Directed modified Cholesky factorizations and convex quadratic relaxations

Ferenc Domes\(^a\),*, Arnold Neumaier\(^a\)

\(^a\)University of Vienna, Faculty of Mathematics, Oskar-Morgenstern-Platz 1, A-1090 Vienna

Abstract

A directed Cholesky factorization of a symmetric interval matrix \(A\) consists of a permuted upper triangular matrix \(R\) such that for all symmetric \(A \in A\), the residual matrix \(A - R^T R\) is positive semidefinite with tiny entries. This must hold with full mathematical rigor, although the computations are done in floating-point arithmetic.

Similarly, a directed modified Cholesky factorization of a symmetric interval matrix \(A\) consists of a nonsingular permuted upper triangular matrix \(R\) and a non-negative diagonal matrix \(D\) such that for all \(A \in A\) the residual matrix \(A + D - R^T R\) is positive semidefinite with tiny entries.

The paper shows how to construct a directed modified Cholesky factorization with the additional property that the entries of \(D\) are tiny, too, if \(A\) is nearly positive definite, and they are zero for numerically positive definite matrices. The construction is based on an improved version of the directed Cholesky factorization Domes & Neumaier [1], which performs better on nearly singular positive definite matrices. The improved method also allows one to select a set of columns which are eliminated before the other columns are processed. If the factorization fails, but the selected part was successfully processed an incomplete factorization is returned, needed for the new modified factorization. For the new factorizations and relaxation methods detailed algorithms are given. Directed rounding or interval computations are used to make sure that the methods are rigorous in spite of the use of floating point arithmetic.

*Corresponding author

Email addresses: Ferenc.Domes@univie.ac.at (Ferenc Domes), Arnold.Neumaier@univie.ac.at (Arnold Neumaier)

Preprint submitted to Linear Algebra and its Applications July 3, 2014
As application, new techniques are given for pruning boxes in the presence of an additional quadratic constraint, a problem relevant for branch and bound methods for global optimization and constraint satisfaction. Using either the improved directed Cholesky or the directed modified Cholesky factorization, a convex quadratic relaxation is created and an improved box enclosing the set of points in the original box satisfying the relaxed constraint. If the quadratic constraint is strictly convex and the box is sufficiently big, the relaxation and the enclosure are optimal up to rounding errors.

Numerical test show the usefulness of the new factorization methods in the context of pruning.

Keywords: directed Cholesky factorization, quadratic constraints, interval analysis, constraint satisfaction problems, bounding ellipsoids, interval hull, rounding error control, verified computing

2000 MSC: 90C20 Quadratic programming, 15A23, Factorization of matrices, 49M27 Decomposition methods

1. Introduction

A directed Cholesky factorization of a symmetric interval matrix $A$ consists of a permuted upper triangular matrix $R$ such that for all symmetric $A \in A$, the residual matrix $A - R^T R$ is positive semidefinite with tiny entries. This must holds with full mathematical rigor, although the computations are done in floating-point arithmetic.

Domes & Neumaier [1] presented two methods for obtaining a directed Cholesky factorization. The method proved to be useful for other researchers (e.g., [4, 6, 7, 8]). In this paper we discuss two new methods: the improved directed Cholesky factorization and the directed modified Cholesky factorization. The development of these methods was motivated by our research in global optimization (Domes & Neumaier [2]) where we had to factorize the reduced Hessian for which theory requires, that a certain submatrix is positive definite, but the remainder of the matrix can be indefinite.

In general the improved directed Cholesky factorization (discussed in Section [3] performs better on nearly singular positive definite matrices as the methods from [1]. The method also allows us to select a set of columns which are eliminated before the other columns are processed. If the factorization fails, but the selected part was successfully processed the method returns an incomplete factorization which can be useful in many applications.
A directed modified Cholesky factorization of a symmetric interval matrix \( A \) consists of a nonsingular matrix \( R \) and a non-negative diagonal matrix \( D \) such that the residual matrix \( A + D - R^T R \) is positive semidefinite and its entries are tiny for all \( A \in \mathbf{A} \). In addition to this if \( A \) is nearly positive definite the entries of \( D \) are expected to be tiny and zero for numerically positive definite matrices. This method works for severely ill-conditioned and even for indefinite matrices. In addition to this like in the improved directed Cholesky factorization a certain set of columns can be selected which are eliminated before the other columns are processed. An algorithm for constructing a directed modified Cholesky factorization is presented in Section 4. It is a correctly rounding interval version of the (approximate) modified Cholesky factorization discussed, e.g., in Schnabel & Eskow [10, 11].

In the second part of this paper we discuss enhancements to the indefinite case of the \texttt{ehull} enclosure technique from Domes & Neumaier [1] for strictly convex quadratic constraints. Two new methods are presented in Section 5. One of them is based on the improved directed Cholesky factorization from Section 3 and the other one is based on the directed modified Cholesky factorization from Section 4. The new methods widen the application scope of the old \texttt{ehull} since they can compute convex quadratic relaxations for general quadratic constraints. They also perform better on problems where the quadratic coefficient matrix is positive definite but nearly singular.

For the new factorizations and relaxation methods detailed algorithms are given. Directed rounding or interval computations are used to make sure that the methods are rigorous in floating point arithmetic. Numerical test of the new directed Cholesky factorization methods are presented in Section 6.

The techniques presented give the possibility to obtain rigorous bounds on variables that are consequences of the constraints, without the need of giving explicit bounds on them. As already the original \texttt{ehull} enclosure, this makes the method a convenient step in branch and bound methods for solving constrained optimization problems (e.g., [3, 6, 7, 9]).

