2009 International Symposium on Nonlinear Theory and its Applications
NOLTA'09, Sapporo, Japan, October 18-21, 2009

Accurate and Robust Inverse Cholesky Factorization

Takeshi Ogita† and Shin'ichi Oishi‡

†Department of Mathematical Sciences, Tokyo Woman's Christian University
2-6-1 Zempukuji, Suginami-ku, Tokyo 167-8585, Japan
‡Department of Applied Mathematics, Faculty of Science and Engineering, Waseda University
Tokyo 169-8555, Japan
E-mail: ogita@lab.twcu.ac.jp, oishi@waseda.jp

Abstract—In this paper, an algorithm of matrix factorization based on Cholesky factorization for extremely ill-conditioned matrices is proposed. The Cholesky factorization is widely used for solving a system of linear equations whose coefficient matrix is symmetric and positive definite. However, it sometimes breaks down by the presence of an imaginary root due to the accumulation of rounding errors. To overcome this, a robust algorithm named inverse Cholesky factorization is investigated, which never breaks down as long as the matrix is symmetric and positive definite. Numerical results are also presented.

1. Introduction

Matrix factorizations such as LU, QR and Cholesky factorizations are frequently discussed in numerical linear algebra since they are used as building blocks of scientific computing. Following the previous paper [6] by the first author, we propose an algorithm for accurately computing an inverse Cholesky factorization of a real symmetric and positive definite $n \times n$ matrix A, especially for A being extremely ill-conditioned, i.e. the condition number of A is beyond the reciprocal of working precision.

Let $\mathbf{u}$ denote the unit roundoff of floating-point arithmetic, which is equal to the working precision in use. In IEEE standard 754 double precision, $\mathbf{u} = 2^{-53}$. Let $\kappa(A) := \|A\| \cdot \|A^{-1}\|$ as the condition number of A, where $\|\cdot\|$ stands for Euclidean norm throughout the paper. We consider to treat the case where

$$\kappa(A) \gg \mathbf{u}^{-1}.$$

This means no correct digit can be expected in an approximate solution $\widetilde{x}$ when solving a linear system $Ax = b$ in working precision $\mathbf{u}$.

In our previous work [6], we have presented algorithms for accurately calculating inverse LU and inverse QR factorizations. In those algorithms we rely on the fact that standard numerical algorithms for LU and QR using pure floating-point arithmetic rarely break down due to the rounding error, which works as some kind of the regularization. However, when a standard Cholesky factorization such as $A = \widehat{R}^T\widehat{R}$ for an ill-conditioned matrix A is executed, it sometimes breaks down by the presence of an imaginary root due to the accumulation of rounding errors, even if A is actually positive definite. Namely, we cannot directly apply our proposed framework in [6] to the Cholesky factorization.

In this paper, we suggest to compute a good approximate inverse X of the Cholesky factor $\widehat{R}^{-1}$ satisfying

$$A^{-1} \approx XX^T.$$

The proposed algorithm is completely stable in the sense of numerical computations, i.e., barring the presence of underflow or overflow, it never fails as long as the matrix is symmetric and positive definite. In other words, if the algorithm breaks down, then the matrix is proved to be not positive definite.

With the same spirit as the previous work [6] and Rump's method for inverting extremely ill-conditioned matrices [9, 12], we emphasize that pure floating-point arithmetic and standard numerical algorithms are utilized as much as possible. The only one exception is that we need an algorithm of accurately computing dot products, more precisely, as if computed in k-fold working precision and rounded into ℓ pieces of working precision floating-point numbers for any $k \geq 2$ and $1 \leq \ell \leq k$. For example, such accurate dot product algorithms have been developed in [7, 12, 14, 15] for the purpose, and they are very fast.

The rest of the paper is organized as follows: In the following section, we state notation and definitions used in this paper. In Section 3, we present a concrete algorithm of an accurate inverse Cholesky factorization. Finally, some numerical results are presented for illustrating the performance of our proposed algorithm in Section 4.

