



Cholesky factorization

Nicholas J. Higham*

This article aimed at a general audience of computational scientists, surveys the Cholesky factorization for symmetric positive definite matrices, covering algorithms for computing it, the numerical stability of the algorithms, and updating and downdating of the factorization. Cholesky factorization with pivoting for semidefinite matrices is also treated. © 2009 John Wiley & Sons, Inc. *WIREs Comp Stat* 2009 1 251–254

INTRODUCTION

A symmetric $n \times n$ matrix A is positive definite if the quadratic form $x^T A x$ is positive for all non-zero vectors x or, equivalently, if all the eigenvalues of A are positive. Positive definite matrices have many important properties, not least that they can be expressed in the form $A = X^T X$ for a non-singular matrix X . The Cholesky factorization is a particular form of this factorization in which X is upper triangular with positive diagonal elements; it is usually written as $A = R^T R$ or $A = L L^T$ and it is unique. In the case of a scalar ($n = 1$), the Cholesky factor R is just the positive square root of A . However, R should, in general, not be confused with the square roots of A , which are the matrices Y such that $A = Y^2$, among which there is a unique symmetric positive definite square root, denoted $A^{1/2}$ (Section 1.7 in Ref 1).

The Cholesky factorization (sometimes called the Cholesky decomposition) is named after André-Louis Cholesky (1875–1918), a French military officer involved in geodesy.² It is commonly used to solve the normal equations $A^T A x = A^T b$ that characterize the least squares solution to the overdetermined linear system $A x = b$.

A variant of Cholesky factorization is the factorization $A = L D L^T$, where L is unit lower triangular (i.e., has unit diagonal) and D is diagonal. This factorization exists and is unique for positive definite matrices. If D is allowed to have non-positive diagonal entries, the factorization exists for some (but not all) indefinite matrices. When A is positive definite the Cholesky factor is given by $R = D^{1/2} L^T$.

COMPUTATION

The Cholesky factorization can be computed by a form of Gaussian elimination that takes advantage of the symmetry and definiteness. Equating (i, j) elements in the equation $A = R^T R$ gives

$$\begin{aligned} j = i: \quad a_{ii} &= \sum_{k=1}^i r_{ki}^2, \\ j > i: \quad a_{ij} &= \sum_{k=1}^i r_{ki} r_{kj}. \end{aligned} \quad (1)$$

These equations can be solved to yield R a column at a time, according to the following algorithm:

```

for j = 1:n
  for i = 1:j - 1
     $r_{ij} = (a_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}) / r_{ii}$ 
  end
   $r_{jj} = (a_{jj} - \sum_{k=1}^{j-1} r_{kj}^2)^{1/2}$ 
end

```

The positive definiteness of A guarantees that the argument of the square root in this algorithm is always positive and hence that R has a real, positive diagonal. The algorithm requires $n^3/3 + O(n^2)$ flops and n square roots, where a flop is any of the four elementary scalar arithmetic operations $+$, $-$, $*$, and $/$.

The algorithm above is just one of many ways of arranging Cholesky factorization and can be identified as the ‘jik’ form based on the ordering of the indices of the three nested loops. There are five other orderings, yielding algorithms that are mathematically equivalent but that have quite different efficiency for large dimensions depending on the computing environment, by which we mean both the programming language and the hardware. In

*Correspondence to: higham@ma.man.ac.uk

School of Mathematics, The University of Manchester, Manchester M13 9PL, UK

DOI: 10.1002/wics.018

modern libraries such as LAPACK,³ the factorization is implemented in partitioned form, which introduces another level of looping in order to extract the best performance from the memory hierarchies of modern computers. To illustrate, we describe a partitioned Cholesky factorization algorithm. For a given block size r , we can write

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{12}^T & A_{22} \end{bmatrix} = \begin{bmatrix} R_{11}^T & 0 \\ R_{12}^T & I_{n-r} \end{bmatrix} \begin{bmatrix} I_r & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ 0 & I_{n-r} \end{bmatrix}, \quad (2)$$

where A_{11} and R_{11} are $r \times r$. One step of the algorithm consists of computing the Cholesky factorization $A_{11} = R_{11}^T R_{11}$, solving the multiple right-hand side triangular system $R_{11}^T R_{12} = A_{12}$ for R_{12} , and then forming the Schur complement $S = A_{22} - R_{12}^T R_{12}$; this procedure is repeated on S . This partitioned algorithm does precisely the same arithmetic operations as any other variant of Cholesky factorization, but it does the operations in an order that permits them to be expressed as matrix operations. The block operations defining R_{12} and S are level 3 BLAS operations,⁴ for which efficient computational kernels are available on most machines. In contrast, a *block* LDL^T factorization (the most useful form of block factorization for a symmetric positive definite matrix) has the form $A = LDL^T$, where

$$L = \begin{bmatrix} I & & & \\ L_{21} & I & & \\ \vdots & & \ddots & \\ L_{m1} & \dots & L_{m,m-1} & I \end{bmatrix}, \quad D = \text{diag}(D_{ii}), \quad (3)$$

where the diagonal blocks D_{ii} are, in general, full matrices. This factorization is mathematically different from a Cholesky or LDL^T factorization (in fact, for an indefinite matrix, it may exist when the factorization with 1×1 blocks does not). It is of most interest when A is block tridiagonal⁵ [Chapter 13].

