4.6 Solution of a Positive-Definite System with Cholesky Factorization

A. Purpose

This subroutine computes the solution vector \( \mathbf{x} \) for a system of equations of the form

\[
P\mathbf{x} = \mathbf{d},
\]

where \( P \) is an \( N \times N \) positive-definite symmetric matrix, and \( \mathbf{d} \) is an \( N \)-vector. This subroutine also returns the Cholesky factor of \( P \) and thus is applicable where computing the Cholesky factor is the objective.

B. Usage

B.1 Program Prototype, Double Precision

```fortran
DOUBLE PRECISION P(LDP, \geq N), D(\geq N), U, TOL
INTEGER LDP, N, IERR
```

Assign values to \( P(,), \) \( LDP, N, I(\,), U, \) and \( TOL \).

```
CALL DCHOL (P, LDP, N, D, U, TOL, IERR)
```

The solution vector \( \mathbf{x} \) will be stored in \( D(\,). \) Additional computed quantities that may be of interest to the user in some situations will be stored in \( P(,) \) and \( U \).

B.2 Argument Definitions

\( P(,) \) [inout] On entry this array must contain the \( N \times N \) symmetric positive-definite matrix \( P \) of Eq. (1). It suffices to provide only the elements on and above the diagonal. On return this array will contain the \( N \times N \) upper triangular matrix \( F \) defined by Eq. (2) on and above the diagonal positions of the array \( P(,) \). Locations of the array \( P(,) \) below the diagonal will not be referenced or modified by this subroutine.

\( LDP \) [in] Dimension of the first subscript of the storage array \( P(,) \). Require \( LDP \geq N \).

\( N \) [in] Order of the matrix \( P \). Require \( N \geq 1 \).

\( D(\,) \) [inout] On entry \( D(\,) \) must contain the vector \( \mathbf{d} \) of Eq. (1). On return \( D(\,) \) contains the solution vector \( \mathbf{x} \) for Eq. (1).

\( \text{U} \) [inout] If \( \text{U} \) contains the number \( \rho \) of Eq. (10) or (15) respectively on entry, then on return \( \text{U} \) will contain the number \( \rho \) of Eq. (11) or (16) respectively. If the user is not interested in having the number \( \rho \) computed, \( \text{U} \) should be zero on entry and will be unchanged on return.

\( \text{TOL} \) [in] A user-provided relative tolerance parameter to be used in the conditioning test of Eq. (17). We suggest setting \( \text{TOL} \) to a value of \( 10^{-1(k+1)} \) where \( k = \min(\kappa_A, k_h) \). Here \( \kappa_A \) is the user’s estimate of the number of significant decimal digits in the elements of the matrix \( A \) and \( k_h \) is the corresponding estimate for \( \mathbf{b} \). See Eq. (7) or (12) for the definitions of \( A \) and \( \mathbf{b} \). If the \( \text{TOL} \) input is \( < \varepsilon \), where \( \varepsilon \) is the relative machine precision (i.e. the smallest positive number such that \( 1.0+\varepsilon \neq 1.0 \) in the machine’s floating point arithmetic), then \( \varepsilon \) is used for \( \text{TOL} \) internally.

\( \text{IERR} \) [out] On return this is set to 0 if \( t_{\text{min}} \) defined in Eq. (17) is greater than 0. Otherwise results are of questionable validity and \( |\text{IERR}| \) will equal the index of the equation that resulted in the value for \( t_{\text{min}} \). See Section E for more details.

B.3 Modifications for Single Precision

We recommend the use of double precision for this computation except on machines such as the Cray that have \( 10^{-14} \) precision in single precision. To use single precision change \( \text{DCHOL} \) to \( \text{SCHOL} \), and the \( \text{DOUBLE PRECISION} \) type statement to \( \text{REAL} \).

C. Examples and Remarks

Consider the least-squares problem \( A\mathbf{x} \simeq \mathbf{b} \) where \( A \) is the 3 by 2 matrix and \( \mathbf{b} \) is the 3-vector defined by the DATA statements in the program \( \text{DRDCHOL} \) below. This program forms normal equations by computing \( P = A^T A \) and \( \mathbf{d} = A^T \mathbf{b} \). It also computes \( u = \mathbf{b}^T \mathbf{b} \).

It uses the subroutine \( \text{DCHOL} \) to solve the normal equations \( P\mathbf{x} = \mathbf{d} \), and to compute the quantity \( \text{RNORM} = \rho = \| \mathbf{b} - A\mathbf{x} \| \). Output from this program is given in the file \( \text{ODDCHOL} \).