**Acknowledgments** This research was supported by the Austrian Science Fund (FWF) under the contract numbers P23554-N13 and P22239-N13.

2. **Notation**

2.1. **Matrices**

\( \mathbb{R}^{m \times n} \) denotes the vector space of all \( m \times n \) matrices \( A \) with real entries \( A_{ik} \ (i = 1, \ldots, m, \ k = 1, \ldots, n) \), and \( \mathbb{R}^n = \mathbb{R}^{n \times 1} \) denotes the vector space
of all column vectors of length \( n \). For vectors and matrices, the relations \( =, \neq, <, >, \leq, \geq \) and the absolute value \( |A| \) of a matrix \( A \) are interpreted component-wise.

The \( n \)-dimensional identity matrix is denoted by \( I \) and the \( n \)-dimensional zero matrix is denoted by \( 0 \). The transpose of a matrix \( A \) is denoted by \( A^T \), and \( A^{-T} \) is short for \( (A^T)^{-1} \). The \( i \)th row vector of a matrix \( A \) is denoted by \( A_i \), and the \( j \)th column vector by \( A_j \). For an \( n \times n \) matrix \( A \), \( \text{diag}(A)_i = A_{ii} \).

The number of elements of an index set \( N \) is denoted by \( |N| \). The set \( \neg N \) denotes the complement of \( N \). Let \( I \subseteq \{1, \ldots, m\} \) and \( J \subseteq \{1, \ldots, n\} \) be index sets and let \( n_I := |I|, n_J := |J| \). For an \( n \)-dimensional vector \( x, x_J \) denotes the \( n_J \)-dimensional vector built from the components of \( x \) selected by the index set \( J \). For an \( m \times n \) matrix \( A \), the expression \( A_{i:} \) denotes the \( n_I \times n \) matrix built from the rows of \( A \) selected by the index sets \( I \). Similarly, \( A{:}J \) denotes the \( m \times n_J \) matrix built from the columns of \( A \) selected by the index sets \( J \). Instead of using the index sets \( I \) and \( J \) we also write \( A_{i:k,i:l} \) if \( I = \{i, i+1, \ldots, k\} \) and \( J = \{j, j+1, \ldots, l\} \).

2.2. Boxes

A box \( \mathbf{x} = [\underline{x}, \overline{x}] \), i.e., the Cartesian product of the closed real intervals \( x_i := [\underline{x}_i, \overline{x}_i] \), representing a (bounded or unbounded) axiparallel box in \( \mathbb{R}^n \). \( \mathbb{R}^n \) denotes the set of all \( n \)-dimensional boxes. To take care of one-sided bounds on variables, the values \( -\infty \) and \( \infty \) are allowed as lower and upper bounds of a box, respectively. The condition \( x \in \mathbf{x} \) is equivalent to the collection of simple bounds

\[
\underline{x}_i \leq x_i \leq \overline{x}_i \quad (i = 1, \ldots, n),
\]

or, with inequalities on vectors and matrices interpreted component-wise, to the two-sided vector inequality \( \underline{x} \leq x \leq \overline{x} \). Apart from two-sided constraints, this includes with \( x_i = [a, a] \) variables \( x_i \) fixed at a particular value \( x_i = a \), with \( x_i = [a, \infty] \) lower bounds \( x_i \geq a \), with \( x_i = [-\infty, a] \) upper bounds \( x_i \leq a \), and with \( x_i = [-\infty, \infty] \) free variables. For the notation in interval analysis we mostly follow \[5\].

2.3. Uncertain vectors and matrices

To rigorously account for inaccuracies in computed entries of a matrix, we use interval matrices, standing for uncertain real matrices whose coefficients
are between given lower and upper bounds. Note that all boxes may be considered as interval vectors, i.e., column vectors ($n \times 1$ matrices) with uncertain components, whose values are known only to lie within given intervals. The midpoint, width and the radius of an interval matrix $A$ are the noninterval matrices defined by

$$\text{mid}(A) := (\overline{A} + \underline{A})/2, \quad \text{wid}(A) := \overline{A} - \underline{A}, \quad \text{rad}(A) := \text{wid}(A)/2,$$

respectively. An interval, interval vector, or interval matrix is called thin or degenerate if its width is zero, and thick if its width is positive. A real matrix $A$ is identified with the thin interval matrix with $A = \overline{A} = \underline{A}$.

The expression $A := [\underline{A}, \overline{A}] \in \mathbb{IR}^{m \times n}$ denotes an $m \times n$ interval matrix with lower bound $\underline{A}$ and upper bound $\overline{A}$. $A \in \mathbb{IR}^{n \times n}$ is symmetric if $A_{ik} = A_{ki}$ for all $i, k \in \{1, \ldots, n\}$. The comparison matrix $\langle A \rangle$ of a square interval matrix $A$ is defined by

$$\langle A \rangle_{ij} := \begin{cases} -|A_{ij}| & \text{for } i \neq j, \\ \langle A_{ij} \rangle & \text{for } i = j. \end{cases}$$

3. Improved directed Cholesky factorization

Our algorithm for a directed Cholesky factorization of a symmetric interval matrix $A \in \mathbb{IR}^{n \times n}$ constructs an upper triangular matrix $R' \in \mathbb{IR}^{n \times n}$ and a permutation matrix $P$ such that the residual matrix $\Gamma := A - P^T R' R' P$ is positive semidefinite for all $A \in \mathcal{A}$, and all $\Delta^0_{ij}$ are tiny. If we combine the matrices $P$ and $R$ to $R := R' P$ (which is in general no longer upper triangular), we obtain $\Gamma = A - R^T R$.

