SIAM J. MATRIX ANAL. APPL.
Vol. 19, No. 4, pp. 1097–1110, October 1998

016

A MODIFIED CHOLESKY ALGORITHM BASED ON A SYMMETRIC INDEFINITE FACTORIZATION*

SHEUNG HUN CHENG† AND NICHOLAS J. HIGHAM†

Abstract. Given a symmetric and not necessarily positive definite matrix A, a modified Cholesky algorithm computes a Cholesky factorization $P(A+E)P^T = R^TR$, where P is a permutation matrix and E is a perturbation chosen to make $A+E$ positive definite. The aims include producing a small-normed E and making $A+E$ reasonably well conditioned. Modified Cholesky factorizations are widely used in optimization. We propose a new modified Cholesky algorithm based on a symmetric indefinite factorization computed using a new pivoting strategy of Ashcraft, Grimes, and Lewis. We analyze the effectiveness of the algorithm, both in theory and practice, showing that the algorithm is competitive with the existing algorithms of Gill, Murray, and Wright and Schnabel and Eskow. Attractive features of the new algorithm include easy-to-interpret inequalities that explain the extent to which it satisfies its design goals, and the fact that it can be implemented in terms of existing software.

Key words. modified Cholesky factorization, optimization, Newton's method, symmetric indefinite factorization

AMS subject classification. 65F05

PII. S0895479896302898

1. Introduction.

Modified Cholesky factorization is a widely used technique in optimization; it is used for dealing with indefinite Hessians in Newton methods [11], [21] and for computing positive definite preconditioners [6], [20]. Given a symmetric matrix A, a modified Cholesky algorithm produces a symmetric perturbation E such that $A+E$ is positive definite, along with a Cholesky (or $\mathrm{LDL^T}$) factorization of $A+E$. The objectives of a modified Cholesky algorithm can be stated as follows [21].

O1. If A is "sufficiently positive definite" then E should be zero.

O2. If A is indefinite, $\|E\|$ should not be much larger than

$$\min\{\,\|\Delta A\| : A+\Delta A \text{ is positive definite}\,\}$$

for some appropriate norm.

O3. The matrix $A+E$ should be reasonably well conditioned.

O4. The cost of the algorithm should be the same as the cost of standard Cholesky factorization to highest order terms.

Two existing modified Cholesky algorithms are one by Gill, Murray, and Wright [11, section 4.4.2.2], which is a refinement of an earlier algorithm of Gill and Murray [10], and an algorithm by Schnabel and Eskow [21].

The purpose of this work is to propose an alternative modified Cholesky algorithm that has some advantages over the existing algorithms. In outline, our approach is to compute a symmetric indefinite factorization

$$PAP^T = LDL^T, \tag{1.1}$$

*Received by the editors April 26, 1996; accepted for publication (in revised form) by P. Gill June 4, 1997; published electronically July 17, 1998. The research of the second author was supported by Engineering and Physical Sciences Research Council grant GR/H/94528.

http://www.siam.org/journals/simax/19-4/30289.html

†Department of Mathematics, University of Manchester, Manchester, M13 9PL, England (chengsh@ma.man.ac.uk, na.nhigham@na-net.ornl.gov).

where P is a permutation matrix, L is unit lower triangular, and D is block diagonal with diagonal blocks of dimension 1 or 2, and to provide the factorization

$$P(A+E)P^T = L(D+F)L^T, \tag{1.2}$$

where F is chosen so that $D+F$ (and hence also $A+E$) is positive definite.[1] This approach is not new; it was suggested by Moré and Sorensen [19] for use with factorizations (1.1) computed with the Bunch–Kaufman [3] and Bunch–Parlett [4] pivoting strategies. However, for neither of these pivoting strategies are all the conditions (O1)–(O4) satisfied, as is recognized in [19]. The Bunch–Parlett pivoting strategy requires $O(n^3)$ comparisons for an $n \times n$ matrix, so condition (O4) does not hold. For the Bunch–Kaufman strategy, which requires only $O(n^2)$ comparisons, it is difficult to satisfy conditions (O1)–(O3), as we explain in section 3.

We use a new pivoting strategy for the symmetric indefinite factorization devised by Ashcraft, Grimes, and Lewis [2], for which conditions (O1)–(O3) are satisfied to within factors depending only on n and for which the cost of the pivot searches is *usually* negligible. We describe this so-called bounded Bunch–Kaufman (BBK) pivoting strategy and its properties in the next section.

There are two reasons why our algorithm might be preferred to those of Gill, Murray, and Wright and of Schnabel and Eskow (henceforth denoted the GMW algorithm and the SE algorithm, respectively). The first is a pragmatic one: we can make use of any available implementation of the symmetric indefinite factorization with the BBK pivoting strategy, needing to add just a small amount of post-processing code to form the modified Cholesky factorization. In particular, we can use the efficient implementations for both dense and sparse matrices written by Ashcraft, Grimes, and Lewis [2], which make extensive use of levels 2 and 3 BLAS for efficiency on high-performance machines. In contrast, in coding the GMW and SE algorithms one must either begin from scratch or make nontrivial changes to an existing Cholesky factorization code.

The second attraction of our approach is that we have a priori bounds that explain the extent to which conditions (O1)–(O3) are satisfied—essentially, if L is well conditioned then an excellent modified Cholesky factorization is guaranteed. For the GMW and SE algorithms it is difficult to describe under what circumstances the algorithms can be guaranteed to perform well.

2. Pivoting strategies. We are interested in symmetric indefinite factorizations (1.1) computed in the following way. If the symmetric matrix $A \in \mathbb{R}^{n\times n}$ is nonzero, we can find a permutation Π and an integer $s = 1$ or 2 so that

$$\Pi A \Pi^T = \begin{array}{c} \\ s \\ n-s \end{array} \begin{array}{c} \begin{array}{cc} s & n-s \end{array} \\ \begin{bmatrix} E & C^T \\ C & B \end{bmatrix} \end{array},$$

with E nonsingular. Having chosen such a Π we can factorize

$$\Pi A \Pi^T = \begin{bmatrix} I_s & 0 \\ CE^{-1} & I_{n-s} \end{bmatrix} \begin{bmatrix} E & 0 \\ 0 & B - CE^{-1}C^T \end{bmatrix} \begin{bmatrix} I_s & E^{-1}C^T \\ 0 & I_{n-s} \end{bmatrix}. \tag{2.1}$$

[1] Strictly, (1.2) is not a Cholesky factorization, since we allow $D+F$ to have 2×2 diagonal blocks, but since any such blocks are positive definite it seems reasonable to use the term "modified Cholesky factorization."

