Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

W. Hoffmann, K. Potma

NUMVEC FORTRAN library manual
Chapter: Simultaneous linear equations

Department of Numerical Mathematics   Report NM-R8903   January

NUMVEC
is a library of NUMerical software for VECtor and parallel computers in FORTRAN.
The documentation conforms as much as possible to that of the NAG - library. A Subsection: *11.1 Vectorization information* mentions, a.o., whether the code is standard FORTRAN 77. If not, information is given about special vector-syntax used and about specific machine(s) to which the code is aimed.
The source code described can be obtained by writing to the NUMVEC Library Manager at the CWI.

NUMVEC FORTRAN Library manual

Chapter: Simultaneous Linear Equations

W. Hoffmann, K. Potma,
Department of Computer Systems
Faculty of Mathematics and Computer Science
University of Amsterdam
Kruislaan 409, 1098 SJ Amsterdam, The Netherlands

This document describes a set of NUMVEC FORTRAN Library routines, dealing with the full-rank linear least-squares problem and the orthogonal basis problem. In particular, it contains a subroutine for calculating the factors of a QR decomposition with a compactly stored orthogonal factor. This storage scheme is customary when using Householder reflections. Moreover, it contains subroutines for calculating the product of such a coded orthogonal matrix (or its transposed) with a vector and a subroutine for calculating the explicit form of this orthogonal matrix.

1980 Mathematics subject classification (1985 revision): 65F20, 65V05, 15A23.
1982 CR Categories: 5.14.

Keywords & Phrases: least squares problems, orthogonal basis problem, QR Decomposition, Householder reflections.

Note: The implementations are available in FORTRAN 200 (the CYBER 200 series FORTRAN, a superset of standard FORTRAN including vector extensions).

Report NM-R8903
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

EXPLQ - NUMVEC FORTRAN Library Routine Document

1. Purpose

EXPLQ calculates the $m \times n$ matrix Q from the QR factorization as calculated by HSHVOX, in its explicit form.

2. Specifications

```
      SUBROUTINE  EXPLQ(A, Q, IA, M, N, W)
C     INTEGER  IA, M, N
C     REAL  A(IA, N), Q(IA, N), W(N)
```

3. Description

The sequence of Householder reflections that defines matrix Q is applied, in reverse order, to the identity matrix of appropriate size to form matrix Q.

4. References

None.

5. Parameters

A - REAL array of DIMENSION (IA,p) where p ⩾ N.
Before entry, A should contain in its lower trapezoidal part the Householder vectors as calculated by HSHVOX.
Unchanged on exit.

Q - REAL array of DIMENSION (IA,p) where p ⩾ N.
On exit Q contains the $m \times n$ orthogonal matrix defining the QR factorization of A as calculated by HSHVOX.

IA - INTEGER.
On entry, IA specifies the first dimension of arrays A and Q as declared in the calling (sub)program (IA ⩾ M).
Unchanged on exit.

M - INTEGER.
On entry, M specifies the number of rows of matrices A and Q.
Unchanged on exit.

N - INTEGER.
On entry, N specifies the number of columns of matrices A and Q.
Unchanged on exit.

W - REAL array of DIMENSION at least (M).
Used as work space.

6. Error indicators and warnings

None.

7. Auxiliary routines

No auxiliary routines are used.

8. Timing

The time taken is proportional to MN^2 and is approximately equal to the time of HSHVOX for the same sizes of M and N.

9. Storage

There are no internally declared arrays.

10. Accuracy

The measure of orthogonality, $\|Q^TQ-I\|_2$, is roughly equal to machine-precision.

11. Further comments

None.

11.1. Vectorization information

The routine is written in FORTRAN 200, making use of its vector syntax extensions.

12. Keywords

Orthogonal basis.
QR factorization.

