAM Rev. 11* (1969), 391–395
3. CLEMENT, P. A. A class of triple-diagonal matrices for test purposes. *SIAM Rev. 1* (1959), 50–52.
4. CLINE, A. K., AND REW, R. K. A set of counter-examples to three condition number estimators. *SIAM J. Sci Stat. Comput. 4* (1983), 602–611.
5. COLEMAN, T. F., AND VAN LOAN, C. F. *Handbook for Matrix Computations*. Society for Industrial and Applied Mathematics, Philadelphia, 1988.
6. DAVIS, P. J. *Circulant Matrices*. Wiley, New York, 1977.
7. DEMMEL, J. W., AND MCKENNEY, A A test matrix generation suite. LAPACK Working Note #9, Courant Institute of Mathematical Sciences, New York, 1989.
8. DORR, F. W. An example of ill-conditioning in the numerical solution of singular perturbation problems. *Math. Comput. 25* (1971), 271–283.
9. DUFF, I. S., GRIMES, R. G., AND LEWIS, J. G. Sparse matrix test problems. *ACM Trans. Math. Softw. 15* (1989), 1–14.

10. ENRIGHT, W. H., AND PRYCE, J.D. Two FORTRAN packages for assessing initial value methods. *ACM Trans. Math. Softw. 13* (1987), 1-27.
11. FRANK, W. L. Computing eigenvalues of complex matrices by determinant evaluation and by methods of Danilewski and Wielandt. *J. Soc. Indust. Appl. Math. 6* (1958), 378-392.
12. GAY, D. M. Electronic mail distribution of linear programming test problems. Math. Program. Soc. COAL Newsl. (Dec. 1985), pp. 10-12.
13. GEAR, C. W. A simple set of test matrices for eigenvalue programs. *Math. Comput. 23* (1969), 119-125.
14. GOLUB, G. H., AND VAN LOAN, C. F. *Matrix Computations.* Second Ed., Johns Hopkins University Press, Baltimore, Md., 1989.
15. GRAHAM, R. L., AND SLOANE, N. J. A. Anti-Hadamard matrices. *Linear Algebra Appl. 62* (1984), 113-137.
16. GREGORY, R. T., AND KARNEY, D. L. *A Collection of Matrices for Testing Computational Algorithms.* Wiley, New York, 1969.
17. HIGHAM, N. J. FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation (Algorithm 674). *ACM Trans. Math. Softw. 14* (1988), 381-396.
18. HIGHAM, N. J. A collection of test matrices in MATLAB. Tech. Rep. 89-1025, Dept. of Computer Science, Cornell Univ., 1989; also Numerical Analysis Report No. 172, Dept. of Mathematics, Univ. of Manchester, 1989.
19. HIGHAM, N. J. Experience with a matrix norm estimator. *SIAM J. Sci. Stat. Comput. 11*, 4 (1990), 804-809.
20. HIGHAM, N. J., AND HIGHAM, D. J. Large growth factors in Gaussian elimination with pivoting. *SIAM J. Matrix Anal. Appl. 10*, 2 (1989), 155-164.
21. HORN, R. A., AND JOHNSON, C. R. *Matrix Analysis.* Cambridge University Press, 1985.
22. KAHAN, W. Numerical linear algebra. *Canadian Math. Bull. 9* (1966), 757-801.
23. KNUTH, D. E. *The Art of Computer Programming, Volume 1, Fundamental Algorithms.* Second Ed., Addison-Wesley, Reading, Mass., 1973.
24. LAUCHLI, P. Jordan-Elimination und Ausgleichung nach kleinsten Quadraten. *Numer. Math. 3* (1961), 226-240.
25. LOTKIN, M. A set of test matrices. M.T.A.C., 9 (1955), 153-161.
26. LUSTIG, I. J. An analysis of an available set of linear programming test problems. *Comput. Oper. Res. 16* (1989), 173-184.
27. MOLER, C. B., LITTLE, J. N., AND BANGERT, S. *PC-Matlab User's Guide* and *Pro-Matlab User's Guide.* The MathWorks Inc., South Natick, Mass., 1987.
28. MORÉ, J. J., GARBOW, B. S., AND HILLSTROM, K. E. Testing unconstrained optimization software. *ACM Trans. Math. Softw. 7* (1981), 17-41.
29. RICE, J. R., AND BOISVERT, R. E. *Solving Elliptic Problems using ELLPACK.* Springer Verlag, New York, 1985.
30. ROESLER, F. Riemann's hypothesis as an eigenvalue problem. *Linear Algebra Appl. 81* (1986), 153-198.
31. RUHE, A. Closest normal matrix finally found! *BIT 27* (1987), 585-598.
32. RUTISHAUSER, H. On test matrices. *Programmation en Mathématiques Numériques, Éditions Centre Nat. Recherche Sci., Paris, 165* (1966), 349-365.
33. SIGMON, K. *MATLAB Primer.* Dept. of Mathematics, Univ. of Florida, Gainesville, 1989.
34. STEWART, G. W. The efficient generation of random orthogonal matrices with an application to condition estimators. *SIAM J. Numer. Anal. 17* (1980), 403-409.
35. WATHEN, A. J. Realistic eigenvalue bounds for the Galerkin mass matrix. *IMA J. Numer. Anal. 7* (1987) 449-457.
36. WESTLAKE, J. R. *A Handbook of Numerical Matrix Inversion and Solution of Linear Equations.* Wiley, New York, 1968.
37. ZIELKE, G. Report on test matrices for generalized inverses. *Computing 36* (1986), 105-162.

Received November 1989; revised June 1990; accepted September 1990