SIAM J. MATRIX ANAL. APPL.
Vol. 17, No. 1, pp. 35–46, January 1996





# ON THE STABILITY OF CHOLESKY FACTORIZATION FOR SYMMETRIC QUASIDEFINITE SYSTEMS*

PHILIP E. GILL†, MICHAEL A. SAUNDERS‡, AND JOSEPH R. SHINNERL†

**Abstract.** Sparse linear equations $Kd = r$ are considered, where $K$ is a specially structured symmetric *indefinite* matrix that arises in numerical optimization and elsewhere. Under certain conditions, $K$ is *quasidefinite*. The Cholesky factorization $PKP^T = LDL^T$ is then known to exist for any permutation $P$, even though $D$ is indefinite.

Quasidefinite matrices have been used successfully by Vanderbei within barrier methods for linear and quadratic programming. An advantage is that for a sequence of $K$'s, $P$ may be chosen once and for all to optimize the sparsity of $L$, as in the positive-definite case.

A preliminary stability analysis is developed here. It is observed that a quasidefinite matrix is closely related to an unsymmetric positive-definite matrix, for which an $LDM^T$ factorization exists. Using the Golub and Van Loan analysis of the latter, conditions are derived under which Cholesky factorization is stable for quasidefinite systems. Some numerical results confirm the predictions.

**Key words.** indefinite systems, symmetric quasidefinite (sqd) systems, unsymmetric positive-definite systems, backward stability, condition number, barrier methods, linear programming

**AMS subject classifications.** 49D37, 65F05, 65K05, 90C30

## 1. Introduction.

We define a matrix $K$ to be *symmetric quasidefinite* (sqd) if there exists a permutation matrix $\Pi$ that reorders $K$ to the form

$$\Pi K \Pi^T = \begin{pmatrix} H & A^T \\ A & -G \end{pmatrix}, \tag{1.1}$$

where $H \in \mathbb{R}^{n\times n}$ and $G \in \mathbb{R}^{m\times m}$ are symmetric and positive definite. Such a $K$ is indefinite and nonsingular. Vanderbei [Van91], [Van94] has shown that sqd matrices are *strongly factorizable*; i.e., for *every* permutation $P$ there exist a diagonal $D$ and a unit lower-triangular $L$ such that

$$PKP^T = LDL^T. \tag{1.2}$$

We refer to (1.2) as a Cholesky factorization, while emphasizing that $K$ is indefinite and $D$ has both positive and negative diagonals. The usual stability analysis therefore does not apply, and the factorization may be unstable.

An example sqd matrix is

$$K = \begin{pmatrix} 1 & 1 \\ 1 & -\epsilon \end{pmatrix} = \begin{pmatrix} 1 & \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & \\ & -(1+\epsilon) \end{pmatrix} \begin{pmatrix} 1 & 1 \\ & 1 \end{pmatrix}. \tag{1.3}$$

The Cholesky factors exist for all values of $\epsilon$, and can be computed accurately in finite precision for any $\epsilon$. The symmetrically permuted system

$$PKP^T = \begin{pmatrix} -\epsilon & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & \\ -\frac{1}{\epsilon} & 1 \end{pmatrix} \begin{pmatrix} -\epsilon & \\ & 1+\frac{1}{\epsilon} \end{pmatrix} \begin{pmatrix} 1 & -\frac{1}{\epsilon} \\ & 1 \end{pmatrix} \tag{1.4}$$

* Received by the editors July 13, 1993; accepted for publication (in revised form) by S. Hammarling January 17, 1995. This research was supported by Department of Energy contract DE-FG03-92ER25117, National Science Foundation grants DMI-9204208 and DMI-9204547, and Office of Naval Research grant N00014-90-J-1242.

† Department of Mathematics, University of California, San Diego, La Jolla, CA 92093-0012 (pgill@ucsd.edu and jshinnerl@ucsd.edu).

‡ Department of Operations Research, Stanford University, Stanford, CA 94305-4022 (mike@sol-michael.stanford.edu).

has Cholesky factors for any nonzero $\epsilon$, but as noted in [Van91], the factorization becomes unstable in finite-precision arithmetic as $|\epsilon| \to 0$.