13. Example

To calculate an orthogonal basis for the columnspace of matrix A where

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 1 & 0 & -1 \\ 1 & 2 & 3 \\ 1 & 0 & 1 \end{pmatrix}$$

13.1. Program text

```
C
C     ORTBAS EXAMPLE PROGRAM TEXT
C     NUMVEC 1988
C     MARK 1
C
      PROGRAM ORTBAS(OUTPUT,TAPE6=OUTPUT)
      REAL A(4,3), Q(4,3), DIAGR(3), W1(3), W2(4)
      INTEGER M, N, I, J, NZER, NMAX
      DATA ((A(I,J),J=1,3),I=1,4)
     + / 1., 2., 1.,
     +   1., 0.,-1.,
     +   1., 2., 3.,
```

```
     +    1., 0., 1. /
      M = 4
      N = 3
      NMAX = 4
      WRITE(6,99996)
      DO 40 I = 1, M
        WRITE(6,99997)(A(I,J),J = 1, 3)
   40 CONTINUE
      CALL HSHVOX(A, NMAX, M, N, DIAGR, W2, 0, NZER)
      CALL EXPLQ(A, Q, NMAX, M, N, W1)
      WRITE(6,99998)
      DO 50 I = 1, M
        WRITE(6,99999)(Q(I,J),J = 1, 3)
   50 CONTINUE
      STOP
99996 FORMAT ('1ORTHOGONAL BASIS PROGRAM'/'0MATRIX A')
99997 FORMAT (3(1X,F7.3))
99998 FORMAT ('0CALCULATED BASIS: ')
99999 FORMAT (3(1X,F7.3))
      END
```

13.2. Program results

```
ORTHOGONAL BASIS PROGRAM

MATRIX A
  1.000   2.000   1.000
  1.000   0.000  -1.000
  1.000   2.000   3.000
  1.000   0.000   1.000

CALCULATED BASIS:
  0.500   0.500   0.500
  0.500  -0.500   0.500
  0.500   0.500  -0.500
  0.500  -0.500  -0.500
```

HSHVOX - NUMVEC FORTRAN Library Routine Document

1. Purpose

HSHVOX calculates a QR-factorization of a matrix by means of Householder-reflections.

2. Specifications

```
      SUBROUTINE HSHVOX(A, IA, M, N, DIAGR, W, IP, NZER)
C     INTEGER  IA, M, N, IP, NZER
C     REAL  A(IA, N), DIAGR(N), W(M)
```

3. Description

Given an $m \times n$ matrix A, $m \geqslant n$, a factorization $A = QR$ is calculated where Q is $m \times n$ orthogonal and R is $n \times n$ upper triangular. The routine uses Householder's method with optional scaling of the columnvectors for protection against overflow.
Matrix Q is delivered in factorized form, each factor being defined by the appropriate 'Householder vector'. If the diagonal of R contains p (say) entries equal to zero, then the rank of matrix A is at most $n-p$.

4. References

[1] Golub, G.H., Van Loan, C.F., Matrix Computations, North Oxford Academic, Oxford, 1983.

[2] Hoffmann, W., Definition and use of Householder reflections, Report CS-88-05, University of Amsterdam, Department of Computer Systems, 1988.

5. Parameters

A - REAL array of DIMENSION (IA,p) where p $\geqslant$ N.
Before entry, A should contain the elements of the real matrix.
On successful exit, it will contain, in its lower trapezoidal part, the Householder vectors defining the reflections applied and, in its strictly upper triangular part the elements of the strict upper triangle of the calculated matrix R.

IA - INTEGER.
On entry, IA specifies the first dimension of array A as declared in the calling (sub)program (IA $\geqslant$ M).
Unchanged on exit.

M - INTEGER.
On entry, M specifies the number of rows of matrix A.
Unchanged on exit.

N - INTEGER.
On entry, N specifies the number of columns of matrix A.
Unchanged on exit.

DIAGR - REAL array of DIMENSION at least (N).
On successful exit DIAGR will contain the diagonal of R.

