AD-A063 473 NAVAL WEAPONS CENTER CHINA LAKE CALIF F/G 9/2

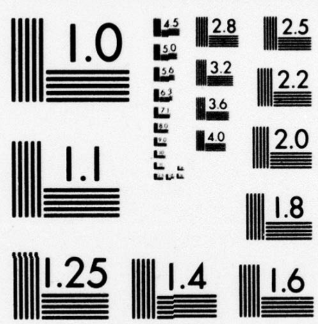
TEST PROCEDURE - LINEAR EQUATION ROUTINES.(U)

MAR 75 L W LUCAS

UNCLASSIFIED NWC-TM-2410 GIDEP-E095-0475 NL

1 OF 1
AD
A063473

END
DATE
FILMED
3-79
DDC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

95-0475

NWC Technical Memorandum 2410

LEVEL II

TEST PROCEDURE - LINEAR EQUATION ROUTINES

by

L. W. Lucas

Research Department

March 1975

35p.

NWC-TM-2410

GIDEP

E095-0475

Approved for public release; distribution unlimited. This is an informal report of the Naval Weapons Center and is not to be used as authority for action.

NAVAL WEAPONS CENTER

China Lake, California 93555

8 MAY 1978

NR

GOVERNMENT–INDUSTRY DATA EXCHANGE PROGRAM

GENERAL DOCUMENT SUMMARY SHEET

Please Type All Information – See Instructions on Reverse

1. ACCESS NUMBER: E095-0475	2. COMPONENT/PART NAME PER GIDEP SUBJECT THESAURUS: Computers, Software, NOC	
3. APPLICATION: Engineering	4. MFR NOTIFICATION: ☐ NOTIFIED ☒ NOT APPLICABLE	5. DOCUMENT ISSUE (Month/Year): March 1975
6. ORIGINATOR'S DOCUMENT TITLE: Test Procedure - Linear Equation Routines		7. DOCUMENT TYPE: ☒ GEN RPT ☐ NONSTD PART ☐ SPEC
8. ORIGINATOR'S DOCUMENT NUMBER: NWC TM 2410	9. ORIGINATOR'S PART NAME/IDENTIFICATION: N/A	
10. DOCUMENT (SUPERSEDES) (SUPPLEMENTS) ACCESS NO.: None	11. ENVIRONMENTAL EXPOSURE CODES: N/A	
12. MANUFACTURER: N/A	13. MANUFACTURER PART NUMBER: N/A	14. INDUSTRY/GOVERNMENT STANDARD NUMBER: N/A

15. OUTLINE, TABLE OF CONTENTS, SUMMARY, OR EQUIVALENT DESCRIPTION

The general test procedure is to apply each candidate routine to a number of test problems (with known answers) and to record the resulting accuracy and execution time. It is assumed that each candidate routine can solve the problem

$$AX=B \qquad (1)$$

where:

A is a general n x n coefficient matrix,

B is a general n x m right-hand-side matrix, and

X is an n x m solution matrix.

If B is set equal to the n x n identity matrix I (so that m = n), then X will be an approximation to A^{-1}. The accuracy achieved can be measured by computing the norm of the residual matrix

$$R = AX - I \qquad (\text{or} \qquad R = XA - I)$$

or better, by computing the norm of the error matrix

$$E = X - A^{-1}.$$

This approach is used in order to avoid having to solve ~~Eq. (1)~~ the first equation repeatedly using various right-hand sides B for each test matrix A. Execution time is measured by the system clock.

78 11 28 111

16. KEY WORDS FOR INDEXING: Error Measures; Test Output; Test Matrices; Univac 1110; Fortran (Doc Des--M)

17. GIDEP REPRESENTATIVE: M. H. Sloan	18. PARTICIPANT ACTIVITY AND CODE: Naval Weapons Center, China Lake (X7)

DD FORM 1 OCT 77 2000

35

FOREWORD

This work was done in connection with the author's position as Numerical Mathematics Coordinator, Central Computing Facility (CCF) and was supported by CCF funds. The actual testing of routines under the procedure outlined here is continuing. This is a preliminary report, subject to revision or withdrawal, and is not to be used as the basis for official action.

D. E. ZILMER
Head, Mathematics Division
Research Department
31 March 1975

NWC TM 2410, published by Code 607, 50 copies.

CONTENTS

4

INTRODUCTION

The general test procedure is to apply each candidate routine to a number of test problems (with known answers) and to record the resulting accuracy and execution time. It is assumed that each candidate routine can solve the problem

$$AX = B \tag{1}$$

where

A is a general $n \times n$ coefficient matrix

B is a general $n \times m$ right-hand-side matrix

X is an $n \times m$ solution matrix

If B is set equal to the $n \times n$ identity matrix I (so that $m = n$), then X will be an approximation to A^{-1}. The accuracy achieved can be measured by computing the norm of the residual matrix

$$R = AX - I \qquad (\text{or} \qquad R = XA - I)$$

or better, by computing the norm of the error matrix

$$E = X - A^{-1}.$$

This approach is used in order to avoid having to solve Eq.(1) repeatedly using various right-hand sides B for each test matrix A. Execution time is measured by the system clock.

ERROR MEASURES

Measures of error used in previous studies of linear equation routines [4,5] are[1]

---

[1] See Appendix A for definitions of the matrix norms $\|\cdot\|_S$, $\|\cdot\|_F$, $\|\cdot\|_M$, $\|\cdot\|_\infty$.