This process is repeated recursively on the $(n-s) \times (n-s)$ Schur complement

$$S = B - CE^{-1}C^T,$$

yielding the factorization (1.1) on completion. This factorization costs $n^3/3$ operations (the same cost as Cholesky factorization of a positive definite matrix) plus the cost of determining the permutations Π.

The Bunch–Parlett pivoting strategy [4] searches the whole submatrix S at each stage, requiring a total of $O(n^3)$ comparisons, and it yields a matrix L whose maximum element is bounded by 2.781. The Bunch–Kaufman pivoting strategy [3], which is used with the symmetric indefinite factorization in both LAPACK [1] and LINPACK [7], searches at most two columns of S at each stage, so it requires only $O(n^2)$ comparisons in total. The Bunch–Kaufman pivoting strategy yields a backward stable factorization [16], but $\|L\|_\infty$ is unbounded, even relative to $\|A\|_\infty$, which makes this pivoting strategy unsuitable for use in a modified Cholesky algorithm, for reasons explained in section 3.

To describe the BBK pivoting strategy [2] it suffices to describe the pivot choice for the first stage of the factorization.

ALGORITHM BBK (BBK pivoting strategy). *This algorithm determines the pivot for the first stage of the symmetric indefinite factorization applied to a symmetric matrix* $A \in \mathbb{R}^{n\times n}$.

$\alpha := (1+\sqrt{17})/8$ (≈ 0.64)
$\gamma_1 :=$ maximum magnitude of any subdiagonal entry in column 1.
If $\gamma_1 = 0$ there is nothing to do on this stage of the factorization.
if $|a_{11}| \geq \alpha\gamma_1$
    use a_{11} as a 1×1 pivot ($s = 1$, $\Pi = I$).
else
    $i := 1$; $\gamma_i := \gamma_1$
    repeat
        $r :=$ row index of first (subdiagonal) entry of maximum magnitude in column i.
        $\gamma_r :=$ maximum magnitude of any off-diagonal entry in column r.
        if $|a_{rr}| \geq \alpha\gamma_r$
            use a_{rr} as a 1×1 pivot ($s = 1$, Π swaps rows and columns 1 and r).
        else if $\gamma_i = \gamma_r$
            use $\begin{bmatrix} a_{ii} & a_{ri} \\ a_{ri} & a_{rr} \end{bmatrix}$ as a 2×2 pivot ($s = 2$, Π swaps rows and columns 1 and i, and 2 and r).
        else
            $i := r$, $\gamma_i := \gamma_r$.
        end
    until a pivot is chosen
end

The repeat loop in Algorithm BBK searches for an off-diagonal element a_{ri} that is simultaneously the largest in magnitude in the rth row and the ith column, and it uses this element to build a 2×2 pivot; the search terminates prematurely if a suitable 1×1 pivot is found.

The following properties noted in [2] are readily verified, using the property that

any 2×2 pivot satisfies

$$\left| \begin{bmatrix} a_{ii} & a_{ri} \\ a_{ri} & a_{rr} \end{bmatrix}^{-1} \right| \leq \frac{1}{\gamma_r(1-\alpha^2)} \begin{bmatrix} \alpha & 1 \\ 1 & \alpha \end{bmatrix}.$$

1. Every entry of L is bounded by $\max\{1/(1-\alpha), 1/\alpha\} \approx 2.78$.
2. Every 2×2 pivot block D_{ii} satisfies $\kappa_2(D_{ii}) \leq (1+\alpha)/(1-\alpha) \approx 4.56$.
3. The growth factor for the factorization, defined in the same way as for Gaussian elimination, is bounded in the same way as for the Bunch–Kaufman pivoting strategy, namely, by $(1+\alpha^{-1})^{n-1} \approx (2.57)^{n-1}$.

Since the value of γ_i increases strictly from one pivot step to the next, the search in Algorithm BBK takes at most n steps. The cost of the searching is intermediate between the cost for the Bunch–Kaufman strategy and that for the Bunch–Parlett strategy. Matrices are known for which the entire remaining submatrix must be searched at each step, in which case the cost is the same as for the Bunch–Parlett strategy. However, Ashcraft, Grimes, and Lewis [2] found in their numerical experiments that on average less than $2.5k$ comparisons were required to find a pivot from a $k \times k$ submatrix, and they give a probabilistic analysis which shows that the expected number of comparisons is less than $ek \approx 2.718k$ for matrices with independently distributed random elements. Therefore we regard the symmetric indefinite factorization with the BBK pivoting strategy as being of similar cost to Cholesky factorization, while recognizing that in certain rare cases the searching overhead may increase the operation count by about 50%.

The symmetric indefinite factorization with the BBK pivoting strategy is backward stable; the same rounding error analysis as for the Bunch–Kaufman pivoting strategy is applicable [2], [16].

The modified Cholesky algorithm of the next section and the corresponding analysis are not tied exclusively to the BBK pivoting strategy. We could use instead the "fast Bunch–Parlett" pivoting strategy from [2], which appears to be more efficient than the BBK strategy when both are implemented in block form [2]. We mention in passing that a block implementation of the SE algorithm has been developed by Daydé [5]. Alternatively, we could use one of the pivoting strategies from [8], [9].

3. The modified Cholesky algorithm. We begin by defining the distance from a symmetric matrix $A \in \mathbb{R}^{n\times n}$ to the symmetric matrices with minimum eigenvalue $\lambda_{\min}$ at least δ, where $\delta \geq 0$:

$$\mu(A,\delta) = \min\{\, \|\Delta A\| : \lambda_{\min}(A+\Delta A) \geq \delta \,\}. \tag{3.1}$$

The distances in the 2- and Frobenius norms, and perturbations that achieve them, are easily evaluated (cf. [12, Thms. 2.1, 3.1]).

