A Revised Modified Cholesky Factorization Algorithm [1]

Robert B. Schnabel
Elizabeth Eskow

University of Colorado at Boulder
Department of Computer Science
Campus Box 430
Boulder, Colorado 80309-0430 USA

[1]Research supported by Air Force Office of Scientific Research Grant F49620-97-1-0164, Army Research Office Contract DAAH04-94-G-0228, and NSF grant CDA-9502956.

Abstract

A modified Cholesky factorization algorithm introduced originally by Gill and Murray and refined by Gill, Murray and Wright, is used extensively in optimization algorithms. Since its introduction in 1990, a different modified Cholesky factorization of Schnabel and Eskow has also gained widespread usage. Compared with the Gill-Murray-Wright algorithm, the Schnabel-Eskow algorithm has a smaller *a priori* bound on the perturbation added to ensure positive definiteness, and some computational advantages, especially for large problems. Users of the Schnabel-Eskow algorithm, however, have reported cases from two different contexts where it makes a far larger modification to the original matrix than is necessary and than is made by the Gill-Murray-Wright method. This paper reports a simple modification to the Schnabel-Eskow algorithm that appears to correct all the known computational difficulties with the method, without harming its theoretical properties, or its computational behavior in any other cases. In new computational tests, the modifications to the original matrix made by the new algorithm appear virtually always to be smaller than those made by the Gill-Murray-Wright algorithm, sometimes by significant amounts. The perturbed matrix is allowed to be more ill-conditioned with the new algorithm, but this seems to be appropriate in the known contexts where the underlying problem is ill-conditioned.

1 Introduction

Modified Cholesky factorizations are widely used in optimization. A numerically stable modified Cholesky factorization algorithm was introduced by Gill and Murray in 1974 [9]. Given a symmetric, not necessarily positive definite matrix $A \in R^{n \times n}$, a modified Cholesky factorization calculates a Cholesky (i.e. LL^T or LDL^T) factorization of a positive definite matrix $A + E$ in a way that attempts to satisfy four goals: 1) If A is safely positive definite, E is 0 ; 2) If A is indefinite, $\|E\|_\infty$ is not much greater than the magnitude of the most negative eigenvalue of A, $\lambda_1(A)$; 3) $A + E$ is reasonably well-conditioned; 4) The cost of the factorization is only a small multiple of n^2 operations more than the $O(n^3)$ cost of the standard Cholesky factorization.

The factorization of Gill and Murray was subsequently refined by Gill, Murray and Wright [10] (hereafter referred to as GMW81). This version has been widely used in optimization methods since its inception. More recently, Schnabel and Eskow [12] (hereafter referred to as SE90) introduced a factorization that is based on different techniques. Both factorizations choose E to be diagonal. Both satisfy properties 1, 3 and 4 mentioned above; they differ in how closely they satisfy property 2. The SE90 factorization has a significantly smaller *a priori* bound on $\|E\|$, where in this paper $\|E\|$ is always the infinity norm, and in computational tests, it appears that $\|E\|$ is smaller for the SE90 factorization than the GWM81 factorization in most cases as well. In practice, both factorizations appear very satisfactory for use in optimization algorithms and both are now widely used.

While the overall computational experience with the SE90 factorization since its introduction appears to have been quite good, a few instances have arisen where its performance is poor. The SE90 paper contained one example where the amount that is added to A, while within the theoretical bounds, is far larger than the magnitude of $\lambda_1(A)$, and also far larger than the amount added by the GWM81 factorization. In the first

years following the publication of the SE90 algorithm, Wolfgang Hartmann of SAS made us aware of another problem with similar behavior. More recently, David Gay, Michael Overton and Margaret Wright encountered a class of problems, arising in primal-dual interior methods for constrained optimization [8], where the SE90 factorization again sometimes added far too much, while the GMW81 factorization performed well.

All the known examples where the SE90 factorization adds too much (i.e. the ratio of $\|E\|$ to the magnitude of the most negative eigenvalue of A is greater than, say, 5) turn out to be matrices A that are the sum of a large positive semi-definite matrix B and a much smaller (in norm) indefinite matrix C. In these cases, one wants $\|E\|$ to be of order $\|C\|$, but instead the SE90 algorithm sometimes produces $\|E\|$ of order $\|B\|$. In the experience of Gay, Overton and Wright, this introduced difficulties in the constrained optimization algorithm using the SE90 factorization that were not experienced when using the GWM81 factorization.

This paper introduces a simple modification to the SE90 modified Cholesky factorization that remedies these difficulties, without harming its computational performance in any other known cases. The modification is to tighten slightly the condition under which the algorithm switches from Phase 1 (standard Cholesky factorization) to Phase 2, thereby making it slightly more likely to stay in Phase 1 at a given iteration of the factorization. The theoretical effect of this change is to increase the upper bound on $\|E\|$ by a factor of at most 1.1. The modification resolves all the problem cases for the SE90 factorization of which we are aware.

Section 2 contains brief background on the modified Cholesky factorization, including the methods of GMW81 and SE90. This section is not intended to be a comprehensive reference; for more background on the modified Cholesky factorization or its use in optimization, see GMW81, SE90, or Dennis and Schnabel [7]. Section 3 motivates the change in the SE90 example that this paper introduces, using the problematic example from SE90. In Section 4 we present the complete new algorithm; several other very minor changes related to the main change and to badly conditioned problems are included. Section 5 briefly presents the theoretical results for the new method. In Section 6 we summarize the results of computational tests of the new algorithm, and the methods of GMW81 and SE90, on the problems of Gay, Overton and Wright, on a problem of Hartmann, and on the random test problems that were used in SE90 to assess the behavior of the factorizations. Fortran code for the revised factorization will be available from the authors.