5

$$a = \frac{1}{n^2} \|R\|_S \qquad [4,5]$$

$$f = \frac{1}{n} \|R\|_F \qquad [4,5]$$

$$m = \frac{1}{n} \|R\|_M \qquad [5]$$

$$q = \frac{1}{n^2} \|E\|_S \qquad [4]$$

Newman and Todd [5] caution that matrices A and X exist for which $AX - I$ is almost 0 while elements of $XA - I$ are arbitrarily large. In their tests $AX - I$ and $XA - I$ varied by as much as three orders of magnitude. Lietzke *et al.* [4] report that

1. $a \leqslant f$ for each test matrix and each routine tested.
2. a and f are not reliable estimators of q.
3. The following estimator for $\|E\|_\infty$ seems to work for well-conditioned matrices:

$$\ell = \frac{\|XR\|_\infty}{1 - \|R\|_\infty}$$

where $R = AX - I$.

Measures of error recommended here are listed below. The Frobenius norm is recommended, since (like the Euclidean matrix norm) it is compatible with the Euclidean vector norm and invariant under unitary transformations.

Type of Error	Computation
Actual relative error	$\dfrac{\|E\|_F}{n\varepsilon \|A^{-1}\|_F}$
Actual absolute error	$\dfrac{1}{n\varepsilon} \|E\|_F$
Estimated absolute error	$\dfrac{\|XR\|_F}{n\varepsilon(1 - \|R\|_F}$
Residual error	$\dfrac{1}{n\varepsilon} \|R\|_F$

where $E = X - A^{-1}$ and $R = AX - I$. The above calculations must be done in double precision.

6

The infinity norm may also be used--as a check on the computations, and to test the conjecture of Lietzke *et al.* The normalization factor $\frac{1}{\varepsilon}$ (where $\varepsilon = 2^{-26} \cong 1.5 \times 10^{-8}$ is the single-precision machine 'infinitesimal' on UNIVAC 1100 Series computers) is used to scale calculated errors to unity. The normalization factor $\frac{1}{n}$ is used to scale the calculated errors to the same units as the elements of the error matrix.

TEST OUTPUT

For each candidate routine and each test matrix the following summary output is needed.

Heading identifying routine
Name of test matrix
Order of test matrix
Value of error return flag
Time for solution
Logarithm of condition number of test matrix
Actual relative error
Actual absolute error
Estimated absolute error
Residual error

In addition, the following detailed output should be available upon request.

Test matrix A
Calculated inverse X
Exact inverse A^{-1}
Error matrix $E = X - A^{-1}$
Residual matrix $R = AX - I$

TEST MATRICES

The test matrices recommended for use are listed below, together with a summary of their properties. These test matrices were culled from [3,7]. Each family of test matrices (except Wilkinson) is defined for any order n. The test matrices and their exact inverses are easily computed. The test matrices (except Newman-Todd) all have integer elements and span the range from well-conditioned to very ill-conditioned. One family (the Pei matrices) have a parameter that allows one to vary the condition number. Subroutines for generating the test matrices and their inverses are documented in Appendix B.

7

Matrix	n	Condition[a]	Form
Wilkinson	6	good	nonsymmetric
inverse Hilbert[b]	small	very bad	symmetric
Newman and Todd[b]	all	good	orthogonal, symmetric
Rutishauser[b]	small	very bad	lower triangular
Pei[c]	all	variable	symmetric and positive definite
Givens[b]	all	fair	symmetric and positive definite

[a] Precise condition numbers are given in Appendix A.
[b] Used in study by Newman and Todd [5].
[c] Used by Newman and Todd [5] for $a = n$ and by Lietzke *et al.* [4] for $a = 1$.

Wilkinson Matrix

This matrix (#3.4 in [3]) is for initial program checkout. It is well-conditioned and nonsymmetric.

$$W = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & -1 \\ -1 & 1 & 1 & 0 & 0 & 1 \\ 1 & -1 & 1 & 1 & 0 & -1 \\ -1 & 1 & -1 & 1 & 1 & 1 \\ -1 & -1 & 1 & -1 & 1 & -1 \end{bmatrix}$$

$$W^{-1} = \frac{1}{32} \begin{bmatrix} 16 & 8 & -4 & 2 & -1 & 1 \\ 0 & 16 & 8 & -4 & 2 & -2 \\ 0 & 0 & 16 & 8 & -4 & 4 \\ 0 & 0 & 0 & 16 & 8 & -8 \\ 0 & 0 & 0 & 0 & 16 & 16 \\ 16 & -8 & 4 & -2 & 1 & -1 \end{bmatrix}$$

Hilbert Matrix

These matrices (#3.8 in [3]) are very ill-conditioned. The Hilbert matrix H_n of order n is defined by

$$H_n = (h_{ij})$$

8

where

$$h_{ij} = \frac{1}{i+j-1} \qquad (i,j = 1,2,\ldots,n)$$

Because they are so badly conditioned, Hilbert matrices are often used to test matrix inversion routines. But this must be done properly. Inverses of the Hilbert matrices $T_n = H_n^{-1}$ are used as the test matrices, since all elements of T_n are integers. Reference [3] lists the inverse Hilbert matrices T_n for $n = 2,3,\ldots,10$. For testing purposes it probably suffices to try $n = 3$, 5, and 7. Integer overflow will occur on the UNIVAC 1110 for $n = 8$. Some facts about Hilbert matrices, based on information in [2], are summarized below. Here $C_2(H_n)$ denotes the ℓ_2 condition number of H_n (see Appendix A) and $\ell(T_n)$ denotes the largest element of T_n.