2. Notation and definitions

For real matrices $A = (a_{ij}), B = (b_{ij}) \in \mathbb{R}^{m\times n}$, we denote by $|A| = (|a_{ij}|) \in \mathbb{R}^{m\times n}$ a nonnegative matrix consisting of entrywise absolute values, and an inequality $A \leq B$ is understood entrywise, i.e., $a_{ij} \leq b_{ij}$ for all (i, j). Moreover, the notation $A \geq O$ (or $A > O$) means that all elements of A are nonnegative (positive). Similar notation applies to real vectors.

For constructing a completely stable algorithm of an inverse Cholesky factorization, a little knowledge of interval arithmetic is required.

Let $\langle a, r\rangle$ denote an interval of the midpoint-radius representation such that

$$\langle a, r\rangle := \{x \in \mathbb{R} : |x - a| \le r\}$$

for some $a \in \mathbb{R}$, $0 \le r \in \mathbb{R}$. Let $\langle b, s\rangle$ denote an interval in a similar way. Then the basic operations (addition, subtraction and multiplication) are defined as follows (See, e.g. [4, 1, 5] for details):

$$\begin{aligned}
\langle a, r\rangle + \langle b, s\rangle &:= \langle a + b, r + s\rangle \\
\langle a, r\rangle - \langle b, s\rangle &:= \langle a - b, r + s\rangle \\
\langle a, r\rangle \cdot \langle b, s\rangle &:= \langle ab, |a|s + r|b| + rs\rangle
\end{aligned}$$

Note that the above operations for interval numbers with the midpoint-radius representation can efficiently be extended to those for interval matrices [11] due to the second author of this paper.

For later use, we define the magnitude of an interval matrix $\langle X, Y\rangle$ by

$$\mathrm{mag}\langle X, Y\rangle := |X| + Y.$$

For readability we denote by $\varphi(\gamma)$ a constant such as $\varphi(\gamma) = c \cdot \gamma$ where c is a constant of $O(1)$ with $0 < c \ll \mathbf{u}^{-1}$.

3. Accurate inverse Cholesky factorization

In this section, we present an algorithm of computing an accurate inverse Cholesky factorization.

3.1. Accurate dot product

Let $\mathbb{F}$ be a set of floating-point numbers in working precision, e.g. double precision. Let $x, y \in \mathbb{F}^n$. We assume that an accurate computation of a dot product $x^T y$ is available as the tool of obtaining $s_\ell := \sum_{i=1}^{\ell} s^{(i)}$ with $s^{(i)} \in \mathbb{F}$ such that

$$\left|\frac{x^T y - s_\ell}{x^T y}\right| \le \varphi(\mathbf{u}^\ell) + \varphi(\mathbf{u}^k)\mathrm{cond}(x^T y),\ k \ge 2,\ 1 \le \ell \le k$$

for $x^T y \neq 0$. Here $\mathrm{cond}(x^T y)$ is the condition number of dot product [7] defined by

$$\mathrm{cond}(x^T y) := 2\frac{|x^T||y|}{|x^T y|}, \quad x^T y \neq 0.$$

This means we can calculate $x^T y$ as if computed in k-fold working precision and rounded into ℓ pieces of working precision floating-point numbers s_i, $1 \le i \le \ell$. Fortunately, we already have such accurate dot product algorithms proposed in [7, 12, 14, 15] at hand. In this paper, the cases of $\ell = 1$, $\ell = \lceil k/2 \rceil$ and $\ell = k$ appear.

Throughout the paper, we use the notation

$$C_\ell = \{A \cdot B\}_k^\ell$$

which satisfies

$$|AB - C_\ell| \le \varphi(\mathbf{u}^\ell)|AB| + \varphi(\mathbf{u}^k)|A||B| \tag{1}$$

for $A \in \mathbb{R}^{m\times p}$ and $B \in \mathbb{R}^{p\times n}$. In the case of $\ell = 1$, we abbreviate it as $C = \{A \cdot B\}_k$.

3.2. Algorithm

We have discussed about the accuracy of LU and QR factorizations in [6], respectively. Similarly, we should consider how to define that of a Cholesky factorization.