If the directed Cholesky factorization fails since $A$ is not (numerically) positive definite, we want the resulting partial factorization to satisfy at least some of the properties of the full factorization. We achieve this by appropriately modifying Algorithm \texttt{DirCholP} by DOMES & NEUMAIER [1], Algorithm 5.5, which either computes a directed Cholesky factor $R$ and a permutation matrix $P$ such that the residual matrix $E$ is positive semidefinite and is very small with respect to $A$, or terminates with an error message and returns an incomplete factorization.

Theorem 5.6 of DOMES & NEUMAIER [1] (which details the properties of the factorization resulting from the old algorithm) still holds for the resulting Algorithm [1] with trivial modifications. In case the factorization failed after
Algorithm 1: Improved directed Cholesky factorization (IDirChol)

**Input:** A symmetric interval matrix $A \in \mathbb{IR}^{n \times n}$ and the index set $M$ (which can be empty) with $M \subseteq \{1, \ldots, n\}$ and $|M| = m$.

**Success:** The upper triangular matrix $R \in \mathbb{R}^{n \times n}$ and a permutation matrix $P \in \mathbb{R}^{n \times n}$ such that $PAP^T - R^TR$ is positive semidefinite for all symmetric $A \in A$.

**Incomplete:** In case $M \neq \emptyset$, the complete factorization failed but the first $m$ steps were successful, return the matrix $A^m \in \mathbb{IR}^{n-m \times n-m}$, $P$ and $R^m \in \mathbb{R}^{m \times m}$.

1. If $A_{ii} < 0$ for any $i \in M$ then return signaling failure;
2. Put $A_1 = A$, $N = M$, $A^m = \emptyset$, $R = 0_n$, $R^m = \emptyset$, $P = I_n$ and change to upward rounding mode;
3. for $k = 1, \ldots, n$ do
4. Find the pivot element
   
   $$\alpha = \max(\text{diag}(\hat{A})),$$

   $$\hat{A} := \begin{cases} A_{NN} & \text{if } N \neq \emptyset, \\ A_k \in \mathbb{R}^{n-k+1} & \text{otherwise.} \end{cases}$$

   Let $j$ denote the index of the pivot element in $A_k$; exchange row $j$ with the first row and column $j$ with the first column of $A_k$.

   Exchange the same rows and columns in the matrix $P$;
5. If the pivot was selected from $N$ remove its index from $N$;
6. Partition the permuted interval matrix $A_k$ as:
   $$A_k = \begin{pmatrix} \alpha_k & \mathbf{a}^T \\ \mathbf{a}_k & B_k \end{pmatrix}$$

   if $\alpha_k \leq 0$ then return $A^m$, $P$, $R^m$ and an error message; else
7. Choose $0 < \gamma_k < 1$, $\rho_k = \gamma_k \sqrt{\alpha_k}$ and $r_k = (\alpha_k + a_k)/(2\rho_k)$;
8. Set $R_{kk} = \rho_k$, $R_{k,k:n} = r_k^T$ and compute $\delta_k := -(\overline{\alpha}_k + \rho_k^2)$,
9. $d_k := \max(\overline{\alpha}_k + \rho_k(-r_k), \rho_k r_k - \overline{\alpha}_k)$;
10. if the residual pivot $\delta_k \leq 0$ then
11. return $A^m$, $P$, $R^m$ and an error message;
12. else Set $A_{k+1} := [B_k - r_k r_k^T - d_k d_k^T / \delta_k, \overline{B}_k + (-r_k) r_k^T + d_k d_k^T / \delta_k]$;
13. end
14. if $M \neq \emptyset$ and $k = m$ then put $A^m = A_k$ and $R^m := R_{MM}$;
15. end
16. return The matrix $R$ and the permutation matrix $P$.
$k$ steps, the incomplete factorization computed by the new algorithm still has the property that $\Gamma^k := (PAP^T)_{KK} - R_{KK}^T R_{KK}$ is positive semidefinite for $K := \{1, \ldots, k\}$. This will be particularly useful later in case both $M \neq \emptyset$, $R_m \neq \emptyset$, since we need the fact that $\Gamma_m := (PAP^T)_{MM} - R_m^T R_m$ is positive semidefinite.

The method given in Domes & Neumaier DomNeuGCf for selecting the parameter $\gamma_k$ in line 8 of Algorithm 1 proved to have some limitations when a submatrix is nearly singular. We therefore present an improved method for selecting the parameter $\gamma_k$ in Algorithm 1.

The improvement is based on the expectation (checked in numerical experiments to be typically valid for a suitable tolerance, e.g., $\kappa = 10^{-6}$) that each $\Gamma^k$ is very small with respect to $A$. Therefore we choose $\rho_k, r_k$ and $\gamma_k$ in Algorithm 1 as follows:

- To make $\Gamma$ positive semidefinite, we have to ensure that $\varepsilon > 0$. Therefore we need $\delta_k > 0$, which is the case when $|\rho_k| < \sqrt{\alpha_k}$. If we also want $\delta_k$ to be very small and assume that $\alpha_k > 0$ (which is true if $A$ is positive definite), we can set $\rho_k = \gamma_k \sqrt{\alpha_k}$ with $\gamma_k < 1$. If in addition to this we choose $\gamma_k \approx 1$, the condition $\delta_k \approx 0$ is also satisfied.