W - REAL array of DIMENSION at least (M).
Used as working space.

IP - INTEGER value.

If IP has the value -1, then Euclidean norms are calculated without overflow protection. This causes the subroutine to be less robust, but may save some processing-time.
If IP has a value different from -1, then Euclidean norms are calculated with overflow protection by appropriate intermediate scaling.

NZER - INTEGER.

If IP is different from -1, then on output NZER is the number of zeroes in DIAGR; if IP equals -1 then NZER becomes zero.

6. Error indicators and warnings

If the routine is used in its robust version, i.e. IP $\neq -1$, then the number of zeroes on R's diagonal is calculated. If this number is greater than 0 then matrix R is singular. It should be stressed that the reverse is not true; for example, a matrix R with exclusively ones on the diagonal and a small norm may be close to a singular matrix.

7. Auxiliary routines

No auxiliary routines are used.

8. Timing

The time taken is approximately proportional to MN^2.
Time in seconds for various M and N on a CYBER 205 (2-pipe) with IP $= -1$:

	N = 25	N = 50	N = 100	N = 200
M = 50	0.0019	0.0070	*	*
M = 100	0.0024	0.0089	0.0321	*
M = 200	0.0034	0.0129	0.0478	0.1684

Time in seconds for various M and N on a CYBER 205 (2-pipe) with IP $\neq -1$:

	N = 25	N = 50	N = 100	N = 200
M = 50	0.0021	0.0073	*	*
M = 100	0.0027	0.0093	0.0328	*
M = 200	0.0038	0.0135	0.0490	0.1703

9. Storage

There are no internally declared arrays.

10. Accuracy

The measurement for orthogonality of Q, $\|Q^TQ-I\|_2$, is small within working precision and the residual $\|A-QR\|_2 / \|A\|_2$ is equally small.

11. Further comments

None.

11.1. Vectorization information

The routine is written in FORTRAN 200, making use of its vector syntax extensions.

12. Keywords

Householder reflection.
QR factorization.
Overdetermined systems.

13. Example

See EXPLQ.

LINLSQ - NUMVEC FORTRAN Library Routine Document

1. Purpose

LINLSQ calculates the least-squares solution of a full-rank overdetermined linear system by means of a QR-factorization and the solution of a triangular system.

2. Specifications

```
      SUBROUTINE  LINLSQ(A, IA, M, N, B, X, W, RES, NZER)
C     INTEGER  IA, M, N, NZER
C     REAL  A(IA, N), B(M), X(M), W(M+N), RES
```

3. Description

Given a linear system with not more unknowns than equations, an attempt is made to calculate its least-squares solution. First the QR factorization of the coefficient matrix A is calculated using Householder reflections.
If none of the diagonal elements of R is zero, then the unique solution is calculated by means of back substitution with the triangular matrix R.
Moreover, the norm of the residual vector is calculated. For a well-posed problem, this norm should be considerably less than the norm of the original right-hand side.
If one or more diagonal elements of R are zero, then no solution is delivered and the number of zeroes is reported; in that case the user is advised to calculate a minimal-norm solution by means of singular value decomposition as is performed by NUMVEC routine LSQMNS; this chapter.

4. References

[1] Golub, G.H., Van Loan, C.F., Matrix Computations, North Oxford Academic, Oxford, 1983.

[2] Hoffmann, W., Definition and use of Householder reflections, Report CS-88-05, University of Amsterdam, Department of Computer Systems, 1988.

5. Parameters

A - REAL array of DIMENSION (IA,p) where p ≥ N.
Before entry, A should contain the elements of the real matrix.
On exit, it will contain information for the QR factorization as calculated by HSHVOX.

IA - INTEGER.
On entry, IA specifies the first dimension of array A as declared in the calling (sub)program (IA ≥ M).
Unchanged on exit.

M - INTEGER.
On entry, M specifies the number of rows of matrix A.
Unchanged on exit.