Strong factorizability is particularly attractive when $K$ is large and sparse and a direct factorization method is used to solve the linear system of equations

$$Kd = r. \tag{1.5}$$

As with positive-definite systems, we choose $P$ in (1.2) to reduce fill-in during the Cholesky factorization. If several similar systems are to be solved, we would like to use the same "ideal" $P$ for each system, as long as the associated factorizations are stable. In this paper we examine conditions under which Cholesky factorization may be used reliably on an sqd matrix $K$. For the example in (1.3)–(1.4), the analysis predicts (of course) that $|\epsilon|$ should not be too small. For certain systems arising in constrained optimization, it predicts that Cholesky factorization should be stable until the iterates are in a small neighborhood of the solution.

**1.1. Notation.** When discussing permutations $P$, we speak of "sparsity interchanges" and "stability interchanges" to indicate the usual criteria for choosing $P$. The spectral condition number is $\kappa_2(K) \equiv \|K\|_2\|K^{-1}\|_2$. The following symbols are used for matrices:

- $A$, $G$, and $H$ are the block components of an arbitrary sqd matrix $K$.
- $B$ is an arbitrary square nonsingular matrix whose triangular factorization $B = LU = LDM^T$ exists in exact arithmetic without row or column interchanges. ($D$ is diagonal, $L$ and $M$ are unit lower triangular, and $DM^T \equiv U$. Although such a factorization does not exist for all nonsingular $B$, when it does exist, the factors are unique.)
- $C$ is a square matrix that is unsymmetric but positive definite.
- $T$ and $S$ are the symmetric and skew-symmetric parts of $C$: $T = (C+C^T)/2$, $S = (C - C^T)/2$, and $C = T + S$.
- $LDL^T$ denotes Cholesky factors of a symmetric matrix: $L$ unit triangular, $D$ diagonal and possibly *indefinite*.
- $LBL^T$ denotes factors of a symmetric indefinite matrix: $L$ unit triangular, $B$ block-diagonal with blocks of order 1 or 2.

**2. Connection with the unsymmetric positive-definite case.** We seek conditions on sqd matrices $K$ that allow stable computation of the Cholesky factorization $PKP^T = LDL^T$ for every permutation $P$. There is no loss of generality in assuming $\Pi = I$ in (1.1). With this convention, observe that

$$K = \begin{pmatrix} H & A^T \\ A & -G \end{pmatrix} \equiv \bar{K}\bar{I}, \tag{2.1}$$

where

$$\bar{K} \equiv \begin{pmatrix} H & -A^T \\ A & G \end{pmatrix} \quad \text{and} \quad \bar{I} \equiv \begin{pmatrix} I_n & \\ & -I_m \end{pmatrix}, \tag{2.2}$$

where $I_n$ and $I_m$ denote the identity matrices of order $n$ and $m$. The matrix $\bar{K}$ is *unsymmetric positive definite*; i.e., $x^T\bar{K}x > 0$ for all nonzero $x$. The main idea of this paper is that (2.1) can be used to characterize the stability of algorithms for symmetric quasidefinite matrices in terms of the stability of Gaussian elimination for unsymmetric positive-definite matrices.

With $\bar{I}$ as above, let $\tilde{I} \equiv P\bar{I}P^T$ for some permutation $P$. The matrix $\tilde{I}$ is diagonal with diagonal entries 1 and $-1$; thus, in any product of the form $\tilde{A} = A\tilde{I}$, $\tilde{A}$ is equal to $A$ with some of its columns scaled by $-1$. Now for every permutation $P$,

$$PKP^T = P\bar{K}\bar{I}P^T = P\bar{K}P^T(P\bar{I}P^T) = (P\bar{K}P^T)\tilde{I}. \tag{2.3}$$