Facts About Hilbert Matrices

n	$C_2(H_n)$	$\|H_n\|_2$	$\|T_n\|_2$	$\ell(T_n)$
2	1.93(+1)	1.27	1.52(+1)	1.20(+1)
3	5.24(+2)	1.41	3.72(+2)	1.92(+2)
4	1.55(+4)	1.50	1.03(+4)	6.48(+3)
5	4.77(+5)	1.57	3.04(+5)	1.79(+5)
6	1.50(+7)	1.62	9.24(+6)	4.41(+6)
7	4.75(+8)	1.66	2.86(+8)	1.33(+8)
8	1.53(+10)	1.70	9.00(+9)	4.25(+9)
9	4.93(+11)	1.73	2.86(+11)	1.22(+11)
10	1.60(+13)	1.75	9.15(+12)	3.48(+12)

Newman-Todd Matrix

These matrices (#3.11 in [3]) are orthogonal and symmetric, but not positive definite. They are defined by

$$A_n = (a_{ij}), \qquad a_{ij} = \sqrt{\frac{2}{n+1}}\,\sin\frac{ij\pi}{n+1}$$

These matrices are well-conditioned, but costly to generate (because of calls to DSIN and DSQRT).

9

Rutishauser Matrix

These matrices (#3.15 in [3] are ill-conditioned and lower triangular. They are defined by

$$R_n = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 1 & -3 & 3 & -1 & 0 & \cdots & 0 \\ 1 & -4 & 6 & -4 & 1 & & \\ \cdot & \cdot & \cdot & \cdot & & \cdot & \\ \cdot & \cdot & \cdot & \cdot & & \cdot & \\ \cdot & \cdot & \cdot & \cdot & & \cdot & \\ 1 & -n+1 & & & & & (-1)^{n-1} \end{bmatrix}$$

The elements of R_n, except for sign, are the numbers in Pascal's triangle, that is, binomial coefficients. R_n is its own inverse.

Pei Matrix

The Pei matrices ([3, p. 18] and #18 in [7]) are symmetric and positive definite. They are defined by

$$P_n(a) = (p_{ij}), \qquad p_{ij} = \begin{cases} a+1 & i = j \\ 1 & i \neq j \end{cases}$$

where $a > 0$. Note that as $a \to 0$, P becomes singular. Eigenvalues of P are $\lambda = a + n$ (simple) and $\lambda = a$ (multiplicity $n - 1$). Thus, for $a \cong n$, P_n is well-conditioned; but for $a \cong 0$, P_n is very ill-conditioned. Recommended choices of the Pei parameter a are

$$a = 64\varepsilon,\ 1,\ n$$

where $\varepsilon = 2^{-26} \cong 1.5 \times 10^{-8}$.

To compute the inverse of $P_n(a)$, let J be the matrix with all elements equal to 1. Then (dropping the subscript n)

$$P = aI + J.$$

Since $J^2 = nJ$, we expect $P^{-1} = Q$ to be linear in J,

$$Q = bI + cJ.$$

Computing the product PQ we find

$$I = PQ = (aI + J)(bI + cJ) = abI + (b + ac + nc)J.$$

Equating coefficients,

$$\left.\begin{aligned} ab &= 1 \\ b + ac + nc &= 0 \end{aligned}\right\} \Longrightarrow \begin{aligned} b &= \frac{1}{a} \\ c &= -\frac{b}{a+n} = -\frac{1}{a(a+n)} \;. \end{aligned}$$

Hence,

$$Q = \frac{1}{a}\left(I - \frac{1}{a+n}\,J\right)$$

or

$$P^{-1} = (q_{ij}), \qquad q_{ij} = \begin{cases} \dfrac{a + n - 1}{a(a+n)} & i = j \\[2ex] -\dfrac{1}{a(a+n)} & i \neq j \;. \end{cases}$$

Givens Matrix

The Givens matrices (#8 in [7]) are poorly-conditioned, symmetric, and positive definite. They are defined by

$$G_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 3 & 3 & \cdots & 3 \\ 1 & 3 & 5 & \cdots & 5 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & 3 & 5 & \cdots & 2n-1 \end{bmatrix}$$

Thus, $G_n = (g_{ij})$ where $g_{ij} = 2\min(i,j) - 1$. The inverse is given by

$$G_n^{-1} = \frac{1}{2}\begin{bmatrix} 3 & -1 & & & & & & \mathbf{0} \\ -1 & 2 & -1 & & & & & \\ & -1 & 2 & \cdot & & & & \\ & & -1 & \cdot & \cdot & & & \\ & & & \cdot & \cdot & \cdot & & \\ & & & & \cdot & \cdot & -1 & \\ \mathbf{0} & & & & & \cdot & 2 & -1 \\ & & & & & & -1 & 1 \end{bmatrix}$$

Note that the inverse Givens matrix is tridiagonal.