N - INTEGER.
On entry, N specifies the number of columns of matrix A.
Unchanged on exit.

B - REAL array of DIMENSION at least (M).
Before entry, B should contain the elements of the right hand side.

Unchanged on exit, but see section 11.

X - REAL array of DIMENSION at least (M).
On exit it will contain the solution vector X in its first N elements.

W - REAL array of DIMENSION at least (M+N).
Used as working space.

RES - REAL.
On exit, RES contains the norm of the residual: $||B-AX||_2$.

NZER - INTEGER variable.
On output NZER contains the number of zero diagonal elements that has been detected during the *QR* factorization.

6. Error indicators and warnings

If NZER is larger then zero, then no solution is calculated; the rank of the matrix is less than or equal to N−NZER. The use of LSQMNS is advised.

7. Auxiliary routines

This routine uses the NUMVEC Library routines HSHVOX, MULQTX, and LSSOLU.

8. Timing

The time taken is approximately proportional to MN^2.
Time in seconds for various M and N on a CYBER 205 (2-pipe):

	N = 25	N = 50	N = 100	N = 200
M = 50	0.0024	0.0078	*	*
M = 100	0.0029	0.0099	0.0340	*
M = 200	0.0041	0.0142	0.0505	0.1735

9. Storage

There are no internally declared arrays.

10. Accuracy

The accuracy of the computed solution depends on the condition of the matrix and on the angle between the right hand side vector and the columnspace of the matrix.

11. Further comments

If the routine is called with the same name for parameters B and X then the solution vector will overwrite the right hand side vector.

11.1. Vectorization information

The routine uses routines which are written in FORTRAN 200, making use of its vector syntax extensions.

12. Keywords

Linear least-squares solution.
Householder reflection.
QR factorization.
Overdetermined systems.

13. Example

To solve the linear least-squares problem for $Ax \approx b$ where

$$A = \begin{bmatrix} -2 & 0 & 0 & 0 \\ 1 & -2 & 0 & 0 \\ 0 & 1 & -2 & 0 \\ 0 & 0 & 1 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 2.01 \\ 3.02 \\ 4.04 \\ 5.08 \\ -3.84 \end{bmatrix}$$

13.1. Program text

```
C
C     LSTSQR EXAMPLE PROGRAM TEXT
C     NUMVEC 1988
C     MARK 1
C
      PROGRAM LSTSQR(OUTPUT,TAPE6=OUTPUT)
      REAL A(5,4), A1(5,4), B(5), X(5), W1(10), W3(5), EPS, RES, SOM
      INTEGER M, N, I, J, NZER, NMAX
      DATA ((A1(I,J),J=1,4),I=1,5),(B(I),I=1,5)
     + / -2., 0., 0., 0.,
     +   1.,-2., 0., 0.,
     +   0., 1.,-2., 0.,
     +   0., 0., 1.,-2.,
     +   0., 0., 0., 1.,
     +   2.01, 3.02, 4.04, 5.08,-3.84 /
      M = 5
      N = 4
      NMAX = 5
      DO 20 J = 1, N
        DO 19 I = 1, M
          A(I,J) = A1(I,J)
   19   CONTINUE
   20 CONTINUE
      WRITE(6,99994)
      DO 40 I = 1, M
        WRITE(6,99995)(A(I,J),J = 1, 4)
   40 CONTINUE
      WRITE(6,99996)(B(I),I = 1, 5)
      CALL LINLSQ(A, NMAX, M, N, B, X, W1, RES, NZER)
      WRITE(6,99997)(X(I),I = 1, 4)
      WRITE(6,99998) RES
      DO 45 I = 1, M
```

```
         W3(I) = B(I)
   45  CONTINUE
       DO 50 J = 1, N
         DO 49 I = 1, M
           W3(I) = W3(I) - A1(I,J) * X(J)
   49    CONTINUE
   50  CONTINUE
       CALL MULQTX(A, NMAX, M, N, W3, W3)
       SOM = 0.
       DO 60 I = 1, N
         SOM = SOM + W3(I) * W3(I)
   60  CONTINUE
       F1 = SQRT(SOM)
       WRITE(6,99999) F1
  997  CONTINUE
       STOP
99994  FORMAT ('1LSTSQR EXAMPLE PROGRAM RESULTS'/'0MATRIX A')
99995  FORMAT (4(1X,F6.2))
99996  FORMAT ('0RIGHT-HANDSIDE VECTOR:'/(1X,F6.2))
99997  FORMAT ('0SOLUTION VECTOR:'/(1X,F6.2))
99998  FORMAT ('0RESIDUAL NORM OF (B - AX) = ',1PE10.3)
99999  FORMAT ('0INNER PRODUCT Q**T(B-AX)  = ',1PE10.3)
       END
```