- The entries of $d_k = a_k - \rho_k r_k$ can be made to vanish by setting $r_k := a_k / \rho_k$. Even when $r_k$ and $\rho_k$ are computed inaccurately, we can get a very small $d_k$ by setting $r_k = \tilde{a}_k / (2 \rho_k)$ where $\tilde{a}_k := \bar{a}_k + \bar{a}_k$.

- To make $d_k^T d_k / \delta_k$ very small, we also have to guarantee that $d_k^T d_k \ll \delta_k$. Due to rounding errors, $d_k^T d_k$ is of order

$$d_k := |\bar{a}_k - \tilde{a}_k| + \varepsilon|\tilde{a}_k|,$$

where $\varepsilon$ is the machine precision, so we want $1 \gg \delta_k = \alpha_k - \gamma_k^2 \alpha_k \gg \tilde{a}_k$.

In order to achieve these goals in each step we need choose a suitable $\gamma_k$ such that the diagonal elements of $A_k$ are likely to remain positive. Writing $\mu_k := \gamma_k^{-2}$, we must ensure that the diagonal of

$$A_k = B_k - r_k r_k^T - d_k d_k^T / \delta_k \approx B_k - \frac{\mu_k}{4 \alpha_k} \left( \tilde{a}_k \tilde{a}_k^T + \tilde{d}_k d_k^T \right)$$
is not so small. If $\tilde{a}_k^T \tilde{a}_k = 0$ then $\tilde{a}_k = \tilde{d}_k = 0$ and we may choose $\gamma_k = 1$. Otherwise we note that the trace is maximal for

$$\mu_k = 1 + \sqrt{\frac{\tilde{d}_k^T \tilde{d}_k}{\tilde{a}_k^T \tilde{a}_k}}.$$ 

Thus we might choose $\gamma_k = 1/\sqrt{\mu_k}$, but in order to avoid that $\gamma_k$ gets small, we safeguard it by

$$\gamma_k := \begin{cases} 
1/\min(2, \sqrt{\mu_k}) & \text{if } \tilde{a}_k^T \tilde{a}_k \neq 0, \\
1 & \text{otherwise}.
\end{cases}$$

Using these choices in Algorithm 1 makes the residual matrix not only positive semidefinite but also very small with respect to $A$ for all $A \in A$.

4. Directed modified Cholesky factorization

We now use the directed Cholesky factorization discussed in Section 3 to define a modified directed Cholesky factorization that also works for indefinite matrices.

The directed modified Cholesky factorization of a symmetric interval matrix $A \in \mathbb{R}^{n \times n}$, consisting of a nonsingular matrix $R \in \mathbb{R}^{n \times n}$ and a nonnegative diagonal matrix $D \in \mathbb{R}^{n \times n}$ such that the residual matrix

$$\Gamma := A + D - R^T R$$

is positive semidefinite for all $A \in A$. We compute this factorization by trying to form the directed Cholesky factorization of $A + D$ for different diagonal matrices $D \geq 0$ until we succeed, starting with $D = 0$ and using the partial results of a failed directed Cholesky factorization to select an improved $D$.

In certain applications (see, e.g., Domes & Neumaier [2]) it is useful to have a preferred index set $M$ where theory predicts that, in all nondegenerate cases, $A_{MM}$ is positive definite. In this case, we want to ensure if possible that

$$D_{ii} = 0 \quad \text{for all } i \in M.$$ 

Therefore we put $m := |M|$, and write $A^m \in \mathbb{R}^{(n-m) \times (n-m)}$ for the interval matrix to be factored after the $m$th pivoting step. If $k$ pivot steps were
**Algorithm 2: Modified directed Cholesky factorization (ModDirChol)**

**Input:** A symmetric interval matrix $A \in \mathbb{IR}^{n \times n}$, the index set $M \subseteq \{1, \ldots, n\}$ indicating that $A_{MM}$ is required to be positive definite and the corresponding positive definiteness violation tolerance $\zeta \ll 1$ (e.g., $\zeta = 10^{-6}$).

**Output:** A nonsingular matrix $R \in \mathbb{IR}^{n \times n}$ and a diagonal matrix $D \in \mathbb{IR}^{n \times n}$ with $D \geq 0$ and $D_{MM} = 0$ such that $\Gamma$ defined by (2) is positive semidefinite for all $A \in A$.

1. Use the improved directed Cholesky factorization (Algorithm 1) with $A$ and $M$ to obtain either $\hat{R}$ and $P$ or $A^m$;
2. if Algorithm 1 was successful then return $R := \hat{R}P$ and $D = 0$;
3. else
   4. Put $m = |M|$ and denote the number of successfully performed pivot steps by $k$;
   5. Find $A'$ and $J$ as given by (3) and (4);
   6. Compute the minimum and the maximum eigenvalues $\lambda, \bar{\lambda}$ of the matrix $A'$ and compute $\gamma := 1 + |\lambda| + |\bar{\lambda}|$;
   7. for $\epsilon = 10^{-12}, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}, 1$ do
      8. if $\epsilon > \zeta$ and $k < m$ then
         9. the claim that $A_{MM}$ is positive definite is significantly violated therefore return signaling failure;
      else
         10. Compute $\sigma = \epsilon \gamma + \max(-\lambda, 0)$ and put $D := \sigma J \in \mathbb{IR}^{n \times n}$;
         11. Use the directed Cholesky factorization Algorithm 1 with $A + D$ and $M$ to obtain either $\hat{R}$ and $P$;
         12. if Algorithm 1 was successful then return $R := \hat{R}P$ and $D$;
      end
   end
15. end
16. return signaling failure

Successfully performed in a failed directed Cholesky factorization, we define the matrix

$$A' = \begin{cases} A & \text{if } k < m, \\ A^m & \text{otherwise,} \end{cases} \quad (3)$$
and the diagonal matrix

\[ J \in \mathbb{R}^{n \times n}, \quad J_{ij} := \begin{cases} 1 & \text{if } i = j \text{ and } (k < m \text{ or } i \notin M), \\ 0 & \text{otherwise}. \end{cases} \] (4)

The next \( D \) is then chosen as a multiple \( \sigma J \) of \( J \), trying increasing values of \( \sigma \) until we succeed.