2 Brief Background on Modified Cholesky Factorizations

The modified Cholesky procedures of GMW and SE90, like the standard Cholesky factorization, can be viewed as recursive procedures. At the beginning of stage j, an $(n-j+1) \times (n-j+1)$ submatrix A_j remains to be factored (with $A_1 = A$). We assume that A_j has the form

$$A_j = \begin{bmatrix} \alpha_j & a_j^T \\ a_j & \hat{A}_j \end{bmatrix} \tag{2.1}$$

where $\alpha_j \in R$ is the current jth diagonal element and is called the pivot, a_j is the current vector of elements in column j below the diagonal, and $\hat{A}_j \in R^{(n-j)\times(n-j)}$. The modified Cholesky factorization chooses a nonnegative amount δ_j to add to α_j, and then calculates $L_{jj} = \sqrt{\alpha_j + \delta_j}$, $L_{ij} = (a_j)_i / L_{jj}$, $i = j+1, \cdots, n$, and

$$A_{j+1} = \hat{A}_j - \frac{a_j a_j^T}{\alpha_j + \delta_j}. \tag{2.2}$$

The challenge in the modified Cholesky factorization is choosing each δ_j. The algorithm must guarantee that each $\delta_j = 0$ if A turns out to be safely positive definite. It also must employ some form of lookahead so that if A is not positive definite, δ_j is chosen to be an appropriate positive quantity beginning at a sufficiently early iteration of the factorization. This is not trivial; for example, waiting to set $\delta_j > 0$ until α_j first becomes negative and then adding amounts $\delta_j > -\alpha_j$ is not satisfactory, as it usually will result in $\|E\|$ much greater than $|\lambda_1|(A)$.

The algorithm of GMW81 chooses each δ_j to be the smallest non-negative number for which

$$0 \leq \frac{\|a_j\|_\infty^2}{\alpha_j + \delta_j} \leq \beta^2 \tag{2.3}$$

(with a minimum of $\delta_j = -2\alpha_j$ if $\alpha_j < 0$), where β is an *a priori* bound selected to minimize a worst-case bound on $\|E\|$ and also to assure that $E = 0$ if A is safely positive definite. The result, with ϵ denoting machine precision, is

$$\beta^2 = \max\{\gamma, \xi/\sqrt{n^2-1}, \epsilon\}, \text{ where } \gamma = \max_i |A_{ii}| \text{ and } \xi = \max_{j<i} |A_{ij}|. \tag{2.4}$$

The requirement that $\beta^2 \geq \gamma$ guarantees $E = 0$ if A is positive definite. The overall *a priori* bound on $\|E\|_{GMW}$ depends on the largest element in brackets in (2.4); the smallest upper bound is

$$n^2\gamma + 2(n-1)\xi. \tag{2.5}$$

which is achieved when $\beta^2 = \xi/\sqrt{n^2-1}$.

The method of SE90 is divided into two phases. The first phase consists of a normal Cholesky factorization in which the factors are overwritten on A. Step j of Phase 1 is allowed to proceed only if α_j is positive and the smallest diagonal of the remaining submatrix at the next step, i.e. at step $j+1$, is "safely" positive, using the following test. Let the vector ζ be defined as

$$\zeta_i = A_{ii} - A_{ij}^2/\alpha_j, \quad i > j. \tag{2.6}$$

SE90 completes step j of the standard Cholesky algorithm only if

$$\min_i \zeta_i \geq \tau\gamma, \quad \text{where } \tau = \epsilon^{\frac{1}{3}}, \tag{2.7}$$

and otherwise switches to Phase 2. Note that the components of ζ would be the diagonal elements of A_{j+1} if step j of the unmodified Cholesky procedure were to be completed;

see (2.2). Satisfaction of (2.7) thus guarantees that all diagonal elements of A_{j+1} are positive, so that there is no test of positivity of α_j for $j > 1$.

Let K_1 denote the number of steps completed during Phase 1, so that $\delta_i = 0$ for $i = 1, ..., K_1$. If $K_1 = n$, A is positive definite and the algorithm terminates. If $K_1 < n$, let

$$\hat{\gamma} = \max_{K_1<i}|A_{ii}| \text{ and } \hat{\xi} = \max_{K_1<i,j<i}|A_{ij}|. \tag{2.8}$$

It is shown in SE90 that the test (2.7) for termination of Phase 1 guarantees that

$$\hat{\gamma} \leq \gamma \text{ and } \hat{\xi} \leq \gamma + \xi. \tag{2.9}$$

If $K_1 < n - 2$, then for $j = K_1 + 1, ..., n - 2$, the value of δ_j is

$$\delta_j = \max\{0, -\alpha_j \ + \max\{\|a_j\|_1, \tau\gamma\}, \delta_{j-1}\} \tag{2.10}$$

This choice of δ_j causes the Gerschgorin intervals of the principal submatrices A_j to contract at each iteration and leads to the following bound:

$$\|E\|_{SE} \leq G + \frac{2\tau}{1-\tau}(G + \gamma) \tag{2.11}$$

where

$$G \leq (n - (k + 1))(\gamma + \xi) \text{ if } K_1 > 0 \text{ and } G \leq \gamma + (n - 1)\xi \text{ if } K_1 = 0. \tag{2.12}$$

The elements δ_{n-1} and δ_n are chosen in a special way that depends on the eigenvalues of the final 2×2 submatrix and still causes (2.12) to be satisfied.

The fact that the bound on $\|E\|$ is linear in n for the SE90 factorization(2.11) and (2.12) and quadratic in n for the GMW81 factorization(2.5) is a key distinction between the methods. In practice, however, both methods usually achieve $\|E\|$ far smaller than these bounds, and $\|E\|$ is often within a factor of 2 of $-\lambda_1(A)$, when $\lambda_1 < 0$. In comparative tests in SE90, the value of $\|E\|$ for the SE90 factorization is almost always smaller than for the GMW81, although the performance of both methods is quite good. The performance of both algorithms is greatly aided by diagonal pivoting strategies employed at each iteration, which do not affect the theoretical properties. The additional cost of both factorizations is at most a small integer multiple of n^2 operations, which is negligible in comparison to the cost of the Cholesky factorization.

Recently, Cheng and Higham [2] have proposed a third type of modified Cholesky factorization, based upon the bounded Bunch-Kaufman pivoting strategy [1]. This factorization differs fundamentally from GMW81 and SE90 in that it adds a non-diagonal matrix to A, by computing the symmetric indefinite factorization LBL^T of a symmetric permutation of A, where L is unit lower triangular and B is block diagonal with 1×1 or 2×2 blocks, and then perturbing B. This approach can be shown to perform well when the condition number of L is not too large. However, as the authors state, the bound on $\|E\|$ is weak if the condition number of LL^T is large, and the worst-case upper bound is exponential in n. It is too early to assess whether this version of the modified Cholesky factorization will have a significant impact in the optimization community.