Let $A = A^T \in \mathbb{F}^{n\times n}$ with $a_{ii} > 0$ for $1 \le i \le n$. Suppose a standard numerical Cholesky factorization of A runs to completion. Here "run to completion" means that no imaginary root appears in the factorization process. Throughout the paper, the Matlab-style notation

$$R = \mathtt{chol}(A)$$

means a floating-point Cholesky factorization of A such that

$$A \approx R^T R,$$

where R is an upper triangular matrix. Then it is known [3] that the computed Cholesky factor R always satisfies

$$\frac{\|A - R^T R\|}{\|A\|} \le \varphi(\mathbf{u}).$$

Thus it is similar to the case of the LU factorization that not so much information on the accuracy of the Cholesky factorization can be obtained from the residual norm $\|A - R^T R\|$. Thus we again need an another criterion.

Suppose the exact Cholesky factorization $A = \widehat{R}^T \widehat{R}$ runs to completion. Then it holds that

$$\kappa(\widehat{R}) = \sqrt{\kappa(A)}.$$

On the other hand, by some kind of 'regularization' due to the rounding errors in floating-point arithmetic, heuristics tells us that a computed factor R satisfies

$$\kappa(R) \approx \sqrt{\min\{\kappa(A), \mathbf{u}^{-1}\}}$$

for any (positive definite) matrix A as long as the (numerical) Cholesky factorization of A runs to completion. Let $\widetilde{x}$ be an approximate solution of a linear system $Ax = b$. If $\kappa(R) \approx \sqrt{\kappa(A)} < \sqrt{\varepsilon_{\mathrm{tol}} \cdot \mathbf{u}^{-1}}$, then it holds that

$$\frac{\|A^{-1}b - \widetilde{x}\|}{\|A^{-1}b\|} \lesssim \varepsilon_{\mathrm{tol}}.$$

However, as mentioned before, the Cholesky factorization of A using floating-point arithmetic sometimes breaks down due to an accumulation of the rounding errors. To avoid the break-down, a diagonal shift applies to A such as $A + \delta I \in \mathbb{F}^{n\times n}$ for some $\delta > 0$. Let $\lambda_{\max}$ and $\lambda_{\min}$ be the largest and the smallest eigenvalues of A, respectively. In the case of $0 < \lambda_{\min} \ll \delta = \alpha\lambda_{\max}$, it holds that

$$\kappa(A + \delta I) = \frac{\lambda_{\max} + \delta}{\lambda_{\min} + \delta} \approx \frac{\lambda_{\max}}{\delta} = \alpha^{-1}. \tag{2}$$

On the other hand, in case of $\delta \ll \lambda_{\min}$, it holds that

$$\kappa(A + \delta I) \approx \frac{\lambda_{\max}}{\lambda_{\min}} = \kappa(A). \tag{3}$$

Let $\widetilde{R}$ be a computed Cholesky factor of $A + \delta I$, i.e.,

$$\widetilde{R} = \mathtt{chol}(A + \delta I).$$

Combining (2) and (3) yields

$$\kappa(\widetilde{R}) \approx \sqrt{\min\{\kappa(A), \alpha^{-1}\}}. \tag{4}$$

The problem is how to choose a suitable constant α. In the accurate inverse LU factorization in [6], it holds that

$$\kappa(AU^{-1}) \approx \mathbf{u} \cdot \kappa(A),$$

where U is an upper triangular matrix obtained by an LU factorization of A. Similarly, the constant α in (4) becomes a drop factor for the condition number of A such that

$$\kappa(\widetilde{R}^{-T} A \widetilde{R}^{-1}) \approx \alpha \cdot \kappa(A).$$

So, it is desirable to choose α as small as possible.