Appendix A

MATRIX NORMS AND CONDITION NUMBERS

The more common matrix and vector norms are defined here, without much explanation. For a more complete discussion of matrix and vector norms and their properties, see [1,6]. Vector norms in common use are

$$\|x\|_1 = \Sigma_i |x_i|$$

$$\|x\|_2 = \sqrt{\Sigma_i x_i^2} \qquad \text{(Euclidean)}$$

$$\|x\|_\infty = \max_i \; |x_i| \qquad \text{(Chebyshev)}.$$

These are particular instances of the p-norm

$$\|x\|_p = \Sigma_i |x_i|^p.$$

Any *vector* norm induces a *matrix* norm according to the following relation:

$$\|A\| = \max_{\|x\|=1} \|Ax\| \qquad \text{(A.1)}$$

where the maximum is taken over all x such that $\|x\| = 1$. Matrix norms induced in this way are <u>compatible</u> with the underlying vector norm, in the sense that

$$\|Ax\| \leq \|A\| \cdot \|x\| \qquad \text{(for all } x\text{)}.$$

The matrix norms induced by the vector p-norms ($p = 1, 2, \infty$) are

$$\|A\|_1 = \max_j \Sigma_i |a_{ij}| \qquad \text{(column sum)}$$

$$\|A\|_2 = \sqrt{\text{largest eigenvalue of } A^T A} \qquad \text{(Euclidean)}$$

$$\|A\|_\infty = \max_i \Sigma_j |a_{ij}| \qquad \text{(row sum)}$$

Matrix norms, besides those induced by vector p-norms, are also used. Among the most useful are

$$\|A\|_S = \Sigma_i \Sigma_j |a_{ij}| \qquad \text{(sum norm)}$$

$$\|A\|_F = \sqrt{\Sigma_i \Sigma_j \; a_{ij}^2} \qquad \text{(Frobenius norm)}$$

$$\|A\|_M = n \max_i \max_j |a_{ij}| \qquad \text{(max norm).}$$

It is easily proven that $\|A\|_S$ is compatible with $\|x\|_1$, that $\|A\|_F$ is compatible with $\|x\|_2$, and that $\|A\|_M$ is compatible with $\|x\|_\infty$. Also,

$$\frac{1}{n} \|A\|_S \leqslant \|A\|_\infty \leqslant \|A\|_M \leqslant n \|A\|_S$$

$$\frac{1}{n} \|A\|_S \leqslant \|A\|_1 \leqslant \|A\|_M \leqslant n \|A\|_S$$

so that the choice of a norm is not too critical in finite-dimensional linear spaces (that is, for matrices).

The <u>condition</u> number of a matrix for a given matrix norm is defined by

$$c(A) = \|A\| \cdot \|A^{-1}\|. \qquad \text{(A.2)}$$

From this definition it is clear that $c(A^{-1}) = c(A)$. The condition number recommended in this report is derived from the Frobenius matrix norm

$$c_F(A) = \|A\|_F \cdot \|A^{-1}\|_F.$$

The Frobenius condition number should be a good approximation to the Euclidean condition number

$$c_2(A) = \sqrt{\text{largest eigenvalue of } A^T A} \Big/ \sqrt{\text{smallest eigenvalue of } A^T A}$$

because of the inequalities

$$\|A\|_2 < \|A\|_F < \sqrt{n} \|A\|_2 .$$

A specialization of the Euclidean condition number, introduced by von Neumann and Goldstine in the early days of computing, is frequently found in the literature (for example, [3]). This condition number is defined by

14

$$c_\lambda(A) = \frac{|\text{largest eigenvalue of } A|}{|\text{smallest eigenvalue of } A|}$$

It is sometimes called the 'spectral' condition number, since the spectral radius of A is the magnitude of the largest eigenvalue of A. When A is symmetric, the Euclidean and spectral condition numbers agree

$$c_2(A) = c_\lambda(A).$$

Spectral condition numbers for some of the test matrices discussed in this report are given below. These are quoted from [3].

matrix	spectral condition number
Newman-Todd	1
Rutishauser	$\sim e^{4n \ln 2}$
Pei	$1 + \frac{n}{a}$
Givens	$\sim \left(\frac{4n}{\pi}\right)^2$

Condition numbers computed from Eq. (A.2) for various matrix norms will, because of the inequalities relating the matrix norms, all be of the same order of magnitude as the spectral condition numbers quoted above.

For any condition number $c(A)$, Forsythe and Moler [2] argue that

$$\log_{10} c(A)$$

is approximately the number of significant digits lost when calculating A^{-1}. A little arithmetic, based on the condition numbers quoted above, yields the following:

matrix		order	$\log_{10} c(A)$
Inverse Hilbert		3	2.72
		5	5.68
		7	7.18
Newman-Todd		all	0
Rutishauser		3	3.61
		5	6.02
		7	8.43
Pei	$a = n$	all	0
	$a = 1$	10	1.
		50	1.70
		100	2.
	$a = 64\varepsilon$	10	7.02
		50	7.72
		100	8.02
	$a = \varepsilon$	10	8.83
		50	9.53
		100	9.83
Givens		10	2.71
		50	4.10
		100	4.71

From the above table it is apparent that condition numbers for families of dense matrices of orders up to several hundred can be classified as follows:

Condition	Functional Form	Example
Good	constant	Newman-Todd, Pei ($a = n$)
Fair	linear in n	Pei ($a \cong 1$)
	quadratic in n	Givens
Very bad	exponential in n	Rutishauser, Hilbert

Note too that Pei ($a = 64\varepsilon = 2^{-20}$) presents a more interesting test problem than Pei ($a = \varepsilon$).