It follows that

$$PKP^T = LDL^T \quad \text{if and only if} \quad P\bar{K}P^T = L\tilde{D}M^T, \tag{2.4}$$

where $\tilde{D} \equiv D\tilde{I}$ and $M \equiv \tilde{I}L\tilde{I}$. The matrices $D$ and $\tilde{D}$ are diagonal, and $L$ and $M$ are unit lower triangular as required. Relations (2.1) and (2.4) can be construed as an alternative proof of Vanderbei's theorem on the strong factorizability of symmetric quasidefinite matrices. For if $K$ is sqd, then $\bar{K}$ and hence $P\bar{K}P^T$ are positive definite; therefore the $LU$ factorization of $P\bar{K}P^T$ exists (cf. [GV89, p. 140]); hence, by (2.4), the $LDL^T$ factorization of $PKP^T$ exists as well.

Since only column signs are involved, it is trivial to show that (2.4) holds in finite precision. If $\hat{L}$ and $\hat{D}$ are the computed factors of $PKP^T$, then $\hat{L}$, $\hat{D}\tilde{I}$, and $\hat{M}^T = \tilde{I}\hat{L}^T\tilde{I}$ are the computed $LU$ factors of $P\bar{K}P^T$. Hence any conditions that ensure stability for the factorization $P\bar{K}P^T = LDM^T$ will also ensure stability for $PKP^T = LDL^T$. In particular, it is safe to factor the quasidefinite matrix $PKP^T$ *without stability interchanges* if and only if it is safe to factor the unsymmetric positive-definite matrix $P\bar{K}P^T$ without stability interchanges.

**3. When stability interchanges are unnecessary.** Throughout this section, we assume that $C$ is an unsymmetric positive-definite matrix. Let $T$ and $S$ be the symmetric and skew-symmetric parts of $C$. Then it is safe to factor $C$ without stability interchanges if

(i) $S$ is not too large compared to $T$; and
(ii) $T$ is not too ill-conditioned.

This follows from results of Golub and Van Loan [GV79], [GV89], which we summarize next.

**3.1. Theorems of Golub and Van Loan.** Let $C = LDM^T$. In the backward error analysis of Gaussian elimination, it is shown that the computed solution $\hat{x}$ to the system $Cx = r$ is the exact solution of the perturbed system $(C + \Delta C)\hat{x} = r$, where the size of $\Delta C$ is bounded by an expression involving the sizes of the computed factors of $C$; say, $\hat{L}$, $\hat{D}$, and $\hat{M}^T$ (cf. (3.1) below). Algorithms that produce $\hat{L}$, $\hat{D}$, and $\hat{M}^T$ of sufficiently bounded size are therefore considered stable.

For general $C$, row or column interchanges are necessary to ensure the existence of the factors, and to prevent them from having large elements. For positive-definite $C$, however, the following theorems can be used with Assumption 3.1 to obtain a satisfactory bound on the sizes of the computed factors without stability interchanges.

(When applied to vectors or matrices, the symbols $|\cdot|$ and $\leq$ are to be interpreted componentwise. The symbol $\mathbf{u}$ denotes the unit round-off, and all floating-point calculations are assumed to conform to the "standard model" described in [GV89, pp. 61–62].)

ASSUMPTION 3.1 (see [GV89, p. 141]). *For some scalar $\gamma$ of moderate size,*

$$\| |\hat{L}|\,|\hat{D}|\,|\hat{M}^T| \|_F \leq \gamma \| |L|\,|D|\,|M^T| \|_F.$$

THEOREM 3.1 (see [GV79, p. 88]). *Let $C \in \mathbb{R}^{n\times n}$ be positive definite and set $T = (C + C^T)/2$ and $S = (C - C^T)/2$. If $C = LDM^T$, then*

$$\| |L|\,|D|\,|M^T| \|_F \leq n\left(\|T\|_2 + \|ST^{-1}S\|_2\right).$$

THEOREM 3.2 (see [GV89, p. 136, Eqn. (4.1.3)]). *Let $B \in \mathbb{R}^{n\times n}$ be a matrix whose $LDM^T$ factorization exists, and let $\hat{L}$, $\hat{D}$, and $\hat{M}$ be the computed factors. Let $\hat{x}$ denote the computed solution to the system $Bx = b$, obtained by the usual methods of forward and backward substitution* (cf. [GV89, p. 97, Algorithm 3.2.3]). *Then $(B + \Delta B)\hat{x} = b$, with*

$$|\Delta B| \leq n\mathbf{u}\left(3|B| + 5|\hat{L}|\,|\hat{D}|\,|\hat{M}^T|\right) + \mathcal{O}(\mathbf{u}^2). \tag{3.1}$$