Algorithm 2 gives a precise description of our modified directed Cholesky factorization algorithm.

5. Ellipsoid relaxations

DOMES & NEUMAIER [1, Section 2] describe a method for computing nearly optimal, rigorous enclosure of a strictly convex quadratic constraint. The method makes use of a directed Cholesky factorization. In this section we present two improved versions of the original method: one of them is based on the improved directed Cholesky factorization from Section 3 and the other one is based on the directed modified Cholesky factorization from Section 4. The new methods widen the application scope of the old one. They also perform better for problems where the quadratic coefficient matrix is positive definite but nearly singular.

For a symmetric interval matrix \( A \in \mathbb{IR}^{n \times n} \), an interval vector \( a \in \mathbb{IR}^{n} \), a constant \( \alpha \) and a box \( x \in \mathbb{IR}^{n} \) we define the uncertain quadratic constraint

\[ x^T A x + 2 a^T x \leq \alpha, \quad x \in x, \quad A \in A, \quad a \in a. \] (5)

For this constraint we compute a strictly convex relaxation (which is nearly optimal in case \( A \) is positive definite) as well as a box \( x' \subseteq x \) such that each \( x \) satisfying (5) is contained in the box \( x' \) (hence the method is rigorous). We first assume that \( A \) does not contain zero rows. This means that each variables occurs non-linearly in (5); the case where some variables enter the constraint only linearly will be discussed separately in Subsection 5.4.

For later use we also define the index sets of the bounded and unbounded variables by

\[ N := \{ i \mid x_i \text{ is bounded} \}, \quad M := \neg N, \] (6)

and denote the subspace of \( \mathbb{R}^{n} \) defined by the directions \( x_i, i \in M \) by \( \mathbb{R}^{M} \).
5.1. Enclosing the solution set of norm constraints

In this subsection we summarize the main result of [1, Section 1] applied to the norm constraint
\[ \|Rx\|_2^2 + 2a^T x \leq \hat{\alpha}. \] (7)

Let \( C \) be the inverse of the matrix \( R \), \( d \in \mathbb{R}^n \) with \( d_i = \inf(\sqrt{(CC^T)_{ii}}) \), \( h = \langle CR \rangle d \), \( \beta = \max\{h_i/d_i \mid i = 1 \ldots n\} \approx 1 \), \( \bar{z} = C^T a \) and \( \bar{x} = -C\bar{z} \). Denote the enclosure of the expression
\[ \|\bar{z} + R\bar{x}\|_2 + \beta^{-1} d^T |a - R^T \bar{z}| \] (8)
by \([\gamma, \gamma]\), and denote the enclosure of the expression
\[ \gamma^2 + \hat{\alpha} - 2a^T \bar{x} - \|R\bar{x}\|_2^2, \] (9)
by \([\Delta, \Delta]\). If \( \Delta \geq 0 \) then by [1, Theorem 1.5 and Corollary 1.6], (7) implies that
\[ \|R(x - \bar{x})\|_2 \leq \delta, \] (10)
must be satisfied with
\[ [\delta, \delta] := [\gamma + \sqrt{\Delta}, \gamma + \sqrt{\Delta}]. \] (11)

Therefore the ellipsoid defined by (10) is an enclosure of (7). By the same theorem we also obtain the box
\[ \hat{x} := \left[ (\delta/\beta)d - \bar{x}, (\delta/\beta)d + \bar{x} \right], \] (12)
enclosing the solution set of (7).

In floating point arithmetic we compute \( \bar{z} \approx R^{-T} a \) and \( \bar{x} \approx -R^{-1}\bar{z} \) by floating point calculations, and the remaining variables optimally, by computing the corresponding expressions with directed rounding or interval arithmetic.

5.2. Ellipsoid relaxation by improved Cholesky factorization

The first method is very similar to the one described in DOMES & NEU-MAIER [1, Section 2], but instead of using the directed Cholesky factorization on \( A \) we use the improved directed Cholesky factorization (Algorithm 1) on \( A \) and the index set \( M \). In the original method, no enclosure was computed if the factorization failed, the constraint was not strictly enough convex. Now if
the improved directed Cholesky factorization fails but \( A_{MM} \) was successfully factored, the results of the incomplete factorization, namely the permutation matrix \( P \) and the matrix \( R^m \). By (1) the residual matrix \( \Gamma^m \) of the incomplete factorization is positive definite, satisfy

\[
\| R^m x_M \|_2^2 + 2a_M^T x_M \leq \alpha', \quad x_M \in x_M, \quad A_{MM} \in A_{MM}, \quad a_M \in a_M, \tag{13}
\]

where

\[
\hat{\alpha} := \alpha - \inf(x_N^T A_{NN} x_N + 2a_N^T x_N). \tag{14}
\]

Since by the definition (6) of \( N \) the bound \( \hat{\alpha} \) is finite, Subsection 5.1 can be applied to the \( m \) dimensional norm constraint (13) to obtain an ellipsoid relaxation (10) of (5) in the subspace \( \mathbb{R}^M \). In addition to this since the box \( \hat{x} \) from (12) encloses the solution set of (13) and (13) is a relaxation of (5) in the subspace \( \mathbb{R}^M \) the box

\[
x' \quad x'_M := \hat{x}, \quad x'_N := x_N, \tag{15}
\]

encloses the solution set of (5). The complete and rigorous method is given in Algorithm 3.