By the backward error analysis of the floating-point Cholesky factorization [16, 2], we have

$$A + \Delta = R^T R, \quad |\Delta| \lesssim n\mathbf{u}|R^T||R| \approx n\mathbf{u}|A|.$$

In [13], Rump has modified and utilized it for verification of positive definiteness:

$$A + \Delta = R^T R, \quad \|\Delta\| \le \gamma\mathbf{u} \cdot \mathrm{tr}(A),$$

where γ is some computable constant and $\gamma \approx n\mathbf{u}$ (see [13]). Let δ be defined by

$$\delta := \min\{\tau \mid A + \tau I \in \mathbb{F}^{n\times n},\ \tau \ge \gamma\mathbf{u} \cdot \mathrm{tr}(A)\}.$$

As long as A is positive definite, $\mathtt{chol}(A + \delta I)$ never breaks down and

$$(A + \delta I) + \widetilde{\Delta} = \widetilde{R}^T \widetilde{R}, \quad \|\widetilde{\Delta}\| \le \delta.$$

In other words, if $\mathtt{chol}(A + \delta I)$ fails, then A is proved to be indefinite.

The idea of our algorithms is as follows: First, we compute an approximate Cholesky factor R_1 of A such that $A + \delta_1 I \approx R_1^T R_1$ in working precision. Next, we compute an approximate inverse of $X_1 \approx R_1^{-1}$ in working precision. Then $X_1^T A X_1$ is computed in *doubled* working precision with its error bound and rounded into an interval matrix $\langle G_2, E_2\rangle$. Iteratively we compute an approximate Cholesky factor R_2 of $G_2 + \delta_2 I$ in working precision, and then compute an approximate inverse of $T_2 \approx R_2^{-1}$ in working precision. After that, $X_1 \cdot T_2$ is computed in doubled working precision and stored into a matrix X_2 in working precision, i.e. $\{X_1 \cdot T_2\}_2 = X_2 \in \mathbb{F}^{n\times n}$. Moreover, $X_2^T A X_2$ is computed in *tripled* working precision with its error bound and rounded into an interval matrix $\langle G_3, E_3\rangle$, and so forth. In general, we aim to develop an algorithm satisfying

$$\kappa(X_m^T A X_m) = \max\{\varphi(\alpha^m)\kappa(A), 1\}.$$

By Sylvester's law of inertia, if A is positive definite, then $X_{k-1}^T A X_{k-1}$ are also positive definite for any k. However, G_k may be indefinite for some k due to the rounding errors. Taking the cases of treating interval matrices $\langle G_k, E_k\rangle$ into account, we need to modify the diagonal shift δ.

By Weyl's theorem, it holds that

$$|\lambda_i(X_{k-1}^T A X_{k-1}) - \lambda_i(G_k)| \le \|E_k\|$$

since $X_{k-1}^T A X_{k-1} \in \langle G_k, E_k\rangle$. To ensure the positive definiteness of $G_k + \delta_k I$ with taking care of the rounding errors, we set δ_k as

$$\delta_k = \gamma\mathbf{u} \cdot \mathrm{tr}(G_k) + \|E_k\|.$$

Our algorithm of an accurate inverse Cholesky factorization with partial pivoting is based on the following iterative refinement of an approximate inverse X of the exact Cholesky factor $\widehat{R}$ of A by the multiplicative corrections:

Algorithm 3.1 (Robust inverse Cholesky factorization) *For a symmetric matrix $A \in \mathbb{F}^{n\times n}$ with $\mathrm{diag}(A) \ge 0$ and a specified tolerance $\varepsilon_{\mathrm{tol}} < 1$, the following algorithm calculates an upper triangular matrix $X = \sum_{i=1}^{m} X^{(i)}$, $X^{(i)} \in \mathbb{F}^{n\times n}$ such that $\|X^T A X - I\| \lesssim \varepsilon_{\mathrm{tol}}$ if such X exists.*