3 Motivating Example for Change to SE90 Algorithm

All of the known matrices for which $\|E\|_{SE}$ is inordinately large appear to be of the form $A = B + C$ where B is a large positive semi-definite matrix (i.e. $B = B_1 B_1^T$ for some $B_1 \in R^{n\times m}$, $m < n$) and C is an indefinite or negative definite matrix with $\|C\| \ll \|B\|$. (Any symmetric indefinite matrix whose largest positive eigenvalue is much larger in magnitude than its most negative eigenvalue can be written in this form.) The potential for a modified Cholesky factorization to have difficulty on matrices of this type is clear: if Phase 2 begins at or before step m (the rank of B), then the size of δ_j is, according to (2.10), likely to be proportional to $\|B\|$ and therefore large. If, on the other hand, Phase 2 begins after step m, δ_j is likely to be proportional to $\|C\|$, which is, by assumption, much smaller than $\|B\|$. Of course, the structure of A, including the value of m, is not known to the algorithm.

The example in SE90 showing where that algorithm has difficulty,

$$A = \begin{bmatrix} 1,890.3 & -1,705.6 & -315.8 & 3,000.3 \\ -1,705.6 & 1,538.3 & 284.9 & -2,706.6 \\ -315.8 & 284.9 & 52.5 & -501.2 \\ 3,000.3 & -2,706.6 & -501.2 & 4,760.8 \end{bmatrix}, \tag{3.1}$$

is of this form with $m = 1$. Its eigenvalues are 8242.9, -0.248, -0.343, and -0.378. After permuting the largest diagonal element to the (1,1) position, the values of ζ computed from (2.6) are -0.265, -0.451,and -0.517, so that condition (2.7) fails and the algorithm switches immediately to Phase 2 with $K_1 = 0$. Using (2.10), $\delta_1 = 1049.4$ is added to the first diagonal, and this is the ultimate value of $\|E\|$. The large value of δ_1 occurs because the calculation of δ_1 is based upon the Gerschgorin bounds for the large (in magnitude) matrix A.

In contrast, with the GMW81 algorithm we have $\alpha_1 = \beta^2 = 4760.8$ and $\|a_1\|_\infty = 3000.3$. Thus, inequality (2.3) is satisfied with $\delta_1 = 0$, so that no modification is made to the (1,1) diagonal. At the second iteration the algorithm adds 1.033 to the diagonal, which turns out to be its maximum element of E for this problem. This small value of δ_2 results because it is based upon the elements of A_2, and $\|A_2\| \ll \|A\|$.

To avoid modifying "too soon", the remedy for the SE90 algorithm is to relax condition (2.7), the test for continuing Phase 1, to allow Phase 1 to continue even if A_{j+1} will have some small negative diagonal elements. In particular we show in Section 5 that, if Phase 1 continues when there is a suitably positive pivot and

$$\min_i \zeta_i \geq -\mu\gamma, \text{ where } 0 < \mu \leq 1 \tag{3.2}$$

and ζ is defined by (2.6), then the bounds (2.11) on element growth in Phase 1 are only slightly worse; see Thereom 5.1. The advantage of using (3.2) rather than (2.7) is that deferring modification may lead to a smaller $\|E\|$ because the principal submatrix of the later iteration may have smaller elements.

If the test (3.2) is used on example (3.1) with $\mu = 0.1$ (or with any $\mu > 1.1 \times 10^{-4}$), the first step of the *unmodified* Cholesky is allowed to proceed, so that $\delta_1 = 0$, and

$$A_2 = \begin{bmatrix} -0.451 & -0.041 & 0.124 \\ -0.041 & -0.265 & 0.061 \\ 0.124 & 0.061 & -0.517 \end{bmatrix}. \tag{3.3}$$

Since all diagonal elements of A_2 are negative, $K_1 = 1$ and the procedure switches to Phase2, giving $E_{2,2} = 0.3666$, $E_{3,3} = E_{4,4} = 0.6649$. That is, the ratio $\|E\|/(-\lambda_1(A))$ is a very acceptable 1.76, as opposed to a poor 2778 for the SE90 algorithm (and 2.73 for the GMW81 algorithm).

4 The Complete Revised Factorization Algorithm

A complete pseudo-code description of our revised modified Cholesky factorization is given in Algorithm 4.1. The key change from the SE90 algorithm is the one discussed in Section 3: the lookahead condition under which the algorithm switches from Phase 1 to Phase 2 is changed from $\min\{(A_{j+1})_{ii}\} \le \tau\gamma$ for some small positive τ (2.7) to $\min\{(A_{j+1})_{ii}\} \le -\mu\gamma$ for some $\mu \le 1$ (3.2). Our implementation uses $\mu = 0.1$.

Several changes have been made to the algorithm in addition to checking (3.2) as part of continuing in Phase 1:

1. Since we now allow small negative diagonal elements in A_j in Phase 1, we must check that the pivot is positive. The test we insert to proceed with step j of Phase 1 is that the pivot element α_j (the maximum diagonal element of A_j) must satisfy

$$\alpha_j \ge \bar{\tau}\gamma, \quad \text{where } \bar{\tau} = \epsilon^{\frac{2}{3}}. \tag{4.1}$$

   This requirement ensures not only that the pivot is positive, but also that the new algorithm retains a (mainly theoretically useful) bound on the condition number of L analogous to that for the SE90 algorithm.

2. At step j of Phase 1, even if (4.1) is satisfied a branch is made to Phase 2 if

$$\min_{i>j} A_{ii} < -\mu\alpha_j, \tag{4.2}$$

   where μ is the quantity from (3.2). Note that, because (3.2) was satisfied at the previous step of Phase 1, it must be true that $\min_{i>j} A_{ii} \ge -\mu\gamma$. When (4.2) holds, the remaining submatrix A_j tends to have at least one negative eigenvalue that is comparable in magnitude to the other eigenvalues of A_j. In this case, the test (4.2) leads to an earlier termination of Phase 1. Practical experience suggests that this leads to a smaller $\|E\|$; this is illustrated in Section 6.