Appendix B

PROGRAM PACKAGE FOR TESTING LINEAR EQUATION ROUTINES

A program package for testing linear equation routines was written according to the specifications of this report. It consists of the following:

Main Program:	LEQTST	- overall logic
Subprograms:	LTxxxx	- generates test matrix and its exact inverse
	LTEST	- calls candidate routine to calculate inverse and measures execution time
	LTERR	- computes error matrix, residual matrix and four measures of error
	FNORM	- returns Frobenius norm of a matrix
	LTPRNT	- prints summary test results
	LTRITE	- writes all matrices, if detailed printout requested
	MXRITE	- writes a single matrix

Program documentation for these program elements is contained in this Appendix, and conforms to the documentation standards of [8].

NAME	FNORM
PURPOSE	To return Frobenius norm of a matrix

$$\|A\|_F = \sqrt{\Sigma_i \Sigma_j \, a_{ij}^2}$$

USAGE F = FNORM (A, N, NN)

A.....The matrix - (N, N)	Input
N.....Order of matrix A	Input
NN....Row dimension of A	Input

ACCESS	LIB NWC*MATHLIB.
ERRORS	None
REMARKS	Although A is single precision, the computation of FNORM is done in double precision.

PROGRAM INFO

Machine	UNIVAC 1110
Language	FORTRAN V
Author	L. W. Lucas, Code 4033 NWC
Date	14 March 1975
Status	Certified, Fully Supported by CCF
Entry Names	FNORM
External Refs	DSQRT, NERR3$
Filename	NWC*MATHLIB
Element/Vers	FNORM
Storage	74 words
Timing	Unknown
Consultant	L. W. Lucas, Ext. 3561

TESTING	Output from FNORM was checked against values computed by hand.
REFERENCE	NWC TM 2410

NAME LEQTST

PURPOSE To test a linear equation routine by using it to compute inverses X of tests matrices A having known inverses. The resulting error $E = X - A^{-1}$ and residual $R = AX - I$ are used to compute several measures of computational error.

USAGE Main program. There are no inputs to LEQTST. User options involve the setting of three program variables.

NN........Parameter variable which controls the maximum size of test matrices used.

DETAIL....Logical variable which if .TRUE. causes all matrices involved - A, X, A^{-1}, E, R - to be printed out, as well as summary test information.

IW........Integer variable which specifies the logical unit number for output.

LEQTST is set up to use the following test matrices.

matrix	order
Wilkinson	6
Inverse Hilbert	3, 5, 7
Newman-Todd	5, 10, 50, 100
Rutishauser	5, 10, 15, 20
Pei, $a = 64\varepsilon$	5, 10, 50, 100
$a = 1$	5, 10, 50, 100
$a = n$	5, 10, 50, 100
Givens	5, 10, 50, 100

The maximum order actually used is controlled by the value of *NN*.

ACCESS IN 4033519*NM-BENCH.LEQTST

REMARKS The size of test matrices used in LEQTST is limited only by the available core storage. The design of LEQTST does not provide for buffering to mass storage. LEQTST is not intended to test sparse matrix routines. The test matrices used are believed to be representative of moderately-sized dense matrices.

PROGRAM INFO

Machine	UNIVAC 1110
Language	FORTRAN V
Author	L. W. Lucas, Code 4033 NWC
Date	14 March 1975
Status	Unsupported
Entry Names	None
External Refs	LTHDG, LTWILK, LTEST, LTERR, FNORM, LTPRNT, LTRITE, LTHILB, LTNEWT, LTRUTH, LTPEI, LTGIVN, NINTR$, ALOG, NSTOP$
Filename	4033519*NM-BENCH.
Element/Vers	LEQTST
Storage	51080 words (*NN* = 100)
Timing	103 seconds (*NN* = 100)
Consultant	None

TESTING The component subprograms for LEQTST were all handchecked.

REFERENCE NWC TM 2410

NAME LTERR

PURPOSE To compute error and residual matrices and four measures of computational error.

USAGE CALL LTERR(A, X, AI, E, R, N, NN, ERR)

A....Test matrix - (N, N)		Input
X....Computed inverse of A		Input
AI...Exact inverse of A		Input
E....Error matrix, $E = X - AI$		Output
R....Residual matrix, $R = AX - I$		Output
N....Order of matrices A, X, AI, E, R		Input
NN...Row dimension of A, X, AI, E, R		Input
ERR..Array of error measures		output

ACCESS IN 4033519*NM-BENCH.LTERR

ERRORS None

REMARKS Although the arrays A, X, AI, E and R are all single precision, the computations within LTERR are performed in double precision. The measures of error returned in ERR are defined in the reference.

PROGRAM INFO

Machine	UNIVAC 1110
Language	FORTRAN
Author	L. W. Lucas, Code 4033 NWC
Date	14 March 1975
Status	Unsupported
Entry Names	LTERR
External Refs	FNORM, DSQRT, NERR3$
Filename	4033519*NM-BENCH.
Element/Vers	LTERR
Storage	333 words
Timing	unknown
Consultant	None

TESTING Output from LTERR was checked against values computed by hand.