1: *Put $X_0 = I$ and $k = 1$. ($\ell := \lceil k/2 \rceil$)*
2: $\langle B_k, E_B\rangle \leftarrow \{A \cdot X_{k-1}\}_k^{\ell+1}$.
3: $\langle C_k, E_C\rangle \leftarrow \{X_{k-1}^T \cdot B_k\}_{\ell+1} + \langle O, |X_{k-1}^T| E_B\rangle$.
4: $\langle G_k, E_k\rangle \leftarrow \frac{1}{2}(\langle C_k, E_C\rangle + \langle C_k^T, E_C^T\rangle)$
5: *If $\|\mathrm{mag}\langle G_k - I, E_k\rangle\| < \varepsilon_{\mathrm{tol}}$, then $X := X_k$ and stop.*
6: *Compute $\delta_k = \gamma\mathbf{u} \cdot \mathrm{tr}(G_k) + \|E_k\|$.*
7: *Compute $S_k = \mathrm{fl}(G_k + \widetilde{\delta_k} I)$ s.t. $S_k \ge G_k + \delta_k I$.*
8: *Check $\mathrm{diag}(S_k) \ge 0$. If not, then stop.*
9: *Cholesky factorization: $S_k \approx R_k^T R_k$.*
10: *If Step 9 failed, then stop.*
11: $T_k \approx R_k^{-1}$.
12: $X_k \leftarrow \{X_{k-1} \cdot T_k\}_{\ell+1}^{\ell}$.
13: *Update $k \leftarrow k + 1$ and return to 2.*

Note that high precision computations (of dot product) are necessary only in Steps 2, 3 and 12. Among them, the results of the high precision computations are stored in high precision in Steps 2 and 12. If the algorithm stops at Step 8 or 10, then A is proved to be indefinite. This is a mathematical statement. Step 4 ensures that G_k is symmetric.

Remark 3.2 *To avoid bad scaling, we can apply diagonal scaling to the input matrix A with suitable powers of* 2*. See [13] for details.*

Remark 3.3 *If Algorithm 3.1 runs to completion, the positive definiteness of A is also ensured.*

4. Numerical results

We present some numerical results showing the behavior of our proposed algorithm (Algorithm 3.1) of an inverse

Table 1: Results for a scaled Hilbert matrix with $n = 21$ and $\kappa(A) \approx 8.16 \cdot 10^{29}$ by the proposed algorithm

k	$\kappa(X_k^T A X_k)$	$(\gamma\mathbf{u})^k \kappa(A)$	$\kappa(X_k)$	$(\gamma\mathbf{u})^{-\frac{k}{2}}$
0	$8.16 \cdot 10^{29}$	$8.16 \cdot 10^{29}$	1	1
1	$2.84 \cdot 10^{15}$	$9.96 \cdot 10^{15}$	$1.70 \cdot 10^{7}$	$9.05 \cdot 10^{6}$
2	$1.08 \cdot 10^{2}$	$1.22 \cdot 10^{2}$	$8.71 \cdot 10^{13}$	$8.19 \cdot 10^{13}$
3	1.00	< 1	$9.03 \cdot 10^{14}$	$> \sqrt{\kappa(A)}$

Table 2: Results for a Rump matrix with $n = 500$ and $\kappa(A) \approx 4.76 \cdot 10^{53}$ by the proposed algorithm

k	$\kappa(X_k^T A X_k)$	$(\gamma\mathbf{u})^k \kappa(A)$	$\kappa(X_k)$	$(\gamma\mathbf{u})^{-\frac{k}{2}}$
0	$4.76 \cdot 10^{53}$	$4.76 \cdot 10^{53}$	1	1
1	$5.97 \cdot 10^{41}$	$5.94 \cdot 10^{41}$	$8.92 \cdot 10^{5}$	$8.95 \cdot 10^{5}$
2	$1.41 \cdot 10^{31}$	$7.42 \cdot 10^{29}$	$1.83 \cdot 10^{11}$	$8.01 \cdot 10^{11}$
3	$3.69 \cdot 10^{20}$	$9.26 \cdot 10^{17}$	$3.59 \cdot 10^{16}$	$7.17 \cdot 10^{17}$
4	$1.00 \cdot 10^{10}$	$1.16 \cdot 10^{6}$	$6.89 \cdot 10^{21}$	$6.41 \cdot 10^{23}$
5	1.28	< 1	$6.10 \cdot 10^{26}$	$> \sqrt{\kappa(A)}$
6	1.00	< 1	$6.90 \cdot 10^{26}$	$> \sqrt{\kappa(A)}$

Cholesky factorization. All computations are done on Matlab 2009a with IEEE 754 double precision arithmetic as working precision ($\mathbf{u} = 2^{-53} \approx 1.1 \cdot 10^{-16}$). As a stopping criterion for Algorithm 3.1, we set $\varepsilon_{\text{tol}} = 10^{-6}$.