From Assumption 3.1 and these theorems, it follows that the computed solution $\hat{x}$ to the positive-definite system $Cx = r$ satisfies $(C + \Delta C)\hat{x} = r$, with

$$\|\Delta C\|_2 \leq \|\Delta C\|_F \leq \mathbf{u}\left(3n\|C\|_F + 5\gamma n^2\left(\|T\|_2 + \|ST^{-1}S\|_2\right)\right) + \mathcal{O}(\mathbf{u}^2). \tag{3.2}$$

Since $\|T\|_2 \leq \|C\|_2$, we have

$$\|T\|_2 + \|ST^{-1}S\|_2 \leq \left(1 + \frac{\|ST^{-1}S\|_2}{\|C\|_2}\right)\|C\|_2. \tag{3.3}$$

RESULT 3.1 (see [GV79, p. 92] and [GV89, p. 141]). *If $C$ is positive definite, the factorization $C = LDM^T$ is stable if $\omega(C)$ is not too large, where*

$$\omega(C) \equiv \frac{\|ST^{-1}S\|_2}{\|C\|_2}. \tag{3.4}$$

**3.2. An alternative indicator.** Because it may not always be clear how the structure of the matrix $ST^{-1}S$ depends on the structure of the original matrix $C$, we observe that $\omega(C) \leq \theta(C)$, where $\theta(C)$ is defined next.

RESULT 3.2. *If $C$ is positive definite, the factorization $C = LDM^T$ is stable if $\theta(C)$ is not too large, where*

$$\theta(C) \equiv \left(\frac{\|S\|_2}{\|T\|_2}\right)^2 \kappa_2(T). \tag{3.5}$$

When $\|S\|$ is not much larger than $\|T\|$, and $T$ is not too ill-conditioned, $\theta(C)$ may provide an adequate guarantee of numerical stability. The straightforward dependence of $\theta(C)$ on $T$ and $S$ makes it easier to estimate than $\omega(C)$.

In certain contexts, however, $\theta(C)$ may be arbitrarily larger than $\omega(C)$. For example, suppose $C$ has the form of $\bar{K}$ in (2.2), with $H = \beta I$, $G = (1/\beta)I$, and $A = \beta^2 I$. It is easily shown that as $\beta \to \infty$, $\theta(C) = \mathcal{O}(\beta)\omega(\mathcal{C})$. Thus, a large value of $\theta(C)$ should not be automatically interpreted to mean that stability interchanges are necessary.

**4. Application to quasidefinite matrices.** For our purposes, the role of $C$ is played by $P\bar{K}P^T$ in §2. Since it is easily shown that $\omega$ and $\theta$ are invariant under symmetric permutations of their arguments, we assume $C = \bar{K}$ in (2.2). In this case,

$$T = \frac{1}{2}(\bar{K} + \bar{K}^T) = \begin{pmatrix} H & \\ & G \end{pmatrix} \quad \text{and} \quad S = \frac{1}{2}(\bar{K} - \bar{K}^T) = \begin{pmatrix} & -A^T \\ A & \end{pmatrix},$$

so that

$$ST^{-1}S = \begin{pmatrix} -A^TG^{-1}A & \\ & -AH^{-1}A^T \end{pmatrix}.$$

From the Golub and Van Loan analysis, stability of the factorization can be guaranteed if $\omega(\bar{K}) \equiv \|ST^{-1}S\|_2/\|\bar{K}\|_2$ is not too large. In terms of $K$ rather than $\bar{K}$, we therefore have the following result for sqd matrices of the form (1.1).