For programming convenience one may prefer to store the matrix \( A \) and the vector \( \mathbf{b} \) together in the same array. The code in \( \text{DRDCHOL2} \) shows how this example can be programmed storing \( A \) and \( \mathbf{b} \) together in the array \( \text{AB}() \) and using the array \( \text{PDU}() \) to hold \( P, \) \( \mathbf{d}, \) and \( u \).

D. Functional Description

Given the problem \( P\mathbf{x} = \mathbf{d} \), where \( P \) is an \( N \times N \) positive-definite symmetric matrix, there exists an upper triangular \( N \times N \) matrix \( F \) satisfying

\[
F^T F = P
\]
Eq. (2) defines the Cholesky decomposition of $P$. The upper triangular elements of $F$ will be computed from those of $P$ by the following equations, where $i = 1, \ldots, N$.

$$
\begin{align*}
    g_i &= p_{i,i} - \sum_{k=1}^{i-1} f_{k,i}^2 \\
    f_{i,i} &= g_i^{1/2} \\
    p_{i,j} &= \sum_{k=1}^{i-1} f_{k,i} f_{k,j} \\
    f_{i,j} &= \frac{p_{i,j}}{f_{i,i}}, \quad j = i + 1, \ldots, N
\end{align*}
$$

(3) (4) (5)

In these formulas the summation is to be skipped when $i = 1$.

After computing $F$ the subroutine solves the lower triangular system of equations,

$$
F^T y = d
$$

and then computes the vector $x$ which satisfies $Px = d$ by solving the upper triangular system

$$
Fx = y
$$

Besides computing the solution vector $x$ this subroutine uses the input number $u$ given in the Fortran variable $U$ to compute

$$
\rho = \left( \max(0, u - y^T y) \right)^{1/2}
$$

(6)

This number $\rho$ is stored in $U$ on return. If the problem $Px = d$ arose as the system of normal equations for a least-squares problem and if $u$ was computed appropriately by the user then $\rho$ represents the norm of the residual vector for the least-squares problem.

Specifically if the user wishes to solve the least-squares problem of minimizing

$$
\|b - Ax\| = \left( (b - Ax)^T (b - Ax) \right)^{1/2}
$$

(7)

then $P$, $d$, and $u$ should be initialized as

$$
\begin{align*}
    P &= A^T A \\
    d &= A^T b \\
    u &= b^T b
\end{align*}
$$

(8) (9) (10)

Then theoretically, the quantity, $u - y^T y$ of Eq. (6) will be nonnegative and the number $\rho$ of Eq. (6) will have the interpretation

$$
\rho = \|b - Ax\|
$$

(11)

More generally, if the user is solving the weighted least-squares problem of minimizing

$$
\left( (b - Ax)^T W (b - Ax) \right)^{1/2}
$$

(12)

where $W$ is a positive definite symmetric matrix, then $P$, $d$, and $u$ should be initialized as

$$
\begin{align*}
    P &= A^T W A \\
    d &= A^T W b \\
    u &= b^T W b
\end{align*}
$$

(13) (14) (15)

Then, theoretically, the quantity $u - y^T y$ of Eq. (6) will be nonnegative and the number $\rho$ of Eq. (6) will have the interpretation

$$
\rho = \left( (b - Ax)^T W (b - Ax) \right)^{1/2}
$$

(16)

The Cholesky factor matrix $F$ will appear in the upper triangular portion of the array $P(,)$ on return. If $\text{IERR} \geq 0$, the user can input this matrix $F$ to the library subroutine DCOV2 of Chapter 4.2 to compute the unscaled covariance matrix for the associated least-squares problem. This requires building the IP() array: IP(I) = I, for $I = 1, \ldots, N$.

Theoretically the numbers $g_i$ of Eq. (3) will be strictly positive for all $i$ if and only if the symmetric matrix $P$ is positive-definite. If all $g_i$ are positive but the ratio $g_i / p_{i,i}$ is very small for some $i$ this is an indication that the problem is ill-conditioned. The square of the relative tolerance parameter $\text{TOL}$ is used to test this ratio. Let

$$
t_{\text{min}} = \min_{1 \leq i \leq N} \left\{ \frac{g_i - (\text{TOL})^2}{|p_{i,i}|} \right\}.
$$

(17)

If $t_{\text{min}} \geq 0$, then $\text{IERR}$ is set to 0. Otherwise let $m$ be a value of $i$ that gives the minimum value in Eq. (17). Then $\text{IERR}$ is set to $m$ if $g_m > 0$, and is set to $-m$ otherwise. See Section E below for more details.