THEOREM 3.1. *Let the symmetric matrix $A \in \mathbb{R}^{n\times n}$ have the spectral decomposition $A = Q\Lambda Q^T$ (Q orthogonal, $\Lambda = \operatorname{diag}(\lambda_i)$). Then, for the Frobenius norm,*

$$\mu_F(A,\delta) = \left(\sum_{\lambda_i < \delta} (\delta - \lambda_i)^2 \right)^{1/2}$$

and there is a unique optimal perturbation in (3.1)*, given by*

$$\Delta A = Q \operatorname{diag}(\tau_i) Q^T, \qquad \tau_i = \begin{cases} 0, & \lambda_i \geq \delta, \\ \delta - \lambda_i, & \lambda_i < \delta. \end{cases} \tag{3.2}$$

For the 2*-norm,*

$$\mu_2(A,\delta) = \max(\,0, \delta - \lambda_{\min}(A)\,),$$

and an optimal perturbation is $\Delta A = \mu_2(A,\delta)I$. *The Frobenius norm perturbation* (3.2) *is also optimal in the* 2*-norm.* □

Our modified Cholesky algorithm has a parameter $\delta \geq 0$ and it attempts to produce the perturbation (3.2).

ALGORITHM MC (modified Cholesky factorization). *Given a symmetric matrix* $A \in \mathbb{R}^{n\times n}$ *and a parameter* $\delta \geq 0$ *this algorithm computes a permutation matrix* P, *a unit lower triangular matrix* L, *and a block diagonal matrix* D *with diagonal blocks of dimension* 1 *or* 2 *such that*

$$P(A+E)P^T = LDL^T$$

and $A+E$ *is symmetric positive definite* (*or symmetric positive semidefinite if* $\delta = 0$). *The algorithm attempts to ensure that if* $\lambda_{\min}(A) < \delta$ *then* $\lambda_{\min}(A+E) \approx \delta$.

1. Compute the symmetric indefinite factorization $PAP^T = L\widetilde{D}L^T$ using the BBK pivoting strategy.
2. Let $D = \widetilde{D} + \Delta\widetilde{D}$, where $\Delta\widetilde{D}$ is the minimum Frobenius norm perturbation that achieves $\lambda_{\min}(\widetilde{D} + \Delta\widetilde{D}) \geq \delta$ (thus $\Delta\widetilde{D} = \mathrm{diag}(\Delta\widetilde{D}_{ii})$, where $\Delta\widetilde{D}_{ii}$ is the minimum Frobenius norm perturbation that achieves $\lambda_{\min}(\widetilde{D}_{ii} + \Delta\widetilde{D}_{ii}) \geq \delta$).

To what extent does Algorithm MC achieve the objectives (O1)–(O4) listed in section 1? Objective (O4) is clearly satisfied, provided that the pivoting strategy does not require a large amount of searching, since the cost of step 2 is negligible. For objectives (O1)–(O3) to be satisfied we need the eigenvalues of A to be reasonably well approximated by those of $\widetilde{D}$. For the Bunch–Kaufman pivoting strategy the elements of L are unbounded and the eigenvalues of $\widetilde{D}$ can differ greatly from those of A (subject to A and $\widetilde{D}$ having the same inertia), as is easily shown by example. This is the essential reason why the Bunch–Kaufman pivoting strategy is unsuitable for use in a modified Cholesky algorithm.

To investigate objectives (O1)–(O3) we will make use of a theorem of Ostrowski [18, p. 224]. Here, the eigenvalues of a symmetric $n\times n$ matrix are ordered $\lambda_n \leq \cdots \leq \lambda_1$.

THEOREM 3.2 (Ostrowski). *Let* $M \in \mathbb{R}^{n\times n}$ *be symmetric and* $S \in \mathbb{R}^{n\times n}$ *nonsingular. Then for each* $k = 1\colon n$

$$\lambda_k(SMS^T) = \theta_k\lambda_k(M),$$

where $\lambda_n(SS^T) \leq \theta_k \leq \lambda_1(SS^T)$. □

Assuming first that $\lambda_{\min}(A) > 0$ and applying the theorem with $M = \widetilde{D}$ and $S = L$, we obtain

$$\lambda_{\min}(A) \leq \lambda_{\max}(LL^T)\lambda_{\min}(\widetilde{D}).$$

Now E will be zero if $\lambda_{\min}(\widetilde{D}) \geq \delta$, which is certainly true if

$$\lambda_{\min}(A) \geq \delta\,\lambda_{\max}(LL^T). \tag{3.3}$$

Next, we assume that $\lambda_{\min}(A)$ is negative and apply Theorems 3.1 and 3.2 to obtain

$$\lambda_{\max}(\Delta\widetilde{D}) = \delta - \lambda_{\min}(\widetilde{D}) \leq \delta - \frac{\lambda_{\min}(A)}{\lambda_{\min}(LL^T)}. \tag{3.4}$$

Using Theorem 3.2 again, with (3.4), yields

$$\begin{aligned}\|E\|_2 = \lambda_{\max}(E) &= \lambda_{\max}(L\Delta\widetilde{D}L^T)\\ &\le \lambda_{\max}(LL^T)\lambda_{\max}(\Delta\widetilde{D})\\ &\le \lambda_{\max}(LL^T)\left(\delta - \frac{\lambda_{\min}(A)}{\lambda_{\min}(LL^T)}\right) \qquad (\lambda_{\min}(A) < 0).\end{aligned} \tag{3.5}$$

A final invocation of Theorem 3.2 gives

$$\lambda_{\min}(A+E) \ge \lambda_{\min}(LL^T)\lambda_{\min}(\widetilde{D}+\Delta\widetilde{D}) \ge \lambda_{\min}(LL^T)\delta$$

and

$$\begin{aligned}\|A+E\|_2 = \lambda_{\max}(A+E) &= \lambda_{\max}(L(\widetilde{D}+\Delta\widetilde{D})L^T)\\ &\le \lambda_{\max}(LL^T)\lambda_{\max}(\widetilde{D}+\Delta\widetilde{D})\\ &= \lambda_{\max}(LL^T)\max(\delta, \lambda_{\max}(\widetilde{D}))\\ &\le \lambda_{\max}(LL^T)\max\left(\delta, \frac{\lambda_{\max}(A)}{\lambda_{\min}(LL^T)}\right).\end{aligned}$$

Hence

$$\kappa_2(A+E) \le \kappa_2(LL^T)\max\left(1, \frac{\lambda_{\max}(A)}{\lambda_{\min}(LL^T)\,\delta}\right). \tag{3.6}$$

We can now assess how well objectives (O1)–(O3) are satisfied. To satisfy objective (O1) we would like E to be zero when $\lambda_{\min}(A) \ge \delta$, and to satisfy (O2) we would like $\|E\|_2$ to be not much larger than $\delta - \lambda_{\min}(A)$ when A is not positive definite. The sufficient condition (3.3) for E to be zero and inequality (3.5) show that these conditions do hold modulo factors $\lambda_{\max,\min}(LL^T)$. Inequality (3.6) bounds $\kappa_2(A+E)$ with the expected reciprocal dependence on δ, again with terms $\lambda_{\max,\min}(LL^T)$. The conclusion is that the modified Cholesky algorithm is guaranteed to perform well if $\lambda_{\min}(LL^T)$ and $\lambda_{\max}(LL^T)$ are not too far from 1.