3. A reduced lower bound, $\bar{\tau}\gamma$, is imposed on the modified diagonal $A_{jj} + \delta_j$, where $\bar{\tau}$ is defined in (4.1). (In the SE90 algorithm, this lower bound is the larger value $\tau\gamma$.) This change leads to two differences between the new algorithm and SE90 when applied to badly conditioned "barely indefinite" matrices for which $|\lambda_1| \ll \|A\|$: $\|E\|$ tends to be smaller with the new algorithm—only slightly larger than $-\lambda_1$; but the condition number of the modified matrix tends to be larger—roughly $1/\bar{\tau} = \epsilon^{-\frac{2}{3}}$

rather than $\epsilon^{-\frac{1}{3}}$ as in SE90. We expect that trading a larger condition number for a smaller modification will be often desirable, for example when the Hessian at the solution is ill-conditioned and the reduced bound allows quadratic convergence to be retained.

4. Special logic is needed to treat the case when $K_1 = n - 1$. (With SE90, step $n - 1$ proceeds only if step n can also be completed, so that this case does not occur.)

The only portions of our code for the modified Cholesky factorization that are not reflected in Algorithm 4.1 are brief special cases to deal with matrices of dimension one, and zero matrices.

Algorithm 4.1 – Revised Modified Cholesky Decomposition Algorithm

Given $A \in \Re^{n \times n}$ symmetric (stored in lower triangle) and $\tau, \bar{\tau}, \mu$ (e.g., $\tau = (macheps)^{\frac{1}{3}}$, $\bar{\tau} = (macheps)^{\frac{2}{3}}$, $\mu = 0.1$), find factorization LL^T of $A + E$, $E \geq 0$

phaseone := true
$\gamma := \max_{1 \leq i \leq n}\{|A_{ii}|\}$
j := 1

(*Phase one, A potentially positive-definite*)
While $j \leq n$ and phaseone = true do
if $max_{j \leq i \leq n}\{A_{ii}\} < \bar{\tau}\gamma$ or $min_{j \leq i \leq n}\{A_{ii}\} < -\mu(max_{j \leq i \leq n}\{A_{ii}\})$
    then phaseone := false (*go to phase two*)
else
    (*Pivot on maximum diagonal of remaining submatrix*)
        $i :=$ index of $max_{j \leq i \leq n}\{A_{ii}\}$
        if $i \neq j$, switch rows and columns of i and j of A
    if $min_{j+1 \leq i \leq n}\{A_{ii} - A_{ij}^2/A_{jj}\} < -\mu\gamma$
        then phaseone := false (*go to phase two*)
        else (* perform *jth* iteration of factorization*)
            $L_{jj} = \sqrt{A_{jj}}$ (*L_{jj} overwrites A_{jj}*)
            For $i := j + 1$ to n do
                $L_{ij} := A_{ij}/L_{jj}$ (*L_{ij} overwrites A_{ij} *)
                For $k := j + 1$ to i do
                    $A_{ik} = A_{ik} - L_{ij}L_{kj}$
            $j := j + 1$
(*end phase one*)

(*Phase two, A not positive-definite*)
if *phaseone* = false and $j = n$ then
    δ (* = E_{nn}*) := $-A_{nn}$+ max $\{\tau(-A_{nn})/(1-\tau), \bar{\tau}\gamma\}$
    $A_{nn} := A_{nn} + \delta$
    $L_{nn} = \sqrt{A_{nn}}$

if *phaseone* = false and $j < n$ then
    $k := j - 1$ (*k = number of iterations performed in phase one*)
    (* Calculate lower Gerschgorin bounds of A_{k+1}*)
        For $i := k + 1$ to n do
            $g_i := A_{ii} - \sum_{j=k+1}^{i-1} |A_{ij}| - \sum_{j=i+1}^{n} |A_{ji}|$

(*Modified Cholesky Decomposition*)
For $j := k+1$ to $n-2$ do
(*Pivot on maximum lower Gerschgorin bound estimate*)
$i :=$ index of $max_{j \le i \le n}\{g_i\}$
if $i \neq j$, switch rows and columns of i and j of A
(*Calculate E_{jj} and add to diagonal*)
$normj := \sum_{i=j+1}^{n} |A_{ij}|$
$\delta(* = E_{nn}*) := \max\{0, -A_{jj} + \max\{normj, \bar{\tau}\gamma\}, \delta prev\}$
if $\delta > 0$ then
$A_{jj} := A_{jj} + \delta$
$\delta prev := \delta$ (* $\delta prev$ will contain $\|E\|_\infty$*)
(*Update Gerschgorin bound estimates*)
if $A_{jj} \neq normj$ then
$temp := 1 - normj/A_{jj}$
for $i := j+1$ to n do
$g_i := g_i + |A_{ij}| * temp$
(*Perform jth iteration of factorization*)
same code as in phase one

(*Final 2×2 submatrix *)
λ_{lo}, λ_{hi} **:=** eigenvalues of $\begin{bmatrix} A_{n-1,n-1} & A_{n,n-1} \\ A_{n,n-1} & A_{n,n} \end{bmatrix}$
$\delta := \max\{0, -\lambda_{lo} + \max\{\tau(\lambda_{hi} - \lambda_{lo})/(1-\tau), \bar{\tau}\gamma\}, \delta prev\}$
if $\delta > 0$ then
$A_{n-1,n-1} := A_{n-1,n-1} + \delta$
$A_{n,n} := A_{n,n} + \delta$
$\delta prev := \delta$
$L_{n-1,n-1} := \sqrt{A_{n-1,n-1}}$ (*overwrites $A_{n-1,n-1}$ *)
$L_{n,n-1} := A_{n,n-1}/L_{n-1,n-1}$ (*overwrites $A_{n,n-1}$ *)
$L_{n,n} := (A_{n,n} - L_{n,n-1}^2)^{1/2}$ (*overwrites $A_{n,n}$ *)
(*End phase two*)

5 Upper Bound on $\|E\|$

A key property of the SE90 factorization is the bound (2.7) on $\|E\|$. In this section we show that the relaxed lookahead strategy of the revised factorization causes only a small growth in this bound. In particular, the term $(\gamma + \xi)$ in (2.8) increases to $(1 + \mu)\gamma + \xi$. (Recall that $\mu \leq 1$; in our implementation $\mu = 0.1$.) Thus the bound grows by at most $(1 + \mu)$ and is still linear in n.