RESULT 4.1. *If $K$ is sqd, the factorization $PKP^T = LDL^T$ is stable for every permutation $P$ if $\omega(K)$ is not too large, where*

$$\omega(K) \equiv \frac{\max\{\|A^TG^{-1}A\|_2, \|AH^{-1}A^T\|_2\}}{\|K\|_2}. \tag{4.1}$$

As in (3.4)–(3.5), we have $\omega(K) \le \theta(K)$, where the latter is readily computed in terms of $A$, $H$, $H^{-1}$, $G$, and $G^{-1}$.

RESULT 4.2. *If $K$ is sqd, the factorization $PKP^T = LDL^T$ is stable for every permutation $P$ if $\theta(K)$ is not too large, where*

$$\theta(K) \equiv \left(\frac{\|A\|_2}{\max\{\|G\|_2, \|H\|_2\}}\right)^2 \max\{\kappa_2(G), \kappa_2(H)\}. \tag{4.2}$$

For example, suppose $\|H\|_2 \ge \|G\|_2$ and $\|G^{-1}\|_2 \ge \|H^{-1}\|_2$. Then

$$\theta(K) \le \frac{\|A\|_2^2}{\|H\|_2}\|G^{-1}\|_2.$$

In general,

(i) $\|A\|_2$ must not be too large compared to $\|H\|_2$ and $\|G\|_2$; and
(ii) $\operatorname{diag}(H, G)$ must not be too ill-conditioned.

**5. The condition number of a quasidefinite system.** To assess the accuracy of computed solutions to $Kd = r$ with $K$ sqd as in (2.1), we must consider both the backward stability of the factorization $PKP^T = LDL^T$ and the forward sensitivity of $d$ to perturbations in $K$. That is, given that our computed solution $\hat{d}$ satisfies the perturbed system

$$(K + \Delta K)\,\hat{d} = r, \tag{5.1}$$

how close is $\hat{d}$ to $d$, the true solution? The usual sensitivity bound takes the form

$$\frac{\|d - \hat{d}\|}{\|d\|} \le \frac{\alpha}{1 - \alpha}, \quad \text{where} \quad \alpha = \frac{\|\Delta K\|}{\|K\|}\kappa_2(K). \tag{5.2}$$

For general $K$, the relative perturbation $\|\Delta K\|/\|K\|$ cannot be suitably bounded without the use of stability interchanges. When $K$ is sqd, however, (3.2) and (3.3) give a bound on this perturbation that is essentially proportional to $1+\omega(C)$, with $C = \bar{K}$ (2.2).

Combining the results of §§3–4, we obtain the following in terms of $K$ rather than $\bar{K}$. (Let $\hat{L}$ and $\hat{D}$ be the computed factors of $K$, and note that $\kappa_2(\bar{K}) = \kappa_2(K) = \kappa_2(PKP^T)$.)

ASSUMPTION 5.1. *For some scalar $\gamma$ of moderate size,*

$$\| |\hat{L}|\,|\hat{D}|\,|\hat{L}^T| \|_F \le \gamma \| |L|\,|D|\,|L^T| \|_F.$$

THEOREM 5.1. *If $K$ is symmetric quasidefinite as in* (2.1), *and if $\hat{d}$ is the computed solution of $Kd = r$,*

$$\frac{\|d - \hat{d}\|}{\|d\|} \le \mathbf{u}\gamma n_K c_K \phi(K), \tag{5.3}$$

*where $n_K$ is the dimension of $K$, $c_K$ depends linearly on $n_K$, $\omega(K)$ is defined in* (4.1), *and*

$$\phi(K) \equiv (1+\omega(K))\,\kappa_2(K). \tag{5.4}$$

A similar result holds with $\omega(K)$ replaced by $\theta(K)$ in (4.2). For the example in (1.3)–(1.4), the condition number is $\phi(K) \approx 1/|\epsilon|$, as we might expect.