Once a Cholesky factorization of A is available, it is straightforward to solve a linear system $Ax = b$. The system is $R^T R x = b$, which can be solved in two steps, costing $2n^2$ flops:

1. Solve the lower triangular system $R^T y = b$,
2. Solve the upper triangular system $R x = y$.

NUMERICAL STABILITY

Rounding error analysis shows that Cholesky factorization has excellent numerical stability properties. We will state two results in terms of the vector

2-norm $\|x\|_2 = (x^T x)^{1/2}$ and the corresponding subordinate matrix norm $\|A\|_2 = \max_{x \neq 0} \|Ax\|_2 / \|x\|_2$, where for symmetric A we have $\|A\|_2 = \max\{|\lambda_i| : \lambda_i \text{ is an eigenvalue of } A\}$. If the factorization runs to completion in floating point arithmetic, with the argument of the square root always positive, then the computed \hat{R} satisfies

$$\hat{R}^T \hat{R} = A + \Delta A_1, \quad \|\Delta A_1\|_2 \leq c_1 n^2 u \|A\|_2, \quad (4)$$

where the subscripted c denotes a constant of order 1 and u is the unit roundoff (or machine precision). Most modern computing environments use IEEE double precision arithmetic, for which $u = 2^{-53} \approx 1.1 \times 10^{-16}$. Moreover, the computed solution \hat{x} to $Ax = b$ satisfies

$$(A + \Delta A_2) \hat{x} = b, \quad \|\Delta A_2\|_2 \leq c_2 n^2 u \|A\|_2. \quad (5)$$

This is a backward error result that can be interpreted as saying that the computed solution \hat{x} is the true solution to a slightly perturbed problem. The factorization is guaranteed to run to completion if $c_3 n^{3/2} \kappa_2(A) u < 1$, where $\kappa_2(A) = \|A\|_2 \|A^{-1}\|_2 \geq 1$ is the matrix condition number with respect to inversion. By applying standard perturbation theory for linear systems to Eq. (5), a bound is obtained for the forward error:

$$\frac{\|x - \hat{x}\|_2}{\|x\|_2} \leq \frac{c_2 n^2 \kappa_2(A) u}{1 - c_2 n^2 \kappa_2(A) u}. \quad (6)$$

The excellent numerical stability of Cholesky factorization is essentially due to the equality $\|A\|_2 = \|R^T R\|_2 = \|R\|_2^2$, which guarantees that R is of bounded norm relative to A . For proofs of these results and more refined error bounds, see Ref 5 [Chapter 10].

SEMIDEFINITE MATRICES

A symmetric matrix A is positive semidefinite if the quadratic form $x^T A x$ is non-negative for all x ; thus A may be singular. For such matrices a Cholesky factorization $A = R^T R$ exists, now with R possibly having some zero elements on the diagonal, but the diagonal of R may not display the rank of A . For example,

$$A = \begin{bmatrix} 1 & -1 & 1 \\ -1 & 1 & -1 \\ 1 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & -1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} = R^T R, \quad (7)$$

and A has rank 2 but R has only one non-zero diagonal element. However, with P , the permutation matrix comprising the identity matrix with its columns in reverse order, $P^TAP = R_1^TR_1$, where

$$R_1 = \begin{bmatrix} \sqrt{2} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \\ 0 & 0 & 0 \end{bmatrix}. \quad (8)$$

More generally, any symmetric positive semidefinite A has a factorization

$$P^TAP = R^TR, \quad R = \begin{bmatrix} R_{11} & R_{12} \\ 0 & 0 \end{bmatrix}, \quad (9)$$

where P is a permutation matrix, R_{11} is $r \times r$ upper triangular with positive diagonal elements, and $\text{rank}(A) = r$. This factorization is produced by using complete pivoting, which at each stage permutes the largest diagonal element in the active submatrix into the pivot position. The following algorithm implements Cholesky factorization with complete pivoting and overwrites the upper triangle of A with R . It is a 'kji' form of the algorithm.

```

Set  $p_i = 1, i = 1:n$ .
for  $k = 1:n$ 
  Find  $s$  such that  $a_{ss} = \max_{k \leq i \leq n} a_{ii}$ .
  Swap rows  $k$  and  $s$  of  $A$ 
  and swap  $p_k$  and  $p_s$ .
   $a_{kk} = \sqrt{a_{kk}}$ 
  for  $j = k+1:n$ 
     $a_{kj} = a_{kj}/a_{kk}$ 
  end
  for  $j = k+1:n$ 
    for  $i = k+1:j$ 
       $a_{ij} = a_{ij} - a_{ki}a_{kj}$ 
    end
  end
end
Set  $P$  to the matrix whose  $j$ th column
is the  $p_j$ th column of  $I$ .
```

An efficient implementation of this algorithm that uses level 3 BLAS⁶ is available in LAPACK Version 3.2. Complete pivoting produces a matrix R that satisfies the inequalities

$$r_{kk}^2 \geq \sum_{i=k}^{\min(j,r)} r_{ij}^2, \quad j = k+1:n, \quad k = 1:r, \quad (10)$$

which imply $r_{11} \geq r_{22} \geq \dots \geq r_{nn}$.