There are two main components in the proof of the bound on $\|E\|$ in SE90. One is the proof (SE90, Lemma 5.1.1 and Theorem 5.1.2) that each δ_j in Phase 2 is less than the magnitude of the most negative Gerschgorin bound of the matrix A_j when the algorithm enters Phase 2. This result is unaffected by the changes in our revised algorithm. The second main component of the proof is the bound on the growth in the elements of A during Phase 1 (SE90, Theorem 5.2.1). This result and proof are modified in a minor way by the new lookahead strategy. For completeness, we include the new statement and proof of this result below. The only new portions are the various terms $\mu\gamma$ below, all of which are absent for the results in SE90 about that algorithm. Note that Theorems 5.1 and 5.2 are true independent of whether pivoting is used at all or what pivoting strategy is used.

Theorem 5.1 *Let $A \in R^{n\times n}$, and let $\gamma = \max\{|A_{ii}|, 1 \leq i \leq n\}$, $\xi = \max\{|A_{ij}|, 1 \leq i < j \leq n\}$. Suppose we perform the standard Cholesky decomposition as described in Phase 1 of Algorithm 4.1 for $k \geq 1$ iterations, yielding the principal submatrix $A_{k+1} \in R^{(n-k)\times(n-k)}$ (whose elements are denoted $(A_{k+1})_{ij}, k+1 \leq i, j \leq n\}$), and let $\hat{\gamma} = \max\{|(A_{k+1})_{ii}|, k+1 \leq i \leq n\}$ and $\hat{\xi} = \max\{|(A_{k+1})_{ij}|, k+1 \leq i < j \leq n\}$. If $(A_{k+1})_{ii} \geq -\mu\gamma, k+1 \leq i \leq n$ for some $\mu \leq 1$, then $\hat{\gamma} \leq \gamma$ and $\hat{\xi} \leq \xi + (1+\mu)\gamma$.*

Proof. Let $A = \begin{bmatrix} B & C^T \\ C & F \end{bmatrix}$, where $B \in R^{k\times k}$, $C \in R^{(n-k)\times k}$, $F \in R^{(n-k)\times(n-k)}$. After k iterations of the Cholesky factorization, the first k columns of the Cholesky factor L have been determined; denote them by $\begin{bmatrix} \bar{L} \\ M \end{bmatrix}$ where $\bar{L} \in R^{k\times k}$ is triangular and $M \in R^{(n-k)\times k}$. Then

$$B = \bar{L}\bar{L}^T, \quad C = M\bar{L}^T, \quad F = MM^T + A_{k+1}. \tag{5.1}$$

Let m_i^T denote the ith row of M. From (5.1), $F_{ii} = \|m_i^T\|_2^2 + (A_{k+1})_{ii}, k+1 \leq i \leq n$, so that from $F_{ii} \leq \gamma$ and $(A_{k+1})_{ii} \geq -\mu\gamma$,

$$\|m_i^T\|_2^2 \leq (1+\mu)\gamma. \tag{5.2}$$

Thus for any off-diagonal element of A_{k+1}, (5.1), (5.2), and the definition of ξ imply

$$|(A_{k+1})_{ij}| \leq |F_{ij} - (m_i^T)(m_j^T)^T| \leq \xi + (1+\mu)\gamma, \tag{5.3}$$

which shows that $\hat{\xi} \leq \xi + (1+\mu)\gamma$. Also for all the diagonal elements of A_{k+1}, $(A_{k+1})_{ii} \geq -\mu\gamma$, $\mu \leq 1$, (5.1), and the definition of γ imply

$$-\mu\gamma \leq (A_{k+1})_{ii} \leq F_{ii} \leq \gamma \tag{5.4}$$

which shows that $\hat{\gamma} \leq \gamma$ and completes the proof. □

The only other change in the revised algorithm that could affect the bound on $\|E\|$ is the use of $\bar{\tau}$ where SE90 uses τ. Since $\bar{\tau} < \tau$, this affects the statement of the main result but not the bound on $\|E\|$. The new growth bound is given below; it is a minor modification of Theorem 5.3.2 of SE90.

Theorem 5.2 *Let A, γ, and ξ be defined as in Theorem 5.1, and suppose the modified Cholesky factorization algorithm 4.1 is applied to A, resulting in the factorization LL^T of $A + E$. If A is positive-definite and at each iteration $L_{jj}^2 \geq \bar{\tau}\gamma$, then $E = 0$. Otherwise, E is a nonnegative diagonal matrix, with*

$$\|E\| \leq \text{Gersch} + \frac{2\tau}{1-\tau}(\text{Gersch} + \gamma), \tag{5.5}$$

where Gersch *is the maximum of the negative of the lower Gerschgorin bounds $\{\ g_i\ \}$ of A_{k+1} that are calculated at the start of Phase two. If $k = 0$ then* Gersch $\leq \gamma + (n-1)\xi$, *otherwise*

$$\text{Gersch} \leq (n - (k+1))((1+\mu)\gamma + \xi). \tag{5.6}$$

6 Computational Results

We have tested our revised factorization method, and the GMW81 and SE90 methods, on the problems where the SE90 method had difficulties, as well as on the broad test set from SE90 and a modification of one of these problem sets designed to be especially difficult for our methods for reasons described below. This section summarizes and analyzes the computational results.

As mentioned in Section 1, the modifications to the SE90 algorithm were motivated in a large part by the matrices sent to us by David Gay, Michael Overton and Margaret Wright. These matrices are condensed primal-dual matrices used in barrier methods for constrained optimization. The 33 matrices sent to us were from problems where the overall optimization method using the SE90 factorization performed less well than the same optimization method using other modified Cholesky factorizations, including GMW81. For each problem, Gay, Overtion and Wright attempted to locate the first optimization iteration where the algorithm using SE90 took a poorer step than the algorithm using other modified Cholesky factorizations, and sent the Hessian matrix from this iteration. It turned out that for two-thirds of these matrices, the SE90 algorithm was adding more than GMW81, by as much as a factor of 10^2 to 10^7 in 8 cases. The problems are quite small, with all but two having dimension between 6 and 15, and the remaining two having dimension 26 and 55.