REFERENCE NWC TM 2410

21

NAME	LTEST
PURPOSE	To call candidate routine for calculating inverse of test matrix and to measure execution time
USAGE	CALL LTEST (A, X, N, NN, TIME, WK, C, IER)

A......Test matrix - (*N*, *N*)		Input
X......Computed inverse of *A*		Output
N......Order of matrices *A*, *X*		Input
NN.....Row dimension of *A*, *X*		Input
TIME...Time to compute inverse		Output
WK.....Work space of length *N*N*		Scratch
C......Work space - (*N*, *N*)		Scratch
IER....Error return flag		Output

ACCESS	IN 4033519*NM-BENCH.LTEST
ERRORS	No errors may occur within LTEST itself. IER returns the value of an error flag from the candidate routine (if any).
REMARKS	The input matrix *A* is copied into the work space *C*, which is passed to the candidate routine, just in case the routine destroys any coefficient matrix given to it. LTEST will obviously have to be rewritten for each candidate routine tested. SETCLK and LKCLKS are entry points to the system clock routine. LEQT1F is the routine being tested.

PROGRAM INFO

Machine	UNIVAC 1110
Language	FORTRAN
Author	L. W. Lucas, Code 4033 NWC
Date	14 March 1975
Status	Unsupported
Entry Names	LTEST
External Refs	SETCLK, LEQTIF, LKCLK, NERR3$
Filename	4033519*NM-BENCH.
Element/Vers	LTEST
Storage	167 words
Timing	unknown
Consultant	None

REFERENCE NWC TM 2410

NAME LTGIVN

PURPOSE To generate the Givens test matrix of order *N* and its exact inverse.

USAGE CALL LTGIVN (G, GI, N, NN)

G.....Test matrix - (*N*, *N*)	Outpu
GI....Exact inverse of G	Output
N.....Order of G	Input
NN....Row dimension of G	Input

ACCESS LIB NWC*MATHLIB.

ERRORS None

REMARKS None

PROGRAM INFO

Machine	UNIVAC 1110
Language	FORTRAN
Author	L. W. Lucas, Code 4033 NWC
Date	14 March 1975
Status	Certified, Fully Supported by CCF
Entry Names	LTGIVN
External Refs	NERR3$
Filename	NWC*MATHLIB
Element/Vers	LTGIVN
Storage	198 words
Timing	unknown
Consultant	L. W. Lucas, Ext. 3561

TESTING Output from LTGIVN was handchecked against Gregory and Karney.

METHOD The Givens test matrix G_n and its inverse are defined by

$$G_n = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 3 & & & \\ & & & \ddots & \\ 1 & 3 & 5 & \cdots & 2n-1 \end{bmatrix}$$

$$G_n^{-1} = \frac{1}{2}\begin{bmatrix} 3 & -1 & & & \\ -1 & 2 & & & \\ & & \ddots & & \\ & & & 2 & 1 \\ & & & 1 & 1 \end{bmatrix}$$

REFERENCE Gregory and Karney. *A Collection of Matrices for Testing Computational Algorithms*. Wiley-Interscience, 1969.

NAME	LTHILB
PURPOSE	To generate the inverse Hilbert test matrix of order *N* and its exact inverse (Hilbert matrix)
USAGE	CALL LTHILB (T, H, N, NN)

T.....Inverse Hilbert matrix - (*N*, *N*) Output
H.....Hilbert matrix Output
N.....Order of T, H Input
NN....Row dimension of T, H Input

ACCESS	LIB NWC*MATHLIB.
ERRORS	None
REMARKS	The inverse Hilbert matrix is used as a test matrix since its elements are integers. Due to integer overflow in computing the elements of T, the maximum usable value of *N* is 7. The elements of H are computed using double precision arithmetic and are exact to single precision.

PROGRAM INFO

Machine	UNIVAC 1110
Language	FORTRAN
Author	L. W. Lucas, Code 4033 NWC
Date	14 March 1975
Status	Certified, Fully Supported by CCF
Entry Names	LTHILB
External Refs	NERR3$
Filename	NWC*MATHLIB
Element/Vers	LTHILB
Storage	254 words
Timing	unknown
Consultant	L. W. Lucas, Ext. 3561

TESTING	Output from LTHILB was handchecked against Gregory and Karney.
METHOD	The algorithm for computing the elements of T is given in Forsythe-Moler, p. 85. The formula defining *H* is

$$H = \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n} \\ \frac{1}{2} & \frac{1}{3} & & & \\ \frac{1}{3} & & \ddots & & \\ \vdots & & & & \\ \frac{1}{n} & & & & \frac{1}{2n-1} \end{bmatrix}$$

REFERENCES

Gregory and Karney
A Collection of Matrices for Testing Computational Algorithms
Wiley-Interscience, 1969

Forsythe and Moler
Computer Solution of Linear Algebraic Systems
Prentice-Hall, 1967

NAME	LTNEWT
PURPOSE	To generate Newman-Todd test matrix of order *N* and its exact inverse.
USAGE	CALL LTNEWT (A, AI, N, NN)

A.....Newman-Todd test matrix - (*N*, *N*)		Output
AI....Exact inverse of A		Output
N.....Order of A		Input
NN....Row dimension of A, AI		Input

ACCESS	LIB NWC*MATHLIB
ERRORS	None
REMARKS	Since *A* is orthogonal, *AI* is set equal to the transpose of *A*. The computations are done in double precision, since the elements of *A* are not exactly representable in single precision.