13.2. Program results

```
LSTSQR EXAMPLE PROGRAM RESULTS

MATRIX A
 -2.00   0.00   0.00   0.00
  1.00  -2.00   0.00   0.00
  0.00   1.00  -2.00   0.00
  0.00   0.00   1.00  -2.00
  0.00   0.00   0.00   1.00

RIGHT-HANDSIDE VECTOR:
  2.01
  3.02
  4.04
  5.08
 -3.84

SOLUTION VECTOR:
 -1.00
 -2.00
 -3.00
 -4.00

RESIDUAL NORM OF (B - AX) =  1.847E-01

INNER PRODUCT Q**T(B-AX)  =  3.047E-13
```

LSSOLU - NUMVEC FORTRAN Library Routine Document

1. Purpose

LSSOLU calculates the solution of a triangular system where the data is delivered as in routines HSHVOX and MULQTX respectively.

2. Specifications

```
      SUBROUTINE  LSSOLU(A, IA, N, DIAG, C, X)
C     INTEGER  IA, N
C     REAL  A(IA, N), DIAG(N), C(N), X(N)
```

3. Description

The solution of a linear system $Rx = c$ with an upper triangular coefficient matrix R and right hand side vector c is solved for x. The strictly upper triangular part of R is given in the corresponding part of A and the diagonal of R is given in DIAG.

4. References

None.

5. Parameters

A - REAL array of DIMENSION (IA,p) where p $\geqslant$ N.
Before entry, A should contain in its strictly upper triangle the corresponding elements of matrix R.
Unchanged on exit.

IA - INTEGER.
On entry, IA specifies the first dimension of array A as declared in the calling (sub)program (IA $\geqslant$ N).
Unchanged on exit.

N - INTEGER.
On entry, N specifies the number of columns of matrix A.
Unchanged on exit.

DIAG - REAL array of DIMENSION at least (N).
Before entry, DIAG should contain the diagonal elements of matrix R.
Unchanged on exit.

C - REAL array of DIMENSION at least (N).
Before entry, C should contain the right hand side vector.
Unchanged on exit, but see section 11.

X - REAL array of DIMENSION at least (N).
On exit, X contains the solution of this system.

6. Error indicators and warnings

None.

7. Auxiliary routines

No auxiliary routines are used.

8. Timing

The time taken is approximately proportional to N^2.

9. Storage

There are no internally declared arrays.

10. Accuracy

The accuracy of the solution depends on the condition of matrix *R*.

11. Further comments

If the routine is called with the same name for parameters B and X then the solution vector will overwrite the right hand side vector.

11.1. Vectorization information

The routine is written in FORTRAN 200, making use of its vector syntax extensions.

12. Keywords

Triangular system.

MULQTX - NUMVEC FORTRAN Library Routine Document

1. Purpose

MULQTX calculates the product $Q^T x$ for a given m-vector x where Q is defined by a sequence of Householder reflections, each one defined by an appropriate Householder vector as calculated by HSHVOX.