Table 6.1 summarizes the performance of the GMW81 and SE90 algorithms, and our new Algorithm 4.1, on these 33 problems. The first column encodes the problem as follows: the set (A is the initial set sent to us, B a second, later set sent to us after we had made some but not all of the modifications reported in this paper), the dimension, and the sequence number within this set and dimension. Columns 2-4 report the ratio of $\|E\|/(-\lambda_1(A))$ for each factorization (for problem B13_1, which is positive definite, this

column contains $\|E\|$ instead). Columns 5-7 report the integer part of the base ten log of the l_2 condition number of $A + E$.

The results show that the new algorithm produces a reasonable value of $\|E\|$ in all cases. The ratio $\|E\|/(-\lambda_1(A))$ is less than 2.4 on all 33 problems, less than 2 on 31 of the 33, and less than 1.4 for 24 of the 33 problems. The value of $\|E\|$ is smaller than that produced by the GMW81 algorithm on all 33 problems except the positive definite matrix where both produce $E = 0$. The values of $\|E\|/(-\lambda_1(A))$ produced by GMW81 are generally in the range 2-5 for these problems. It is not clear, however, that this larger value makes the GMW81 algorithm any less effective in an optimization context. The value of $\|E\|/(-\lambda_1(A))$ for the new algorithm is essentially the same as for SE90 in 10 of the 33 cases and lower in the other 23.

The results also show that the condition numbers of $A + E$ produced by the new algorithm are considerably higher than for the SE90 algorithm, with 13 of the 33 as high as 10^9 to 10^{11}. As discussed in Section 4, this stems directly from the reduction in the minimum allowable value of $(A_{jj} + \delta_j)$ from $(macheps)^{1/3}\gamma$ to $(macheps)^{2/3}\gamma$. This reduction, however, allows the algorithm to produce values of $\|E\|$ hardly larger than $-\lambda_1(A)$ on indefinite problems where $-\lambda_1(A)$ is very small compared to $\|A\|$. The condition numbers produced by the GMW81 algorithm are almost always smaller than those produced by the new algorithm, although the two largest condition numbers produced by GMW81 on this test set, both roughly 10^{12}, exceed the largest condition numbers produced by the new algorithm. It should be noted that the original matrices in these problems are themselves extremely ill-conditioned, and it is important for the modified Cholesky to retain this property.

The change in performance of the new algorithm versus the SE90 algorithm on these problems is directly related to its ability to defer adding to the diagonal until a later iteration of the factorization. The new algorithm begins adding to the diagonal at the same iteration as SE90 in 10 cases (all where SE90 already performed satisfactorily) and later in the remaining 23 cases. In 8 cases it begins adding only one iteration later, but even this can lead to $\|E\|$ being orders of magnitude smaller as was shown by the example in Section 3. In some cases the new algorithm begins adding 7-10 iterations later than SE90, on problems of dimension no greater than 15. The GMW81 and the new algorithm are very similar in when they begin adding to the diagonal: they begin at the same iteration in 21 of the 33 cases, with GMW81 beginning earlier in 7 of the remaining 12 and later in the other 5. The test (4.2) has an impact on 5 of these 33 problems (A6_3, A6_10, A6_12, A6_14, and B8_1), reducing $\|E\|/(-\lambda_1(A))$ from between 2 and 3.4 to 1.3 or less while also reducing the condition number of $A + E$ by about 1 order of magnitude in comparison to the new algorithm without (4.2).

We had received one other report of difficulties from users of the SE90 algorithm, from Wolfgang Hartmann of SAS concerning problems arising in ridge regression. In the

Table 6.1: Performance of existing and new methods on indefinite Hessian matrices

Problem	$\|E\|/(-\lambda_1(A))$			Log_{10} Cond'n number of $A+E$		
	GMW81	SE90	revised SE90	GMW81	SE90	revised SE90
A6_1	1.36	3.57e+02	1.08	5	1	9
A6_2	4.84	1.18	1.18	3	5	7
A6_3	4.84	1.19	1.20	4	5	6
A6_4	2.50	1.27	1.27	5	5	8
A6_5	2.34	6.50	1.44	5	3	9
A6_6	1.69	2.94	1.20	8	5	10
A6_7	1.95	4.61	1.33	12	5	10
A6_8	1.95	6.61	1.13	8	5	10
A6_9	1.95	47.22	1.12	8	5	10
A6_10	5.88	5.39e+06	1.07	8	1	11
A6_11	2.33	7.25e+06	1.64	8	1	7
A6_12	4.84	1.19	1.20	4	5	6
A6_13	2.18	1.32	1.32	2	5	6
A6_14	4.84	1.19	1.20	4	5	6
A6_15	5.18	1.09	1.09	2	5	5
A6_16	2.18	1.32	1.32	2	5	6
A6_17	1.52	1.24	1.24	2	5	6
A13_1	2.25	8.93e+03	1.18	10	5	10
A13_2	2.59	1.50e+04	1.31	8	5	10
A15_1	2.42	2.54e+07	1.89	9	5	11
A15_2	2.37	3.89e+05	1.44	9	3	10
A15_3	1.95	2.18	1.50	6	5	10
B6_1	4.90	52.41	1.77	3	1	8
B6_2	4.49	45.86	2.31	2	1	7
B7_1	1.66	3.45	1.06	2	2	2
B7_2	1.93	11.00	1.30	2	1	7
B7_3	1.96	6.99	1.22	2	1	6
B7_4	1.92	5.32	1.18	2	1	6
B8_1	4.16	871.2	1.27	12	5	10
B13_1	0	27.14 (abs)	0	9	5	9
B13_2	1.76	7.84	1.29	7	5	10
B26_1	9.83	2.23	2.36	1	3	7
B55_1	3.50	1.71	1.71	1	5	6