Note that, since L is unit lower triangular, $e_1^T(LL^T)e_1 = 1$, which implies that $\lambda_{\min}(LL^T) \le 1$ and $\lambda_{\max}(LL^T) \ge 1$. For the BBK pivoting strategy we have $\max_{i,j}|l_{ij}| \le 2.781$, so

$$1 \le \lambda_{\max}(LL^T) \le \operatorname{trace}(LL^T) = \|L\|_F^2 \le n + \tfrac{1}{2}n(n-1)2.781^2 \le 4n^2 - 3n. \tag{3.7}$$

Furthermore,

$$1 \le \lambda_{\min}(LL^T)^{-1} = \|(LL^T)^{-1}\|_2 = \|L^{-1}\|_2^2 \le (3.781)^{2n-2}, \tag{3.8}$$

using a bound from [15, Thm. 8.13 and Prob. 8.5]. These upper bounds are approximately attainable, but in practice are rarely approached. In particular, the upper bound of (3.8) can be approached only in the unlikely event that most of the subdiagonal elements of L are negative and of near maximal magnitude. Note that each 2×2 pivot causes a subdiagonal element $l_{i+1,i}$ to be zero and so further reduces the likelihood of $\|L^{-1}\|_2$ being large.

In the analysis above we have exploited the fact that the extent to which the eigenvalues of A and $\widetilde{D}$ agree can be bounded in terms of the condition of L. If L is well conditioned then the singular values of A are close to the moduli of the eigenvalues of $\widetilde{D}$. We are currently exploring the application of this fact to the computation of rank-revealing factorizations.

4. Comparison with the GMW and SE algorithms. The GMW and SE algorithms both carry out the steps of a Cholesky factorization of a symmetric matrix $A \in \mathbb{R}^{n\times n}$, increasing the diagonal entries as necessary in order to ensure that negative pivots are avoided. (Actually, the GMW algorithm works with an $\mathrm{LDL}^{\mathrm{T}}$ factorization, where D is diagonal, but the difference is irrelevant to our discussion.) Hence both algorithms produce Cholesky factors of $P^T(A+E)P$ with a diagonal E. From Theorem 3.1 we note that the "optimal" perturbation in objective (O2) of section 1 is, in general, full for the Frobenius norm and can be taken to be diagonal for the 2-norm (but is generally not unique). There seems to be no particular advantage to making a diagonal perturbation to A. Our algorithm perturbs the whole matrix, in general.

By construction, the GMW and SE algorithms make perturbations E to A that are bounded a priori by functions of n and $\|A\|$ only. The GMW algorithm produces a perturbation E for which

$$\|E\|_\infty \le \left(\frac{\beta}{\xi} + (n-1)\xi\right)^2 + 2(\alpha + (n-1)\xi^2) + \delta, \tag{4.1}$$

where $\delta \ge 0$ is a tolerance,

$$\alpha = \max_i |a_{ii}|, \quad \beta = \max_{i\ne j} |a_{ij}|, \quad \xi^2 = \max\{\,\alpha, \beta/\sqrt{n^2-1}, u\,\},$$

and u is the unit roundoff [11, p. 110]. For the SE algorithm the perturbation is bounded in terms of a certain eigenvalue bound ϕ obtained by applying Gershgorin's theorem:

$$\|E\|_\infty \le \phi + \frac{2\tau}{1-\tau}(\phi + \alpha), \tag{4.2}$$

where τ is a tolerance, suggested in [21] to be chosen as $\tau = u^{1/3}$. The quantity ϕ satisfies $\phi \le n(\alpha+\beta)$, so (4.2) is a smaller bound than (4.1) by about a factor n.

The bounds (4.1) and (4.2) can be compared with (3.5) for Algorithm MC. The bound (3.5) has the advantage of directly comparing the perturbation made by Algorithm MC with the optimal one, as defined by (3.1) and evaluated in Theorem 3.1, and it is potentially a much smaller bound than (4.1) and (4.2) if $|\lambda_{\min}(A)| \ll |\lambda_{\max}(A)|$ and $\kappa_2(LL^T)$ is not too large. On the other hand, the bound (3.5) can be much larger than (4.1) and (4.2) if $\kappa_2(LL^T)$ is large.

All three algorithms satisfy objective (O1) of not modifying a sufficiently positive definite matrix, though for the GMW and SE algorithms no condition analogous to (3.3) that quantifies "sufficiently" in terms of $\lambda_{\min}(A)$ is available. Bounds for $\kappa_2(A+E)$ that are exponential in n hold for the GMW and SE algorithms [21]. The same is true for Algorithm MC: see (3.6)–(3.8).

To summarize, in terms of the objectives of section 1 for a modified Cholesky algorithm, Algorithm MC is theoretically competitive with the GMW and SE algorithms, with the weakness that if $\kappa_2(LL^T)$ is large then the bound on $\|E\|_2$ is weak.

When applied to an indefinite matrix, the GMW and SE algorithms provide information that enables a direction of negative curvature of the matrix to be produced; these directions are required in certain algorithms for unconstrained optimization in order to move away from nonminimizing stationary points. For an indefinite matrix, Algorithm MC provides immediate access to a direction of negative curvature from the

$\mathrm{LDL}^{\mathrm{T}}$ factorization computed in step 1, and because $\kappa(L)$ is bounded, this direction satisfies conditions required for convergence theory [19].

Finally, we consider the behavior of the algorithms in the presence of rounding errors. Algorithm MC is backward stable because the underlying factorization is [2]: barring large element growth in the symmetric indefinite factorization with the BBK pivoting strategy, the algorithm produces $\mathrm{LDL}^{\mathrm{T}}$ factors not of $P(A+E)P^T$, but of $P(A+E+F)P^T$, where $\|F\|_2 \le c_n u \|A+E\|_2$ with c_n a constant. Although no comments on numerical stability are given in [11] and [21], a simple argument shows that the GMW and SE algorithms are backward stable. Apply either algorithm to A, obtaining the Cholesky factorization $P(A+E)P^T = R^TR$. Now apply the same algorithm to $P(A+E)P^T$: it will not need to modify $P(A+E)P^T$, so it will return the same computed R factor. But since no modification was required, the algorithm must have carried out a standard Cholesky factorization. Since Cholesky factorization is a backward stable process, the modified Cholesky algorithm must itself be backward stable.