5.3. Relaxation by directed modified Cholesky factorization

The second method uses the directed modified Cholesky factorization applied to \( A, M \) and \( \zeta = 0 \). If the factorization is successful, a nonsingular matrix \( R \) and a diagonal matrix \( D \geq 0 \) is obtained such that

\[
A \leq R^T R - D + \Gamma, \quad \forall A \in A \quad \text{and} \quad D_{MM} = 0, \tag{16}
\]

and the residual matrix \( \Gamma \) is positive semidefinite and very small with respect to \( A - D \) (details in Section 4). Substituting (16) into (5) results in

\[
x^T A x + 2a^T x \leq x^T (R^T R - D + \Gamma) x + 2a^T x = \| R x \|^2_2 - x^T Dx + x^T \Gamma x + 2a^T x \leq \alpha,
\]

and using that \( D \geq 0 \) and the residual matrix \( \Gamma \) is positive semidefinite, we end up in

\[
\| R x \|^2_2 + 2a^T x \leq \alpha + x^T Dx - x^T \Gamma x \leq \alpha + x^T Dx \leq \hat{\alpha}, \tag{17}
\]

with

\[
\hat{\alpha} := \alpha + \sup_{x \in \mathbb{x}} x^T Dx = \alpha + \sup_{i \in \sim M} \sum_{i \in \sim M} D_{ii} x_i^2. \tag{18}
\]
Algorithm 3: Ellipsoid relaxation using improved directed Cholesky factorization (ERelIDChol)

**Input:** The constraint \( x^T Ax + 2a^T x \leq \alpha \), \( x \in \mathbb{x} \) with \( A \in \mathbb{A} \) and \( a \in \mathbb{a} \).

**Output:** An ellipsoid relaxation and the rigorous box enclosure \( x \in \mathbb{x}' \).

1. Compute \( M \) by (6) and use the improved directed Cholesky factorization (Algorithm 1) on \( A, M \);
2. if the factorization failed but \( R^m \neq \emptyset \) then
3. We have obtained \( P \) and \( R^m \); put \( R \leftarrow R^m P_M \), \( A \leftarrow A_{MM} \) and compute \( \hat{\alpha} \leftarrow \alpha' \) by (14), using interval arithmetic;
4. else if the factorization was successful then
5. We have obtained \( P \) and \( R \); put \( R \leftarrow RP \) and \( \hat{\alpha} \leftarrow \alpha \);
6. else return signaling failure;
7. Compute the approximative inverse \( C \) of the matrix \( R \);
8. Compute \( d \) with \( d_i = \inf(\sqrt{(CC^T)_{ii}}) \) by using directed rounding;
9. Use upward rounding to compute the values \( h = (CR)d \) and \( \beta = \max\{h_i/d_i \mid i = 1 \ldots n\} \approx 1 \);
10. Set \( \bar{z} = C^T a \) and \( \bar{x} = -C \bar{z} \) and compute an enclosure \([\gamma, \tau]\) for (8), an enclosure \([\Delta, \overline{\Delta}]\) for (9) using interval arithmetic;
11. if \( \Delta < 0 \) then return signaling failure;
12. else
13. Compute the interval \([\delta, \overline{\delta}]\) from (10) by using outward rounding;
14. return The ellipsoid relaxation, given by the norm constraint (11) and the rigorous box enclosure (15);
15. end

This proves that ellipsoid defined by (5) is fully contained in the ellipsoid given by the norm constraint (7). Note that if \( D = 0 \), \( A \) is positive definite, \( \hat{\alpha} = \alpha \) does not depend of the box \( \mathbb{x} \) and since \( \Gamma \) is very small with respect to \( A \), the relative approximation error

\[
\delta(x) := \frac{x^T \Gamma x}{\|Rx\|^2_2},
\]

is also small. Therefore if (5) is strictly convex, (18) is a nearly optimal approximation. On the other hand if \( D \neq 0 \), by (6) the bound (18) has to
be finite; so \((7)\) is a nontrivial inequality.

**Algorithm 4:** Ellipsoid relaxation using directed modified Cholesky factorization (ERelMDChol)

| **Input:** The constraint \(x^TAx + 2a^Tx \leq \alpha, \ x \in x\) with \(A \in A\) and \(a \in a\). |
| **Output:** An ellipsoid relaxation and the rigorous box enclosure \(x \in x'\). |