PROGRAM INFO

Machine	UNIVAC 1110
Language	FORTRAN
Author	*L. W. Lucas*, Code 4033 NWC
Date	14 March 1975
Status	Certified, Fully Supported by CCF
Entry Names	LTNEWT
External Refs	DSQRT, DSIN, NERR3$
Filename	NWC*MATHLIB
Element/Vers	LTNEWT
Storage	142 words
Timing	unknown
Consultant	L. W. Lucas, Ext. 3561.

TESTING Output from LTNEWT was handchecked against Gregory and Karney.

METHOD The Newman-Todd test matrix *A* is defined by

$$A = (a_{ij}), \quad a_{ij} = \frac{2}{n+1} \sin \left(\frac{ij\pi}{n+1}\right)$$

REFERENCE Gregory and Karney
A Collection of Matrices for Testing Computational Algorithms
Wiley-Interscience, 1969

26

NAME	LTPEI
PURPOSE	To generate Pei test matrix of order *N* with parameter *S* and its exact inverse.
USAGE	CALL LTPEI (P, Q, N, NN, S)

P	Pei test matrix - (*N*, *N*)	Output
Q	Exact inverse of P	Output
N	Order of P, Q	Input
NN	Row dimension of P, Q	Input
S	Pei parameter	Input

ACCESS LIB NWC*MATHLIB.

ERRORS None

REMARKS The elements of P are integers and are exact to single precision. The elements of Q are fractions and are computed in double precision. The parameter *S* can be used to vary the condition number of P.

PROGRAM INFO

Machine	UNIVAC 1110
Language	FORTRAN
Author	L. W. Lucas, Code 4033 NWC
Date	14 March 1975
Status	Certified, Fully Supported by CCF
Entry Names	LTPEI
External Refs	NERR3$
Filename	NWC*MATHLIB
Element/Vers	LTPEI
Storage	183 words
Timing	unknown
Consultant	L. W. Lucas, Ext. 3561

TESTING Output from LTPEI was handchecked against Gregory and Karney.

METHOD The formulas defining *P* and *Q*, taken from Gregory-Karney, are

$$P = (p_{ij}) \qquad p_{ij} = \begin{cases} 1 + s & i = j \\ 1 & i \neq j \end{cases}$$

$$Q = (q_{ij}) \qquad q_{ij} = \begin{cases} \dfrac{s + n - 1}{s(s + n)} & i = j \\ -\dfrac{1}{s(s + n)} & i \neq j \end{cases}$$

where *s* is the Pei parameter.

REFERENCE Gregory and Karney
A Collection of Matrices for Testing Computational Algorithms
Wiley-Interscience, 1969

28

NAME	LTPRNT
PURPOSE	To print summary test results for LEQTST
USAGE	CALL LTHDG (IW)

IW.....Unit number for printout

CALL LTPRNT (IW, NAME, N, IER, TIME, COND, ERR)

IW.....Unit number for printout	Input
NAME...Name of test matrix	Input
N......Order of test matrix	Input
IER....Error return from candidate routine	Input
TIME...Time to compute inverse	Input
COND...Logarithm of condition number	Input
ERR....Array of error measures	Input

ACCESS	IN 4033519*NM-BENCH.LTPRNT
ERRORS	None
REMARKS	NAME is a 12-character hollerith string giving the name of the test matrix. Entry LTPRNT is used to print summary test results. Entry LTHDG is used to print headings for the summary test results.

PROGRAM INFO

Machine	UNIVAC 1110
Language	FORTRAN
Author	L. W. Lucas, Code 4033 NWC
Date	14 March 1975
Status	Unsupported
Entry Names	LTHDG, LTPRNT
External Refs	NWDU$, NIØ1$, NIØ2$, NERR3$
Filename	4033519*NM-BENCH
Element/Vers	LTPRNT
Storage	108 words
Timing	unknown
Consultant	None

REFERENCE NWC TM 2410

39

NAME	LTRITE
PURPOSE	To write matrices for LEQTST
USAGE	CALL LTRITE (IW, A, X, AI, E, R, N, NN)

IW.....Logical unit number for output	Input
A......Test matrix - (*N*, *N*)	Input
X......Computed inverse	Input
AI.....Exact inverse	Input
E......Error matrix	Input
R......Residual matrix	Input
N......Order of matrices	Input
NN.....Row dimension of arrays	Input

ACCESS	IN 4033519*NM-BENCH.LTRITE
ERRORS	None
REMARKS	None

PROGRAM INFO

Machine	UNIVAC 1110
Language	FORTRAN
Author	L. W. Lucas, Code 4033 NWC
Date	14 March 1975
Status	Unsupported
Entry Names	LTRITE
External Refs	MXRITE, NERR3$
Filename	4033519*NM-BENCH
Element/Vers	LTRITE
Storage	96 words
Timing	unknown
Consultant	None

REFERENCE NWC TM 2410

NAME LTRUTH

PURPOSE To generate Rutishauser test matrix of order *N* and its exact inverse.