5. Numerical experiments. We have experimented with MATLAB implementations of Algorithm MC and the GMW and SE algorithms. The M-file for the GMW algorithm was provided by M. Wright and sets the tolerance $\delta = 2u$ (which is the value of MATLAB's variable `eps`). The M-file for the SE algorithm was provided by E. Eskow and sets the tolerance $\tau = (2u)^{1/3}$. In Algorithm MC we set $\delta = \sqrt{u}\|A\|_\infty$.

The aims of the experiments are as follows: to see how well the Frobenius norm of the perturbation E produced by Algorithm MC approximates the distance $\mu_F(A,\delta)$ defined in (3.1), and to compare the norms of the perturbations E and the condition numbers of $A+E$ produced by the three algorithms. We measure the perturbations E by the ratios

$$r_F = \frac{\|E\|_F}{\mu_F(A,\delta)}, \qquad r_2 = \frac{\|E\|_2}{|\lambda_{\min}(A)|},$$

which differ only in their normalization and the choice of norm. Algorithm MC attempts to make r_F close to 1. The quantity r_2 is used by Schnabel and Eskow to compare the performance of the GMW and SE algorithms; since E is diagonal for these algorithms, r_2 compares the amount added to the diagonal with the minimum diagonal perturbation that makes the perturbed matrix positive semidefinite.

First, we note that the experiments of Schnabel and Eskow [21] show that the SE algorithm can produce a substantially smaller value of r_2 than the GMW algorithm. Schnabel and Eskow also identified a 4×4 matrix for which the GMW algorithm significantly outperforms the SE algorithm:

$$A = \begin{bmatrix} 1890.3 & -1705.6 & -315.8 & 3000.3 \\ & 1538.3 & 284.9 & -2706.6 \\ & & 52.5 & -501.2 \\ & & & 4760.8 \end{bmatrix}, \tag{5.1}$$

$$\lambda(A) = \{-0.38, -0.34, -0.25, 8.2 \times 10^3\}.$$

We give results for this matrix in Table 5.1; they show that Algorithm MC can also significantly outperform the SE algorithm.

We ran a set of tests similar to those of Schnabel and Eskow [21]. The matrices A are of the form $A = Q\Lambda Q^T$, where $\Lambda = \mathrm{diag}(\lambda_i)$ with the eigenvalues λ_i from one

TABLE 5.1
Measures of E for 4×4 matrix (5.1).

	MC	GMW	SE
r_F	1.3	2.7	3.7×10^3
r_2	1.7	2.7	2.8×10^3

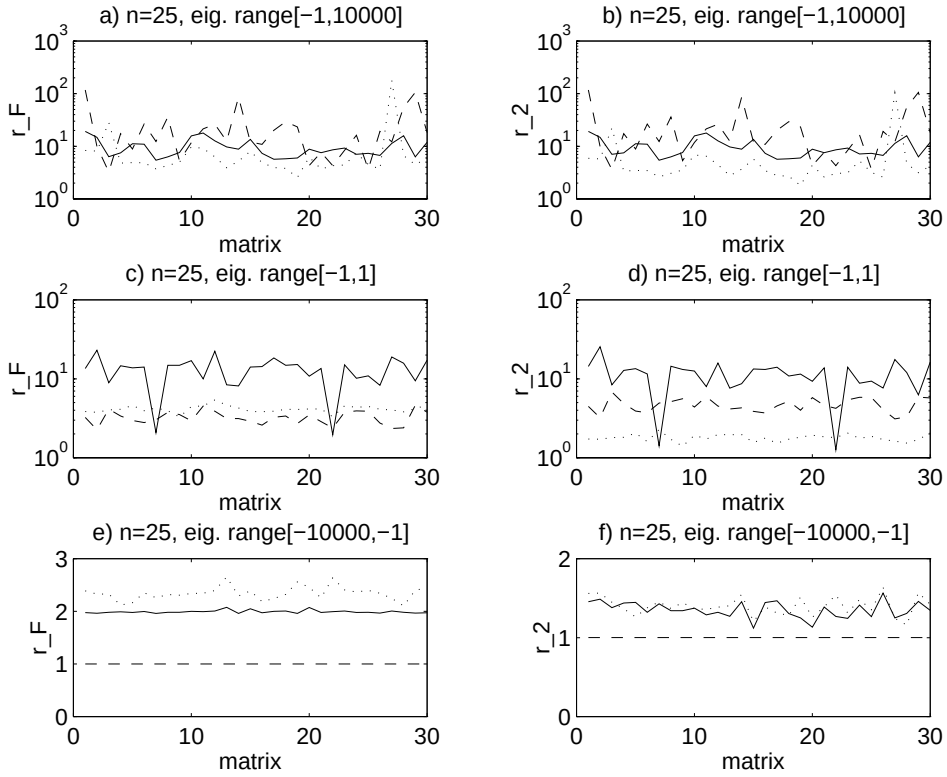

FIG. 5.1. *Measures of E for 30 random indefinite matrices with $n = 25$. Key: GMW —, SE $\cdots$, MC - - -.*

of three random uniform distributions: $[-1, 10^4]$, $[-1, 1]$, and $[-10^4, -1]$. For the first range, one eigenvalue is generated from the range $[-1, 0)$ to ensure that A has at least one negative eigenvalue. The matrix Q is a random orthogonal matrix from the Haar distribution, generated using the routine `qmult` from the Test Matrix Toolbox [14], which implements an algorithm of Stewart [22]. For each eigenvalue distribution we generated 30 different matrices, each corresponding to a fresh sample of Λ and of Q. We took $n = 25, 50, 100$. The ratios r_F and r_2 are plotted in Figures 5.1–5.3. Figure 5.4 plots the condition numbers $\kappa_2(A + E)$ for $n = 25$; the condition numbers for $n = 50$ and $n = 100$ show a very similar behavior. Table 5.2 reports the number of comparisons used by the BBK pivoting strategy on these matrices for each n; the maximum number of comparisons is less than n^2 in each case.