First, a scaled Hilbert matrix H_n is treated. Here H_n is an integer (symmetric positive definite) matrix whose elements are exactly representable in double precision floating-point numbers for $n \leq 21$. For $n = 21$, $\kappa(H_{21}) \approx 8.16 \cdot 10^{29}$. We put $A := H_{21}$. The result is displayed in Table 1.

Next, a slightly modified version of Rump matrix [8] is treated, which is based on `randmat(n,cnd)` in INTLAB [10] and symmetric positive definite. We set `n` $= 500$ and `cnd` $= 10^{50}$. Then $A \in \mathbb{F}^{500\times 500}$ with $\kappa(A) \approx 4.76 \cdot 10^{53}$ is generated. The result is displayed in Table 2.

In both test cases, the condition number of the input matrices (H_{21} and A) is dropped by a factor around $\gamma\mathbf{u}$ in each step until $\kappa(X_k^T A X_k) \approx 1$. It turns out that we obtain an adaptive and robust algorithm of an inverse Cholesky factorization and verification of positive definiteness.

References

[1] G. Alefeld, J. Herzberger: Introduction to Interval Computations, Academic Press, New York, 1983.

[2] J. Demmel: On floating point errors in Cholesky, LAPACK Working Note 14 CS-89-87, Department of Computer Science, University of Tennessee, Knoxville, TN, USA, 1989.

[3] N. J. Higham: Accuracy and Stability of Numerical Algorithms, 2nd ed., SIAM, Philadelphia, PA, 2002.

[4] R. E. Moore: Interval Analysis, Prentice-Hall, Englewood Cliffs, N.J., 1966.

[5] A. Neumaier: Interval Methods for Systems of Equations, Encyclopedia of Mathematics and its Applications, Cambridge University Press, 1990.

[6] T. Ogita: Accurate matrix factorization: Inverse LU and inverse QR factorizations, submitted for publication, 2009.

[7] T. Ogita, S. M. Rump, S. Oishi: Accurate sum and dot product, SIAM J. Sci. Comput., 26:6 (2005), 1955–1988.

[8] S. M. Rump: A class of arbitrarily ill-conditioned floating-point matrices, SIAM J. Matrix Anal. Appl., 12:4 (1991), 645–653.

[9] S. M. Rump: Approximate inverses of almost singular matrices still contain useful information, Forschungsschwerpunktes Informations- und Kommunikationstechnik, Technical Report 90.1, Hamburg University of Technology, 1990.

[10] S. M. Rump: INTLAB – INTerval LABoratory, Developments in Reliable Computing (Tibor Csendes ed.), Kluwer Academic Publishers, Dordrecht, 1999, 77–104.

[11] S. M. Rump: Fast and parallel interval arithmetic, BIT, 39:3 (1999), 534–554.

[12] S. M. Rump: Inversion of extremely ill-conditioned matrices in floating-point, Japan J. Indust. Appl. Math., accepted for publication.

[13] S. M. Rump: Verification of positive definiteness, BIT Numerical Mathematics, 46 (2006), 433–452.

[14] S. M. Rump, T. Ogita, S. Oishi: Accurate floating-point summation part I: faithful rounding, SIAM J. Sci. Comput., 31:1 (2008), 189-224.

[15] S. M. Rump, T. Ogita, S. Oishi: Accurate floating-point summation part II: sign, K-fold faithful and rounding to nearest, SIAM J. Sci. Comput., 31:2 (2008), 1269-1302.

[16] J. H. Wilkinson: A priori error analysis of algebraic processes, Proceedings of International Congress in Mathematics, Izdat Mir, Moscow, 1968, 629–639.