USAGE CALL LTRUTH (R, RI, N, NN, JR)

R.....Rutishauser test matrix - (*N*, *N*)		Output
RI....Exact inverse of R		Output
N.....Order of R, RI		Input
NN....Row dimension of R, RI		Input
JR....Work space array of length *N*		Scratch

ACCESS LIB NWC*MATHLIB.

ERRORS None

REMARKS The scratch array JR is used to store the *j*th column of R, while generating the *j*+1-th column via a recursion formula. R is its own inverse, so RI is just a copy of R.

PROGRAM INFO

Machine	UNIVAC 1110
Language	FORTRAN
Author	L. W. Lucas, Code 4033 NWC
Date	14 March 1975
Status	Certified, Fully Supported by CCF
Entry Names	LTRUTH
External Refs	NERR3$
Filename	NWC*MATHLIB
Element Vers	LTRUTH
Storage	197 words
Timing	unknown
Consultant	L. W. Lucas, Ext. 3561

TESTING Output from LTRUTH was handchecked against Gregory and Karney.

METHOD The Rutishauser test matrix is defined by

$$R = \begin{bmatrix} 1 & & & & \mathbf{0} \\ 1 & -1 & & & \\ 1 & -2 & 1 & & \\ \vdots & & & \ddots & \\ 1 & -n+1 & & & (-1)^{n-1} \end{bmatrix}$$

31

The columns of *R* are, except for sign, the diagonals in Pascal's triangle. Thus, the following recursion formula can be used to generate the elements of *R*.

$$r_{ij} = \begin{cases} 0 & i < j \\ r_{i-1,\, j} - r_{i-1,\, j-1} & i \geq j \end{cases}$$

REFERENCE Gregory and Karney
A Collection of Matrices for Testing Computational Algorithms
Wiley-Interscience, 1969

NAME	LTWILK
PURPOSE	To generate Wilkinson test matrix of order 6 and its exact inverse.
USAGE	CALL LTWILK (W, V, N, NN)

W.....Wilkinson test matrix		Output
V.....Exact inverse of W		Output
N.....Order of W, V		Output
NN....Row dimension of W, V		Input

ACCESS	LIB NWC*MATHLIB.
ERRORS	None
REMARKS	*N* is set equal to 6 by LTWILK.
PROGRAM INFO	

Machine	UNIVAC 1110
Language	FORTRAN
Author	L. W. Lucas, Code 4033 NWC
Date	14 March 1975
Status	Certified, Fully Supported by CCF
Entry Names	LTWILK
External Refs	*NERR3$*
Filename	NWC*MATHLIB
Element/Vers	LTWILK
Storage	165 words
Timing	unknown
Consultant	L. W. Lucas, Ext. 3561

TESTING	Output from LTWILK was handchecked against Gregory and Karney.
METHOD	The internal arrays *X* and *Y* are initialized to the Wilkinson test matrix, and its exact inverse, respectively, which are given in the reference. These are then copied into *W* and *V*, respectively, upon entry to LTWILK.
REFERENCE	Gregory and Karney *A Collection of Matrices for Testing Computational Algorithms* Wiley-Interscience, 1969

NAME	MXRITE
PURPOSE	To write out a general matrix, preceded by a title.
USAGE	CALL MXRITE (IW, A, N, M, NN, TITLE)

IW.....Logical unit number for printout		Input
A......The matrix (*N*, *M*)		Input
N......Rows in A		Input
M......Columns in A		Input
NN.....Row dimension of A		Input
TITLE..Title printed above A		Input

ACCESS	NWC*MATHLIB.
ERRORS	None
REMARKS	TITLE is an 18-character hollerith string printed above A. The matrix is printed out 8 columns across the page. Remaining columns are printed below, again 8 columns across the page.

PROGRAM INFO

Machine	UNIVAC 1110
Language	FORTRAN
Author	L. W. Lucas, Code 4033 NWC
Date	14 March 1975
Status	Certified, Fully Supported by CCF
Entry Names	MXRITE
External Refs	NWDU$, NIØ1$, NIØ2$, NERR3$
Filename	NWC*MATHLIB
Element/Vers	MXRITE
Storage	167 words
Timing	unknown
Consultant	L. W. Lucas, Ext. 3561

REFERENCE	None

REFERENCES

1. Bellman, R. (1970). *Introduction to Matrix Analysis*, 2nd ed., McGraw-Hill, New York. LC76-96425.

2. Forsythe, G., and C. B. Moler (1967). *Computer Solution of Linear Algebraic Systems*, Prentice-Hall, New Jersey.

3. Gregory, R. T., and D. L. Karney (1969). *A Collection of Matrices for Testing Computational Algorithms*, Wiley-Interscience, New York.

4. Lietzke, M. H. *et al.* (1964). "A comparison of several methods for inverting large symmetric positive definite matrices," MATH COMP 18, 449-463.

5. Newman, M., and J. Todd (1958). "The evaluation of matrix inversion programs," J SIAM 6, 466-476.

6. Stewart, G. W. (1973). *Introduction to Matrix Computations*, Wiley, New York. LC72-82636.

7. Westlake, J. R. (1968). *A Handbook of Numerical Matrix Inversion and Solution of Linear Equations*, Wiley, New York.

8. Lucas, L. W. (1975). *Documentation Guidelines for Subprograms*, China Lake, CA, Naval Weapons Center. (NWC TM 2408, publication UNCLASSIFIED.)