In Figure 5.5 we report results for three nonrandom matrices from the Test Matrix

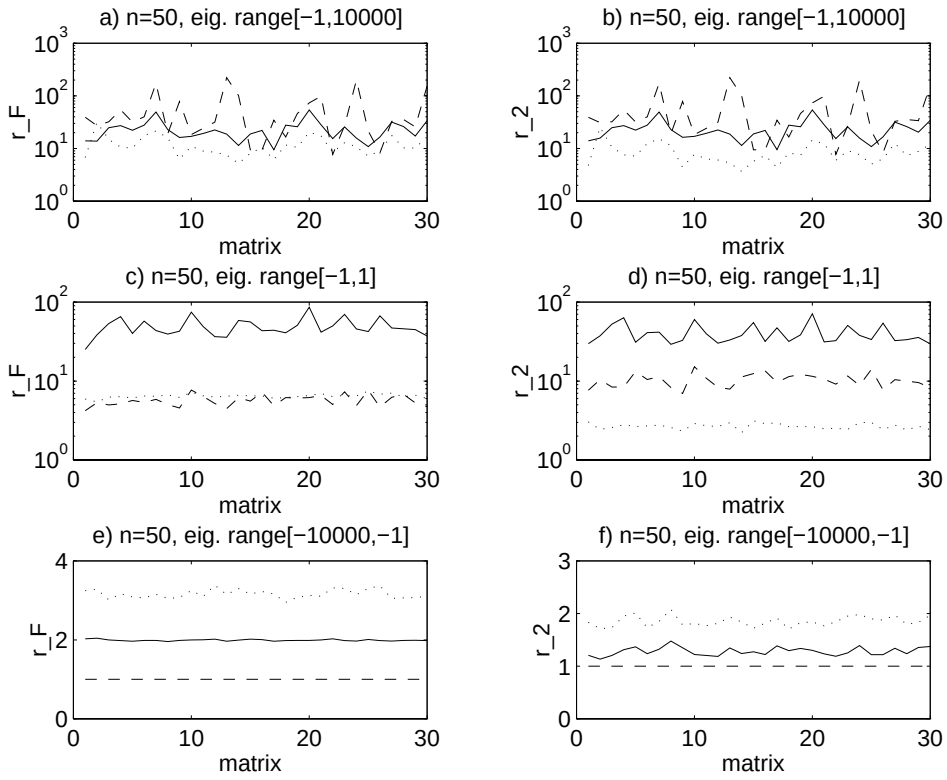

FIG. 5.2. *Measures of E for 30 random indefinite matrices with $n = 50$. Key: GMW —, SE $\cdots$, MC - - -.*

TABLE 5.2
Number of comparisons for BBK pivoting strategy.

n:	25	50	100
max	523	2188	8811
mean	343.9	1432.8	5998.4

Toolbox. `Clement` is a tridiagonal matrix with eigenvalues plus and minus the numbers $n-1, n-3, n-5, \ldots, (1 \text{ or } 0)$. `Dingdong` is the symmetric $n \times n$ Hankel matrix with (i,j) element $0.5/(n-i-j+1.5)$, whose eigenvalues cluster around $\pi/2$ and $-\pi/2$. `Ipjfact` is the Hankel matrix with (i,j) element $1/(i+j)!$.

Our conclusions from the experiments are as follows.

1. None of the three algorithms is uniformly better than the others in terms of producing a small perturbation E, whichever measure r_F or r_2 is used. All three algorithms can produce values of r_F and r_2 significantly greater than 1, depending on the problem.
2. Algorithm MC often achieves its aim of producing $r_F \approx 1$. It produced r_F of order 10^3 for the eigenvalue distribution $[-1, 10^4]$ for each n, and the values of $\kappa_2(LL^T)$ (not shown here) were approximately $100 r_F$ in each such case.

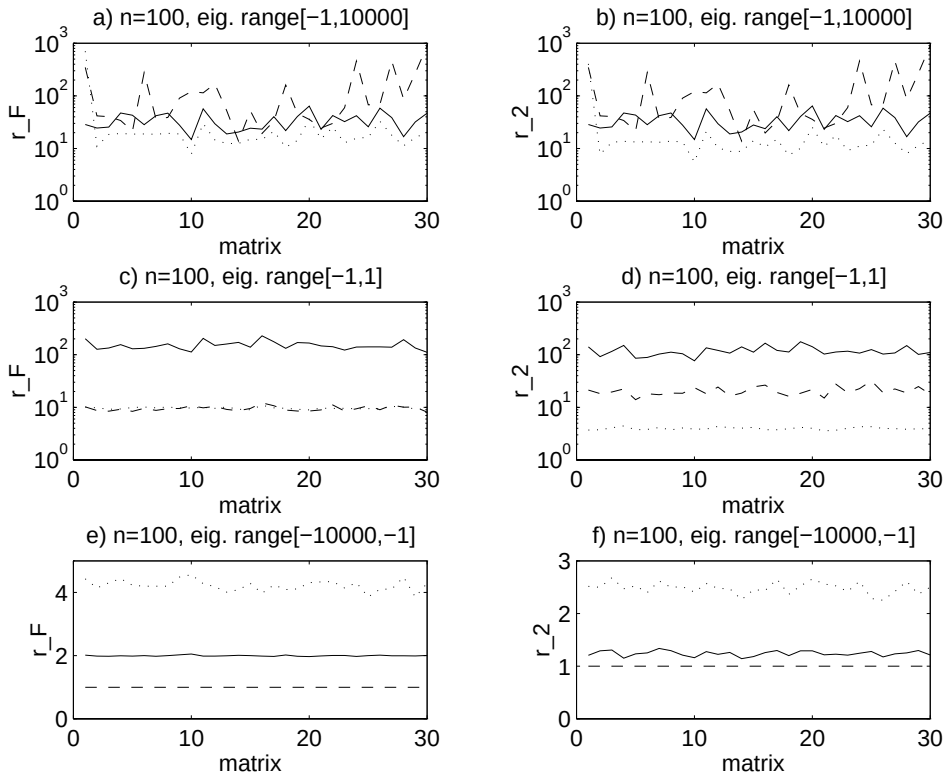

FIG. 5.3. *Measures of E for* 30 *random indefinite matrices with* $n = 100$. *Key: GMW —, SE* $\cdots$, *MC - - -.*

However, often r_F was of order 1 when $\kappa_2(LL^T)$ was of order 10^2 or 10^3, so a large value of $\kappa_2(LL^T)$ is only a necessary condition, not a sufficient one, for poor performance of Algorithm MC; in other words, the bounds of section 3 can be weak.