example sent to us by Hartmann, $n = 6$, and the matrix

$$\begin{bmatrix} 14.8253 & -6.4243 & 7.8746 & -1.2498 & 10.2733 & 10.2733 \\ -6.4243 & 15.1024 & -1.1155 & -0.2761 & -8.2117 & -8.2117 \\ 7.8746 & -1.1155 & 51.8519 & -23.3482 & 12.5902 & 12.5902 \\ -1.2498 & -0.2761 & -23.3482 & 22.7967 & -9.8958 & -9.8958 \\ 10.2733 & -8.2117 & 12.5902 & -9.8958 & 21.0656 & 21.0656 \\ 10.2733 & -8.2117 & 12.5902 & -9.8958 & 21.0656 & 21.0656 \end{bmatrix} \tag{6.1}$$

is positive semidefinite with one zero eigenvalue and five positive eigenvalues ranging from 5 to 82. The positive semi-definite case can be considered a limiting case of the class of problems that motivated our revision. The SE90 algorithm adds 7.50 to the diagonal at iterations 3 through 6, which is undesirable. The GMW81 algorithm adds 1.67×10^{-14} at iteration 6 and produces a condition number of 4.9×10^{15} for $A + E$, whereas our new algorithm adds 1.90×10^{-9} at iteration 6 and produces a condition number of 8.7×10^{10}. Both seem reasonable; the higher value of δ_6 and lower condition number from the new algorithm, compared to GMW81, stem directly from our tolerance on the lowest allowable value of $(A_{jj} + \delta_j)$ discussed in Section 4.

We also reran all the test problems reported in SE90. These consist of 120 randomly generated problems, 10 each of dimension 25, 50, and 75 for each of four eigenvalue ranges: -1 to 1, -10^{-4} to -1, -1 to 10^4 and one negative eigenvalue, and -1 to 10^4 and three negative eigenvalues.

The behavior of the new algorithm on the first two sets of problems (eigenvalue ranges [−1,1] and [-10^{-4}, −1]) is identical to the SE90 algorithm for all problems. As reported in SE90, for both of these classes of matrices, the SE90 algorithm (and the new algorithm) produce values of $\|E\|/(-\lambda_1(A))$ quite close to 1 and considerably lower than the GMW81 algorithm (by 1-2 orders of magnitude), while also producing much smaller condition numbers than the GMW81 algorithm (by about 6 orders of magnitude).

The behavior of the factorization algorithms on the sets with eigenvalue range -1 to 10^4 are very similar to the behavior on the Gay/Overton/Wright problems that help motivate this paper, since their characteristics are very similar. Indeed, the example in Section 3, given in SE90, came from the [-1, 10^4] set with $n = 25$ and three negative eigenvalues, and was the only bad case for the SE90 algorithm of the 120 test problems in that paper. In this paper, we include the results for the one negative eigenvalue set with $n = 75$ (Figures 6A,B), as they are typical but more marked than the $n = 25$ and 50 results. We also include a new set with $n = 75$, eigenvalue range -1 to 10^4, and nine negative eigenvalues (Figures 6C-E), as it is a more extreme example of the problems with the SE90 algorithm than the three negative eigenvalue sets.

The results on these two test sets again show that the values of $\|E\|/(-\lambda_1(A))$ produced by the new algorithm are very good, generally about 1.5 for the first set and 2 for the second. The values produced by the GMW81 algorithm are slightly higher but also very good. The values produced by the SE90 algorithm on the second set are very high (between 70 and 200) in four of the cases; for the first set they are satisfactory but the new algorithm is better. As with some of the Gay/Overton/Wright problems, the condition numbers produced by the new algorithm in these cases are around 10^{10}, while for the GMW81 algorithm they are around 10^5.