If one knows or suspects that the least-squares problem is ill-conditioned it is suggested that the Singular Value Analysis subroutine, Chapter 4.3, be used to obtain a more complete analysis and a more reliable solution for the problem.

A nonnegative definite symmetric matrix has a Cholesky factor even if it is singular. In computing a Cholesky factor for such matrices this subroutine does the following: If $g_i$ of Eq. (3) is nonpositive, Eqs. (4–5) are replaced by

$$
f_{i,j} = 0, \quad j = i, i + 1, \ldots, N
$$

(18)

When solving the triangular systems below Eq. (5), if $f_{i,i} = 0$ the solution components $y_i$ and $x_i$ are set to zero.
If $P$ is a singular nonnegative definite matrix, the matrix $F$ produced in this way is its (nonunique) Cholesky factor, i.e., it satisfies Eq. (2). In such a case Eq. (1) may or may not have a solution and the vector $x$ produced in this way is the solution only if a solution exists.

E. Error Procedures and Restrictions

If $t_{\text{min}} < 0$ in Eq. (17), the subroutine sets IERR nonzero as indicated above. When IERR < 0, at least one row of the augmented matrix [upper triangle of $P$, $D$] will have been set to zero.

If IERR $\neq 0$ we suggest that the user apply the Singular Value Analysis subroutine, Chapter 4.3, to the associated least-squares problem.

F. Supporting Information

The source language is ANSI Fortran 77.

Entry Required Files
DCHOL AMACH, DCHOL
SCHOL AMACH, SCHOL


DRDCHOL

```fortran
program DRDCHOL
>>> 1994–10–19 DRDCHOL Krogh Changes to use M77CON
>>> 1994–08–09 DRDCHOL WVS remove '0' from formats

integer LDP, M, N
parameter (M = 3, N = 2, LDP = 2)
integer I, IERR, J
double precision A(M,N), B(M)
external DDOT
double precision P(LDP,LDP), D(LDP), U, DDOT
data A(1,1), A(1,2), B(1) / 0.7D0, 0.6D0, 1.726D0 /
data A(2,1), A(2,2), B(2) / -0.8D0, 0.5D0, -5.415D0 /
data A(3,1), A(3,2), B(3) / 0.6D0, -0.7D0, 5.183D0 /

U = DDOT(M, B, 1, B, 1)
do 20 I = 1, N
   D(1) = DDOT(M, A(1,1), 1, B, 1)
do 10 J = 1, N
   P(I,J) = DDOT(M, A(1,I), 1, A(1,J), 1)
10      continue
20      continue
call DCHOL(P, LDP, N, D, U, 0.0d0, IERR)
print '(X() = ' ,2f15.6)', D(1), D(2)
print '(RNORM = ' ,f15.6)', U
if (IERR .ne. 0) print '( Matrix failed conditioning test in DCHOL, IERR = '',I3'',I3)' ,IERR
end
```

ODDCHOL

```
X() = 5.000000 -3.000000
RNORM = 0.121614
```

July 11, 2015 Solution of a Positive-Definite System with Cholesky Factorization 4.6–3
DRDCHOL2

>>> 1994–10–19 DRDCHOL2 Krogh Changes to use M77CON
>>> 1993–02–18 DRDCHOL2 CLL.

Demonstration driver for DCHOL

---D replaces "?": DR?CHOL2, ?CHOL, ?DOT

c integer LDPDU, M, N, NP1
parameter (M = 3, N = 2, NP1 = N+1, LDPDU = 3)
integer I, IERR, J
double precision AB(M,NP1)
external DDOT
double precision PDU(LDPDU,LDPDU), DDOT
data AB(1,1), AB(1,2), AB(1,3) / 0.7D0, 0.6D0, 1.726D0 /
data AB(2,1), AB(2,2), AB(2,3) / -0.8D0, 0.5D0, -5.415D0 /
data AB(3,1), AB(3,2), AB(3,3) / 0.6D0, -0.7D0, 5.183D0 /

do 20 I = 1, NP1
   do 10 J = 1, NP1
      PDU(I,J) = DDOT(M, AB(1,I), 1, AB(1,J), 1)
   10 continue
20 continue
call DCHOL(PDU, LDPDU, N, PDU(1,NP1), PDU(NP1,NP1), 0.0d0, IERR)
print '( "X( ) = " ,2f15.6 )', PDU(1,NP1), PDU(NP1,NP1)
print '( "RNORM = " ,f15.6 )', PDU(NP1,NP1)
if (IERR .ne. 0) print '( + Matrix failed conditioning test in DCHOL, IERR = ',I3),' ,IERR
end

ODDCHOL2

X( ) = 5.000000 -3.000000
RNORM = 0.121614