3. The condition numbers $\kappa_2(A+E)$ vary greatly among the algorithms. Our experience is that for $\delta = \sqrt{u}\|A\|_\infty$ Algorithm MC fairly consistently produces condition numbers of order $100/\sqrt{u}$; the condition number is, as predicted by (3.6), much smaller for the random matrices with eigenvalues on the range $[-10^4, -1]$, because the algorithm attempts to perturb all the eigenvalues to δ. The condition numbers produced by the GMW and SE algorithms vary greatly with the type of matrix.

The fact that r_F is close to 1 for the random matrices with eigenvalues in the range $[-10^4, -1]$ for Algorithm MC is easily explained. Let A be negative definite. Then Algorithm MC computes $P(A + E)P^T = L(\delta I)L^T$. Hence

$$\begin{aligned} r_F &= \frac{\|E\|_F}{(\sum_i (\delta - \lambda_i)^2)^{1/2}} \\ &\le \frac{\|E\|_F}{\|A\|_F} = \frac{\|A - \delta \cdot P^T L L^T P\|_F}{\|A\|_F} \end{aligned}$$

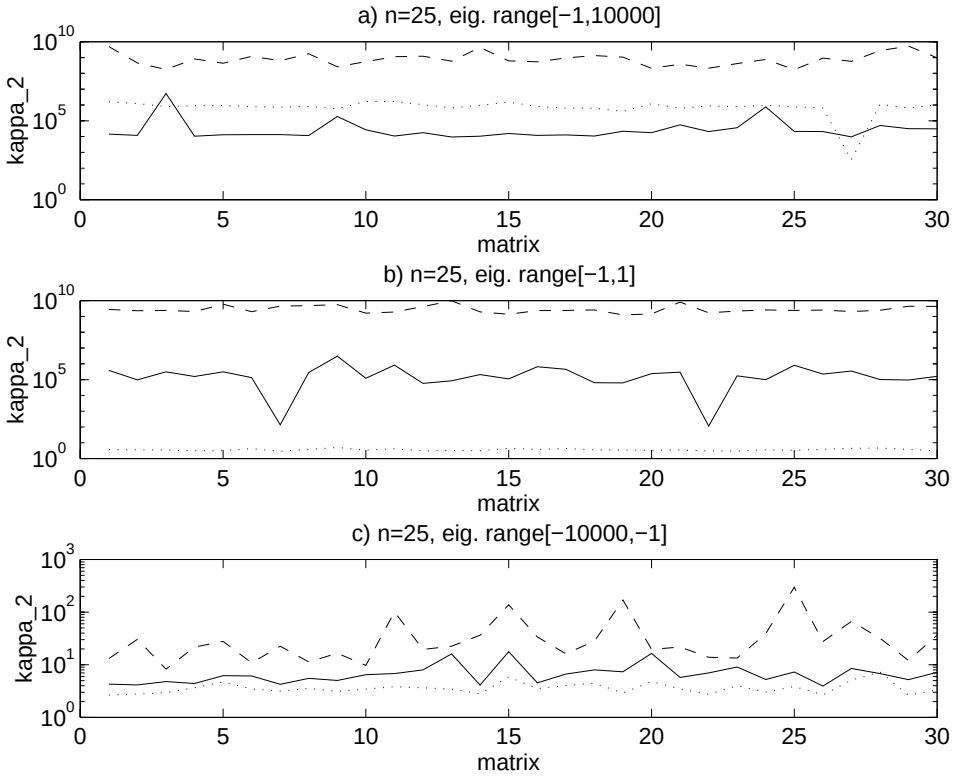

FIG. 5.4. *Condition numbers* $\kappa_2(A+E)$ *for* 30 *random indefinite matrices with* $n = 25$. *Key: GMW —, SE* $\cdots$*, MC - - -.*

$$\leq \frac{\|A\|_F + \delta\|LL^T\|_F}{\|A\|_F}$$
$$\leq 1 + \frac{(4n^2 - 3n)\delta}{\|A\|_F},$$

using (3.7), so r_F can exceed 1 only by a tiny amount for Algorithm MC applied to a negative definite matrix, irrespective of $\kappa_2(LL^T)$.

6. Concluding remarks. Algorithm MC, based on the symmetric indefinite factorization with the bounded Bunch–Kaufman pivoting strategy, merits consideration as an alternative to the algorithms of Gill, Murray, and Wright and Schnabel and Eskow. The results in section 5 suggest that the new algorithm is competitive with the GMW and SE algorithms in terms of the objectives (O1)–(O4) listed in section 1. Algorithm MC has the advantages that the extent to which it satisfies the objectives is neatly, although not sharply, described by the bounds of section 3 and that it can be implemented by augmenting existing software with just a small amount of additional code.

Since all three modified Cholesky algorithms can "fail," that is, they can produce unacceptably large perturbations, it is natural to ask how failure can be detected and what should be done about it. The GMW and SE algorithms produce their (diagonal)

