Author(s) : H. Lipps **Library:** KERNLIB
Submitter : **Submitted:** 01.09.1983
Language : Fortran or Assembler or COMPASS **Revised:**

Symmetric Positive-Definite Linear Systems

Subroutine tSINV (where t = R or D as described below) computes the inverse of a symmetric positive-definite matrix **A**.

Subroutine tSEQN solves a set of linear equations

$$\mathbf{AX} = \mathbf{B} \qquad (*)$$

whose coefficient matrix **A** is symmetric and positive-definite. The determinant det(**A**) of **A** may be calculated by subroutine tSFACT described below.

If several systems of the form (*) are to be solved with the same **A** but differing **B**, a procedure which is appreciably faster than calling subroutine tSEQN repeatedly is to execute a single call to subroutine tSEQN (or subroutine tSFACT if the determinant is required), and then to call subroutine tSFEQN as many times as required. When the last system (*) has been solved, the inverse matrix $\mathbf{A}^{-1}$, if required, may be computed by calling tSFINV.

Subroutine tSEQN and tSFACT both replace the matrix **A** by a lower triangular matrix **L** and an upper triangular matrix **U** such that $\mathbf{LU} = \mathbf{A}$. This **LU** decomposition is referred to below as **lu**(**A**).

Given **lu**(**A**) and some matrix **B**, subroutine tSFEQN replaces **B** by the solution **X** of equation (*) without changing **lu**(**A**). Subroutine tSFEQN may therefore be called repeatedly with differing **B**.

Given **lu**(**A**), subroutine tSFINV replaces **lu**(**A**) by the inverse $\mathbf{A}^{-1}$ of **A**.

Structure:

SUBROUTINE subprograms
User Entry Names: RSFACT, RSEQN, RSFEQN, RSINV, RSFINV
DSFACT, DSEQN, DSFEQN, DSINV, DSFINV
Files Referenced: Printer
External References: TMPRNT (F011), KERMTR (N001), ABEND (Z035)

Usage:

For t = R (type REAL), t = D (type DOUBLE PRECISION):

```
CALL tSINV (N,A,IDIM,IFAIL)
CALL tSEQN (N,A,IDIM,IFAIL,K,B)
CALL tSFACT(N,A,IDIM,IFAIL,DET,JFAIL)
CALL tSFEQN(N,A,IDIM,K,B)
CALL tSFINV(N,A,IDIM)
```

N (INTEGER) Order of the matrix **A**.

A (Type according to t) Two-dimensional array whose first dimension has the value IDIM.

IDIM (INTEGER) First dimension of array A (and of array B if K > 1).

IFAIL (INTEGER) On exit, IFAIL will be set to 0 if **A** is positive-definite, and to -1 otherwise.

DET (Type according to t) On exit, DET will be set to the value det(**A**) unless JFAIL returns a non-zero value.

JFAIL (INTEGER) On exit, JFAIL will be set to zero if det(**A**) can be safely evaluated. Otherwise JFAIL is set as follows:
$= -2$ if **A** is not positive-definite,
$= -1$ if det(**A**) is probably too small,
$= +1$ if det(**A**) is probably too large.

K (INTEGER) Number of columns of the matrices **B** and **X**.

B (Type according to t) In general, a two-dimensional array whose first dimension has the value IDIM. B may be one-dimensional if K = 1. tSEQN accepts a dummy argument B if K = 0.

The contents of arrays A and B on entry and exit are as follows:

tSINV On entry, **A** must be stored in A. On exit, A contains $\mathbf{A}^{-1}$ if IFAIL = 0, or else is undefined.

tSEQN On entry, **A** must be stored in A and **B** in B. On exit, A contains **lu(A)** and B contains **X** if IFAIL = 0, or else A is undefined and B is unchanged.

tSFACT On entry, **A** must be stored in A. On exit, A contains **lu(A)** if IFAIL = 0, or else is undefined. DET contains det(**A**) if JFAIL = 0, contains zero if JFAIL = −1, and is undefined otherwise.

tSFEQN On entry, **lu(A)** must be stored in A, and **B** in B. On exit, A is unchanged and B contains **X**.

tSFINV On entry, **lu(A)** must be stored in A. On exit, A contains $\mathbf{A}^{-1}$.

Method:

Modified Cholesky factorization (without square roots). See Ref. 1.

Accuracy:

On computers with IBM 370 architecture, inner products are accumulated using double precision arithmetic internally for arrays of type REAL.

Notes:

Only those elements a_{ij} of the original matrix **A** for which $i \geq j$ are required on entry to tSINV, tSEQN and tSFACT.

Error handling:

If N < 1 or IDIM < N or K < 0 (tSEQN) or K < 1 (tSFEQN), a message is printed and program execution is terminated by calling ABEND (Z035).

Examples:

Assume that the 10×10 matrix **A** and the 10×3 matrix **B** are stored according to the Fortran convention in arrays A and B respectively of a program containing the declarations

```
REAL A(25,30),B(25,10)
```

To replace **B** by the 10×3 solution matrix **X** of the system of equations $\mathbf{AX} = \mathbf{B}$, with a jump to label 100 if **A** is not positive definite:

```
CALL RSEQN(10,A,25,IFAIL,3,B)
IF(IFAIL .NE. 0) GO TO 100
```

References:

1. J.H. Wilkinson and C. Reinsch (eds.), Handbook for automatic computation, Vol.2: Linear algebra (Springer-Verlag, New York 1971), Chapter 2.

•