An important use of Cholesky factorization is for testing whether a symmetric matrix is positive definite.

The test is simply to run the Cholesky factorization algorithm and declare the matrix positive definite if the algorithm completes without encountering any negative or zero pivots and not positive definite otherwise. This test is much faster than computing all the eigenvalues of A , and it can be shown to be numerically stable: the answer is correct for a matrix $A + \Delta A$ with ΔA satisfying Eq. (4).⁷ When an attempted Cholesky factorization breaks down with a non-positive pivot, it is sometimes useful to compute a vector p such that $p^TAp \leq 0$. In optimization, when A is the Hessian of an underlying function to be minimized, p is termed a direction of negative curvature. Such a p is the first column of the matrix

$$Z = \begin{bmatrix} R_{11}^{-1}R_{12} \\ -I \end{bmatrix}, \quad (11)$$

where $[R_{11} \ R_{12}]$ is the partially computed Cholesky factor, and this choice makes p^TAp equal to the next pivot, which is non-positive by assumption. This choice of p is not necessarily the best that can be obtained from Cholesky factorization, either in terms of producing a small value of p^TAp or in terms of the effects of rounding errors on the computation of p . Indeed this is a situation where Cholesky factorization with complete pivoting can profitably be used. For more details, see Ref 8.

UPDATING AND DOWNDATING

In some applications, it is necessary to modify a Cholesky factorization $A = R^TR$ after a rank 1 change to the matrix A . Specifically, given a vector x such that $A - xx^T$ is positive definite we may need to compute \tilde{R} such that $A - xx^T = \tilde{R}^T\tilde{R}$ (the downdating problem), or given a vector y we may need to compute \tilde{R} such that $A + yy^T = \tilde{R}^T\tilde{R}$. For the updating problem, we can write

$$\begin{aligned} A + yy^T &= R^TR + yy^T = \begin{bmatrix} R^T & y \end{bmatrix} \begin{bmatrix} R \\ y^T \end{bmatrix} \\ &= \begin{bmatrix} R^T & y \end{bmatrix} Q^T \cdot Q \begin{bmatrix} R \\ y^T \end{bmatrix}, \quad Q^TQ = I. \end{aligned} \quad (12)$$

We aim to use the orthogonal matrix Q to restore triangularity; thus, for $n = 3$, for example, we want Q to effect, pictorially,

$$Q \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ \times & \times & \times \end{bmatrix} = \begin{bmatrix} \times & \times & \times \\ 0 & \times & \times \\ 0 & 0 & \times \\ 0 & 0 & 0 \end{bmatrix}, \quad (13)$$

where \times denotes a non-zero element. This can be achieved by taking Q as a product of suitably chosen Givens rotations. The downdating problem is more delicate because of possible cancellation in removing

xx^T from A , and several methods are available, all more complicated than the updating procedure outlined above. For more on updating and downdating, see Ref 9 [Section 3.3].

REFERENCES

1. Higham NJ. *Functions of Matrices: Theory and Computation*. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2008, xx+425 pp. (ISBN 978-0-898716-46-7).
2. Brezinski C. The life and work of André Cholesky. *Numer Algorithms* 2006, 43:279–288.
3. Anderson E, Bai Z, Bischof CH, Blackford S, Demmel JW, et al. *LAPACK Users' Guide*. 3rd ed. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1999, xxvi+407 pp. (ISBN 0-89871-447-8).
4. Dongarra JJ, Du Croz JJ, Duff IS, Hammarling SJ. A set of Level 3 basic linear algebra subprograms. *ACM Trans Math Softw* 1990, 16:1–17.
5. Higham NJ. *Accuracy and Stability of Numerical Algorithms*. 2nd ed. Philadelphia, PA: Society for Industrial and Applied Mathematics; 2002, xxx+680 pp. (ISBN 0-89871-521-0).
6. Hammarling S, Higham NJ, Lucas C. LAPACK-style codes for pivoted Cholesky and QR updating. In: Kågström B, Elmroth E, Dongarra J, Waśniewski J, eds. *Applied Parallel Computing. State of the Art in Scientific Computing. 8th International Workshop, PARA 2006, Number 4699 in Lecture Notes in Computer Science*. Berlin: Springer-Verlag; 2007, 137–146.
7. Higham NJ. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra Appl* 1988, 103:103–118.
8. Guo CH, Higham NJ, Tisseur F. *An Improved Arc Algorithm for Detecting Definite Hermitian Pairs*, MIMS EPrint 2008.115. Manchester: Manchester Institute for Mathematical Sciences, The University of Manchester; 2008.
9. Björck A. *Numerical Methods for Least Squares Problems*. Society for Industrial and Applied Mathematics; Manchester, UK, 1996, xvii+408 pp. (ISBN 0-89871-360-9).