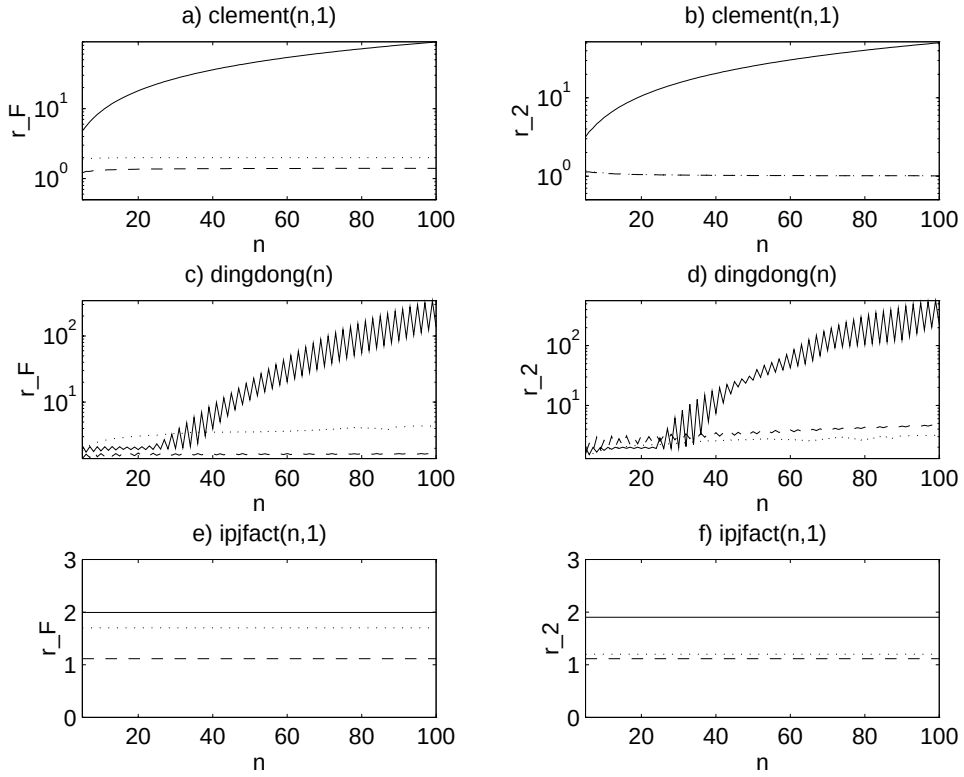

FIG. 5.5. *Measures of E for three nonrandom matrices. Key: GMW —, SE ···, MC - - -.*

perturbations explicitly, so it is trivial to evaluate their norms. For Algorithm MC, the perturbation to A is (see (1.2)) $E = P^T L(D+F)L^T P - A$, which would require $O(n^3)$ operations to form explicitly. However, we can *estimate* $\|E\|_\infty$ using the norm estimator from [13] (which is implemented in LAPACK). The estimator requires the formation of products Ex for certain vectors x, and these can be computed in $O(n^2)$ operations; the estimate produced is a lower bound that is nearly always within a factor 3 of the true norm. For all three algorithms, then, we can inexpensively test whether the perturbation produced is acceptably small. Unfortunately, for none of the algorithms is there an obvious way to improve a modified Cholesky factorization that makes too big a perturbation; whether improvement is possible, preferably cheaply, is an open question. Of course one can always resort to computing an optimal perturbation by computing the eigensystem of A and using the formulae in Theorem 3.1.

We note that we have approached the problem of modified Cholesky factorization from a purely linear algebra perspective. An important test of a modified Cholesky algorithm is to evaluate it in an optimization code on representative problems, as was done by Schlick [20] for the GMW and SE algorithms. This we plan to do for Algorithm MC in future work.

Finally, we mention that a generalization of the modified Cholesky problem motivated by constrained optimization is analyzed in detail in [17].

REFERENCES

[1] E. Anderson, Z. Bai, C. H. Bischof, J. W. Demmel, J. J. Dongarra, J. J. Du Croz, A. Greenbaum, S. J. Hammarling, A. McKenney, S. Ostrouchov, and D. C. Sorensen, *LAPACK Users' Guide, Release* 2.0, 2nd ed., Society for Industrial and Applied Mathematics, Philadelphia, PA, 1995.

[2] C. Ashcraft, R. G. Grimes, and J. G. Lewis, *Accurate symmetric indefinite linear equation solvers*, SIAM J. Matrix Anal. Appl., to appear.

[3] J. R. Bunch and L. Kaufman, *Some stable methods for calculating inertia and solving symmetric linear systems*, Math. Comp., 31 (1977), pp. 163–179.

[4] J. R. Bunch and B. N. Parlett, *Direct methods for solving symmetric indefinite systems of linear equations*, SIAM J. Numer. Anal., 8 (1971), pp. 639–655.

[5] M. J. Daydé, *A Block Version of the Eskow–Schnabel Modified Cholesky Factorization*, Technical report RT/APO/95/8, Dept. Informatique et Maths Appls., ENSEEIHT-IRIT, 31071 Toulouse Cedex, France, 1995.

[6] M. J. Daydé, J.-Y. L'Excellent, and N. I. M. Gould, *On the Use of Element-By-Element Preconditioners to Solve Large Scale Partially Separable Optimization Problems*, Report RAL-95-010, Atlas Centre, Rutherford Appleton Laboratory, Didcot, Oxon, UK, 1995.

[7] J. J. Dongarra, J. R. Bunch, C. B. Moler, and G. W. Stewart, *LINPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1979.

[8] I. S. Duff, N. I. M. Gould, J. K. Reid, J. A. Scott, and K. Turner, *The factorization of sparse symmetric indefinite matrices*, IMA J. Numer. Anal., 11 (1991), pp. 181–204.

[9] I. S. Duff, J. K. Reid, N. Munskgaard, and H. B. Nielsen, *Direct solution of sets of linear equations whose matrix is sparse, symmetric and indefinite*, J. Inst. Math. Appl., 23 (1979), pp. 235–250.

[10] P. E. Gill and W. Murray, *Newton-type methods for unconstrained and linearly constrained optimization*, Math. Programming, 7 (1974), pp. 311–350.

[11] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, London, 1981.

[12] N. J. Higham, *Computing a nearest symmetric positive semidefinite matrix*, Linear Algebra Appl., 103 (1988), pp. 103–118.

[13] N. J. Higham, *FORTRAN codes for estimating the one-norm of a real or complex matrix, with applications to condition estimation (Algorithm* 674*)*, ACM Trans. Math. Software, 14 (1988), pp. 381–396.

[14] N. J. Higham, *The Test Matrix Toolbox for* Matlab *(Version* 3.0*)*, Numerical Analysis report 276, Manchester Centre for Computational Mathematics, Manchester, England, 1995.

[15] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1996.

[16] N. J. Higham, *Stability of the diagonal pivoting method with partial pivoting*, SIAM J. Matrix Anal. Appl., 18 (1997), pp. 52–65.

[17] N. J. Higham and Sheung Hun Cheng, *Modifying the inertia of matrices arising in optimization*, Linear Algebra Appl., to appear.

[18] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, 1985.

[19] J. J. Moré and D. C. Sorensen, *On the use of directions of negative curvature in a modified Newton method*, Math. Programming, 16 (1979), pp. 1–20.

[20] T. Schlick, *Modified Cholesky factorizations for sparse preconditioners*, SIAM J. Sci. Comput., 14 (1993), pp. 424–445.

[21] R. B. Schnabel and E. Eskow, *A new modified Cholesky factorization*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 1136–1158.

[22] G. W. Stewart, *The efficient generation of random orthogonal matrices with an application to condition estimators*, SIAM J. Numer. Anal., 17 (1980), pp. 403–409.