2. Specifications

```
      SUBROUTINE  MULQTX(A, IA, M, N, VECIN, VECOUT)
C     INTEGER  IA, M, N
C     REAL  A(IA, N), VECIN(M), VECOUT(M)
```

3. Description

The sequence of Householder matrices which are defined by the columns of the lower trapezoidal part of matrix A are applied to and accumulated in vector x.

4. References

None.

5. Parameters

A - REAL array of DIMENSION (IA,p) where p ⩾ N.
Before entry, A should contain in its lower trapezoidal part the Householder vectors as calculated by HSHVOX.
Unchanged on exit.

IA - INTEGER.
On entry, IA specifies the first dimension of array A as declared in the calling (sub)program (IA ⩾ M).
Unchanged on exit.

M - INTEGER.
On entry, M specifies the number of rows of matrix A.
Unchanged on exit.

N - INTEGER.
On entry, N specifies the number of columns of matrix A.
Unchanged on exit.

VECIN - REAL array of DIMENSION at least (M).
On entry VECIN should contain the given M-vector which is to be multiplied by Q transposed.
Unchanged on exit, but see section 11.

VECOUT - REAL array of DIMENSION at least (M).
On exit, VECOUT contains in its first N elements the result of the matrix-vector multiplication; the remaining M−N elements are used for working space.

6. Error indicators and warnings

None.

7. Auxiliary routines

No auxiliary routines are used.

8. Timing

The time taken is approximately proportional to MN.

9. Storage

There are no internally declared arrays.

10. Accuracy

The accuracy is up to working precision.

11. Further comments

If the routine is called with the same name for parameters VECIN and VECOUT then the output vector will overwrite the input vector.

11.1. Vectorization information

The routine is written in FORTRAN 200, making use of its vector syntax extensions.

12. Keywords

Householder matrices.

13. Example

See LINLSQ.

MULQX - NUMVEC FORTRAN Library Routine Document

1. Purpose

MULQX calculates the product Qx for a given n-vector x where Q is defined by a sequence of Householder reflections, each one defined by an appropriate Householder vector as calculated by HSHVOX.

2. Specifications

```
      SUBROUTINE  MULQX(A, IA, M, N, VECIN, VECOUT)
C     INTEGER  IA, M, N
C     REAL  A(IA, N), VECIN(M), VECOUT(M)
```

3. Description

The sequence of Householder matrices which are defined by the columns of the lower trapezoidal part of matrix A are backward applied to and accumulated in vector x.

4. References

None.

5. Parameters

A - REAL array of DIMENSION (IA,p) where p $\geqslant$ N.
Before entry, A should contain in its lower trapezoidal part the Householder vectors as calculated by HSHVOX.
Unchanged on exit.

IA - INTEGER.
On entry, IA specifies the first dimension of array A as declared in the calling (sub)program (IA $\geqslant$ M).
Unchanged on exit.

M - INTEGER.
On entry, M specifies the number of rows of matrix A.
Unchanged on exit.

N - INTEGER.
On entry, N specifies the number of columns of matrix A.
Unchanged on exit.

VECIN - REAL array of DIMENSION at least (M).
On entry VECIN should contain in its first N elements the vector which is to be multiplied by Q; the remaining M$-$N elements are used for working space.
Unchanged on exit, but see section 11.

VECOUT - REAL array of DIMENSION at least (M).
On exit, VECOUT contains the result of the matrix-vector multiplication.

6. Error indicators and warnings

None.

7. Auxiliary routines

No auxiliary routines are used.

8. Timing

The time taken is approximately proportional to MN.

9. Storage

There are no internally declared arrays.

10. Accuracy

The accuracy is up to working precision.

11. Further comments

If the routine is called with the same name for parameters VECIN and VECOUT then the output vector will overwrite the input vector.

11.1. Vectorization information

The routine is written in FORTRAN 200, making use of its vector syntax extensions.

12. Keywords

Householder matrices.

13. Example

Analogously to the use of MULQTX; see example in section LINLSQ.