1. Compute \(M\) by (6) and use the directed modified Cholesky factorization (Algorithm 2) on \(A, M\) and \(\zeta = 0\) to obtain \(D\) and \(R\);  
2. if the factorization failed then return signaling failure;  
3. else  
   4. Compute \(\hat{\alpha}\) by (18), using interval arithmetic;  
   5. Compute the approximative inverse \(C\) of the matrix \(R\);  
   6. Compute \(d\) with \(d_i = \inf(\sqrt{(CC^T)_{ii}})\) by using directed rounding;  
   7. Use upward rounding to compute the values \(h = (CR)d\) and \(\beta = \max\{h_i/d_i \mid i = 1 \ldots n\} \approx 1\);  
   8. Set \(\bar{z} = C^Ta\) and \(\bar{x} = -C\bar{z}\) and compute an enclosure \([\gamma, \gamma]\) for (8), an enclosure \([\Delta, \Delta]\) for (9) using interval arithmetic;  
   9. if \(\Delta < 0\) then return signaling failure;  
10. else  
   11. Compute the interval \([\delta, \delta]\) from (10) by using outward rounding;  
   12. return The ellipsoid relaxation, given by the norm constraint (11) and the rigorous box enclosure (19)  
13. end  
14. end

If we apply Subsection 5.1 to the norm constraint (17) we obtain the ellipsoid relaxation (10) of (5) as well as the box

\[
x' := x \cap \bar{x}. \quad (19)
\]

enclosing the solution set of (5). In floating-point arithmetic the computations must be rigorous, as presented in Algorithm 4.
5.4. Removing purely linear terms

If some variables occur only linearly in (5) the corresponding rows and columns of $A$ have only zero entries, therefore $A$ is singular, and depending on the method computing the relaxation becomes impossible or at least inefficient. Therefore we define the index sets

$J := \{ j \mid A_j k = 0, \forall k = 1, \ldots, n \}, \quad K := \neg J,$

and eliminate the variables $x_J$ from (5) obtaining

$x_T^T A_{KK} x_K + 2 a_T^T x_K \leq \alpha', \quad x_K \in x_K, \quad A_{KK} \in A_{KK}, \quad a_K \in a_K,$

(20)

where

$\alpha' := \alpha - \inf(2 a_J^T x_J).$ \hspace{1cm} (21)

If $\alpha'$ is finite we can apply the methods discussed in the previous subsections without modifications to the lower dimensional system (20) instead of (5) to obtain an ellipsoid relaxation of (5) in the subspace spanned by $x_K$ as well as the box enclosure

$x \in x', \quad x_K' := x_K \cap \left[ (\delta/\beta)d - \bar{x}, (\delta/\beta)d + \bar{x} \right], \quad x_J' := x_J.$ \hspace{1cm} (22)

5.5. Diagonal test for indefiniteness

Since the new methods require that at least $A_{MM}$ is positive definite, it is useful to perform a simple diagonal test for positive definiteness and immediately signaling failure if $A_{ii} < 0$ for any $i \in M$. This saves computation time and should be done every time before a factorization of $A$ is computed.

6. Testing the directed Cholesky factorizations

We compared the new improved directed Cholesky factorization and the modified directed Cholesky factorization methods with the old directed Cholesky factorization method on random real interval matrices of different dimension (column \texttt{dim} in the tables below) and width (column \texttt{width} in the tables below). These matrices can be constructed to be positive definite or indefinite and are always nearly singular, with a very small inverse condition number (column \texttt{icond} in the tables below). For the inverse condition number we take the median of the quotients of the absolute value of the smallest
(numerical) eigenvalues and the absolute value of the largest ones of all \(k\) test matrices \(A^{(i)}\), formally:

\[
\text{icond} := \max_i \left( \frac{\lambda_{\min}(A^{(i)})}{\lambda_{\max}(A^{(i)})} \right), \quad i \in \{1, \ldots, k\}.
\]

The following algorithm shows how the test matrices are created:

<table>
<thead>
<tr>
<th>Algorithm 5: Nearly singular interval matrix generator</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> Given is the dimension (n), a tiny singularity factor (\eta \neq 0) with (</td>
</tr>
<tr>
<td><strong>Output:</strong> Nearly singular positive definite (if (\eta &gt; 0)) or indefinite (if (\eta &lt; 0)) interval matrix (A \in \mathbb{IR}^{n \times n}), (A_{ij} \in [0, 1]) of relative width (\omega)</td>
</tr>
<tr>
<td>1. Generate a random matrix (B \in \mathbb{R}^{n-1 \times n}) with (B_{ij} \in [-1, 1]) for all (i = 1, \ldots, n-1) and (j = 1, \ldots, n);</td>
</tr>
<tr>
<td>2. Compute (C = B^T B \in \mathbb{R}^{n \times n}) and (d = \max(C_{ii}));</td>
</tr>
<tr>
<td>3. if (d = 0) then start again with step 1;</td>
</tr>
<tr>
<td>4. else</td>
</tr>
<tr>
<td>5. Generate a random vector (u \in \mathbb{R}^n) with (u_j \in [-1, 1]) for all (j = 1, \ldots, n);</td>
</tr>
<tr>
<td>6. Divide (u) by (\max(</td>
</tr>
<tr>
<td>7. Set (\underline{A} = C/d + \eta u u^T) and (\overline{A} = \underline{A} + \omega</td>
</tr>
<tr>
<td>8. return the interval matrix (A := [\underline{A}, \overline{A}]);</td>
</tr>
<tr>
<td>9. end</td>
</tr>
</tbody>
</table>