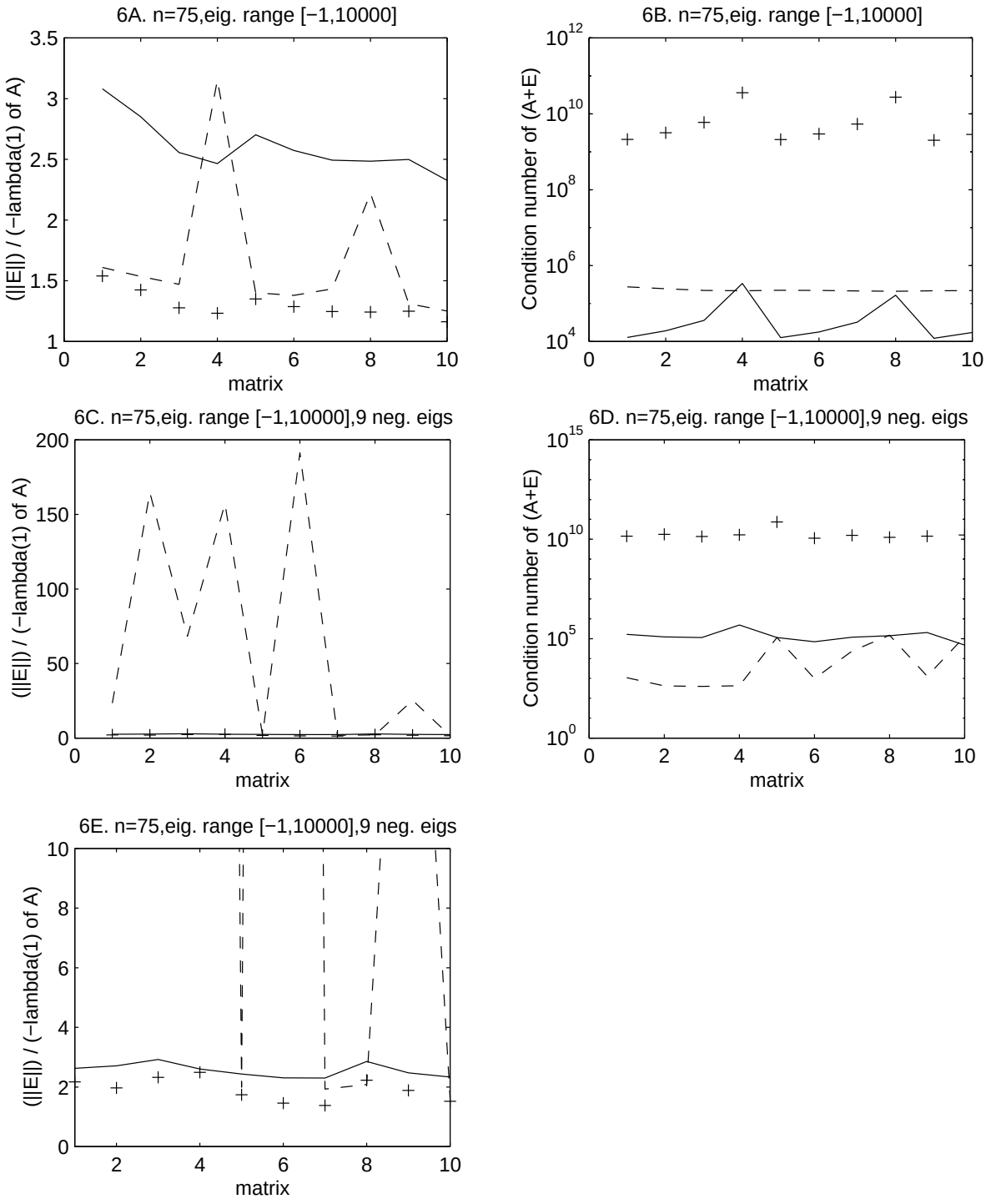

Figure 6: (A,B): Performance of existing and new methods on 10 matrices, each containing one negative eigenvalue. (C-E): Performance of existing and new methods on 10 matrices, each containing nine negative eigenvalues.
Methods: GMW81 ——, SE90 − − −, revised SE90 + + + .

In summary, these results indicate that the modifications introduced in this paper have removed the known difficulties with the SE90 algorithm. The new algorithm produces values of $\|E\|/(-\lambda_1(A))$ in the range 1 to 2.5 for all test matrices considered, including all that are problematic for the SE90 algorithm. The values of $\|E\|/(-\lambda_1(A))$ are virtually always lower than those produced by the GMW81 algorithm, sometimes considerably so. The modifications result in condition numbers of $A+E$ of order $(macheps)^{-2/3}$ in cases when A is barely indefinite ($0 < -\lambda_1(A) \ll \|A\|$). The matrices produced by the GMW81 algorithm generally are better conditioned than those produced by the new algorithm in these cases, although the highest condition numbers produced by the GMW81 algorithm are higher than for the new algorithm. The new algorithm, like SE90, produces very well conditioned matrices in the other types of test cases.

In our opinion, these test results indicate good performance for both the GMW81 algorithm and the new algorithm. Which is used in an optimization context may depend upon the context or upon factors other than those considered in this paper. For example, SE90 has proven useful for large scale codes, including multifrontal approaches, where one does not want to process the full matrix A at once [4]. Here the fact that GMW81 requires a preprocessing step that requires all of A (to compute ξ, which is not used in SE90 or the new algorithm) is the critical difference. (Pivoting is not used in these implementations; recall that this does not weaken the theoretical properties of our algorithm.) In a different context, the SE90 algorithm has led to very good performance when used as a preconditioner in conjugate gradient codes in the LANCELOT software package [3]; it has also been used in this manner by [11]. Additionally, it has proven to be useful in ensuring that the Winget factors within element-by-element preconditioners are definite [6], and has also been implemented in a block version of the factorization [5]. Finally, the results of this paper show that the new algorithm may be a useful way to obtain rough estimates of $-\lambda_1(A)$ in cases where this is useful, for example some trust region methods. For general optimization applications, both factorizations are likely to continue to be used; the lower *a priori* bound on $\|E\|$ for GMW81 and the new algorithm may not be a determining factor since the results of SE90 and this section continue to show that both algorithms reliably produce values of $\|E\|$ that are far lower than these bounds in practice. If our test problems are a good indication, however, the apparently greater robustness of our new method in not producing poor values of $\|E\|$ or excessively high condition numbers of $A + E$ may be an asset.

Acknowledgements. The authors thank David Gay, Michael Overton and Margaret Wright for alerting them to the difficulties of our original modified Cholesky algorithm on their problems from primal-dual methods and for supplying sample test problems. The authors also thank Margaret Wright for many helpful, detailed suggestions regarding the presentation of this paper.

References

[1] C. Ashcraft, R. G. Grimes, and J. G. Lewis, *Accurate symmetric indefinite*

linear equation solvers, SIAM J. Matrix Anal. Appl., 20 (1998) pp. 513–561.

[2] S. H. Cheng, and N. J. Higham, *A Modified Cholesky Algorithm Based on a Symmetric Indefinite Factorization*, SIAM J. Matrix Anal. Appl., 19 (1998) pp. 1097–1110.

[3] A. R. Conn, N. I. M. Gould and P. L. Toint, *Numerical experiments with the* LANCELOT *package (Release A) for large-scale nonlinear optimization*, Math. Programming, 73 (1996), pp. 73–110.

[4] A. R. Conn, N. I. M. Gould and P. L. Toint, LANCELOT*: a Fortran package for large-scale nonlinear optimization (Release A)*, Springer Series in Computational Mathematics, 17, Springer Verlag, Heidelberg, Berlin, New York, 1992.

[5] M. J Daydé, *A Block Version of the Eskow-Schnabel Modified Cholesky Factorization*, Rapport Technique ENSEEIHT-IRIT RT/APO/95/8, (1995).

[6] M. J Daydé, J.-Y. L'Excellent, and N. I. M. Gould *Element-by-Element Preconditioners for Large Partially Separable Optimization Problems*, SIAM J. Sci. Statist. Comput., 18 (1997) pp. 1767–1787.

[7] J. E. Dennis, and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice-Hall, Englewood Cliffs, NJ, 1983, reprinted by SIAM, Philadelphia, PA, 1996.

[8] D. M. Gay, M. L. Overton and M. H. Wright, A primal-dual interior point method for nonconvex nonlinear programming in *Advances in Nonlinear Prgramming* (Y. Yuan, ed), Kluwer Academic Publishers, Dordrecht, 1998, pp. 31–36.

[9] P. E. Gill, and W. Murray, *Newton-type methods for unconstrained and linearly constrained optimization*, Math. Programming, 28 (1974), pp. 311–350.

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

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

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