We first compare the approximate Cholesky factorization (computed with LAPACK, row \(\text{Chol}\) in the tables below), the directed Cholesky factorization with diagonal pivoting (computed by DOMES & NEUMAIER [II] Algorithm 5.5, row \(\text{DirChol}\) in the tables below), the improved directed Cholesky factorization (computed by Algorithm \([1]\) row \(\text{IDirChol}\) in the tables below) and the modified directed Cholesky factorization (computed by Algorithm \([2]\) row \(\text{MDirChol}\) in the tables below) on 200 real positive definite matrices with dimensional 20 and a very small inverse condition number.
<table>
<thead>
<tr>
<th>method</th>
<th>dim</th>
<th>width</th>
<th>iters</th>
<th>icond</th>
<th>diagpert</th>
<th>solved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chol</td>
<td>20</td>
<td>0</td>
<td>200</td>
<td>(1.36 \cdot 10^{-13})</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>DirChol</td>
<td>20</td>
<td>0</td>
<td>200</td>
<td>(1.36 \cdot 10^{-13})</td>
<td>0</td>
<td>2%</td>
</tr>
<tr>
<td>IDirChol</td>
<td>20</td>
<td>0</td>
<td>200</td>
<td>(1.36 \cdot 10^{-13})</td>
<td>0</td>
<td>86%</td>
</tr>
<tr>
<td>MDirChol</td>
<td>20</td>
<td>0</td>
<td>200</td>
<td>(1.36 \cdot 10^{-13})</td>
<td>5.09 (\cdot 10^{-13})</td>
<td>100%</td>
</tr>
</tbody>
</table>

The next table shows how the efficiency of improved directed Cholesky factorization scales with the problem dimension.

<table>
<thead>
<tr>
<th>method</th>
<th>dim</th>
<th>width</th>
<th>iters</th>
<th>icond</th>
<th>diagpert</th>
<th>solved</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDirChol</td>
<td>10</td>
<td>0</td>
<td>200</td>
<td>(1.95 \cdot 10^{-13})</td>
<td>0</td>
<td>97%</td>
</tr>
<tr>
<td>IDirChol</td>
<td>40</td>
<td>0</td>
<td>200</td>
<td>(1.04 \cdot 10^{-13})</td>
<td>0</td>
<td>53%</td>
</tr>
<tr>
<td>IDirChol</td>
<td>100</td>
<td>0</td>
<td>200</td>
<td>(9.94 \cdot 10^{-14})</td>
<td>0</td>
<td>4%</td>
</tr>
</tbody>
</table>

The next table shows how the efficiency of improved directed Cholesky factorization scales with the problem dimension in case the interval entries of the matrices are not thin.

<table>
<thead>
<tr>
<th>method</th>
<th>dim</th>
<th>width</th>
<th>iters</th>
<th>icond</th>
<th>diagpert</th>
<th>solved</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDirChol</td>
<td>10</td>
<td>(10^{-14})</td>
<td>200</td>
<td>(2.03 \cdot 10^{-13})</td>
<td>0</td>
<td>89%</td>
</tr>
<tr>
<td>IDirChol</td>
<td>40</td>
<td>(10^{-14})</td>
<td>200</td>
<td>(1.25 \cdot 10^{-13})</td>
<td>0</td>
<td>28%</td>
</tr>
<tr>
<td>IDirChol</td>
<td>100</td>
<td>(10^{-14})</td>
<td>200</td>
<td>(1.09 \cdot 10^{-13})</td>
<td>0</td>
<td>2%</td>
</tr>
</tbody>
</table>

The next table shows how the efficiency of directed modified Cholesky factorization scales with the problem dimension in case the interval entries of the matrices are not thin.

<table>
<thead>
<tr>
<th>method</th>
<th>dim</th>
<th>width</th>
<th>iters</th>
<th>icond</th>
<th>diagpert</th>
<th>solved</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDirChol</td>
<td>10</td>
<td>0</td>
<td>200</td>
<td>(1.99 \cdot 10^{-13})</td>
<td>(1.58 \cdot 10^{-13})</td>
<td>100%</td>
</tr>
<tr>
<td>MDirChol</td>
<td>40</td>
<td>0</td>
<td>200</td>
<td>(1.26 \cdot 10^{-13})</td>
<td>(1.75 \cdot 10^{-12})</td>
<td>100%</td>
</tr>
<tr>
<td>MDirChol</td>
<td>100</td>
<td>0</td>
<td>200</td>
<td>(1.1 \cdot 10^{-13})</td>
<td>(4.11 \cdot 10^{-10})</td>
<td>100%</td>
</tr>
</tbody>
</table>

The next table shows how the efficiency of directed modified Cholesky factorization scales with the problem dimension in case the interval entries of the matrices are not thin.
<table>
<thead>
<tr>
<th>method</th>
<th>dim</th>
<th>width</th>
<th>iters</th>
<th>icond</th>
<th>diagpert</th>
<th>solved</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDirChol</td>
<td>10</td>
<td>$10^{-14}$</td>
<td>200</td>
<td>$2.01 \cdot 10^{-13}$</td>
<td>$2.34 \cdot 10^{-13}$</td>
<td>100%</td>
</tr>
<tr>
<td>MDirChol</td>
<td>40</td>
<td>$10^{-14}$</td>
<td>200</td>
<td>$1.24 \cdot 10^{-13}$</td>
<td>$2.76 \cdot 10^{-12}$</td>
<td>100%</td>
</tr>
<tr>
<td>MDirChol</td>
<td>100</td>
<td>$10^{-14}$</td>
<td>200</td>
<td>$9.48 \cdot 10^{-14}$</td>
<td>$4.11 \cdot 10^{-10}$</td>
<td>100%</td>
</tr>
</tbody>
</table>

**Discussion.** The tests show that the new methods both improve the quality of the old one in case we want to factor ill-conditioned matrices. In particular the column diagpert shows the average diagonal perturbation

$$\text{med}(\max_{i,k}(D_{kk}^{(i)}))$$

(which can be nonzero only for MDirChol) and the percentage of successfully factored matrices.