Under Assumption 5.1, then, arbitrary symmetric permutations of $Kd = r$ (such as those reducing fill-in) can be solved stably without further permutations as long as $\phi(K)$ is not too large. We therefore interpret $\phi(K)$ to be the condition number of Cholesky factorization without interchanges, applied to an sqd system. In algorithms where sequences of sqd systems are solved, techniques that either reduce $\phi(K)$ or delay its increase will, by postponing the need for stability permutations and hence allowing the unhampered use of sparsity permutations, decrease the total computation time for solving $Kd = r$.

Note that the reduction of $\phi(K)$ is sufficient, but not necessary, for ensuring the accurate solution of $Kd = r$ without interchanging rows and columns for stability. Indeed, Golub and Van Loan [GV79] exhibit a family of unsymmetric positive-definite systems $C$ for which $\omega(C)$ increases without bound but whose computed solutions remain accurate without the use of stability interchanges. Their example suggests that in special cases it may be possible to refine the above results to obtain a sharper bound.

**6. An application in numerical optimization.** The standard linear programming (LP) problem is

$$\begin{array}{ll} \underset{x}{\text{minimize}} & c^T x \\ \text{subject to} & Ax = b, \quad l \le x \le u, \end{array} \tag{6.1}$$

where $A \in \mathbb{R}^{m\times n}$ $(m \le n)$. Barrier methods for computing primal and dual solutions $(x, \pi)$ generate a series of sparse symmetric systems; for example, see [LMS92]. Most authors reduce these to the positive-definite form $AH^{-1}A^T\Delta\pi = v$, for which Cholesky factorization is often efficient, as long as $A$ contains no dense columns. We discuss some alternatives.

**6.1. Regularized LP.** In [GMPS91], [GMPS94], we treat the *regularized* LP problem

$$\begin{array}{ll} \underset{x,\,p}{\text{minimize}} & c^Tx + \frac{1}{2}\|\gamma x\|^2 + \frac{1}{2}\|p\|^2 \\ \text{subject to} & Ax + \delta p = b, \quad l \le x \le u, \end{array} \tag{6.2}$$

where $\gamma$ and $\delta$ are small scalar parameters, typically $10^{-5}$. When the optimal $(x, \pi)$ is not unique, choosing a positive $\gamma$ and $\delta$ (respectively) aids convergence to a solution with minimum $\|x\|$ and $\|\pi\|$. If the constraints $Ax = b$, $l \le x \le u$ have no solution, a positive $\delta$ also permits convergence to a meaningful point.

The systems to be solved are

$$K \begin{pmatrix} \Delta x \\ -\Delta\pi \end{pmatrix} = \begin{pmatrix} w \\ r \end{pmatrix}, \qquad K \equiv \begin{pmatrix} H_0 + \gamma^2 I & A^T \\ A & -\delta^2 I \end{pmatrix}, \tag{6.3}$$

where $H_0$ is diagonal with $(H_0)_{jj} \ge 0$. Choosing $\gamma > 0$ and $\delta > 0$ ensures that $K$ is sqd (though barely!). This was not the original motivation, but in view of Vanderbei's work it raises the question: under what conditions is $PKP^T = LDL^T$ stable for any permutation $P$ (with $D$ diagonal but indefinite)?

In the notation of §4, we have $H = H_0 + \gamma^2 I$ and $G = \delta^2 I$. It is safe to assume that $\|A\| \approx 1$ after the LP problem is suitably scaled. As iterations proceed, some elements of $H_0$ become large and cause $\|K\|$ and $\kappa_2(K)$ to appear large. We eliminate this artificial ill-conditioning by symmetrically scaling the large diagonals of $K$ down to 1. System (6.3) is then equivalent to an sqd system $Kd = r$ in which

$$\|K\| \approx 1, \quad \|A\| \approx 1, \quad \|H\| = 1, \quad \|H^{-1}\| \approx \gamma^{-2}, \quad \|G\| = \delta^2, \quad \|G^{-1}\| = \delta^{-2},$$

with the 2-norm used throughout. The scaling does not alter $AH^{-1}A^T$. Result 4.1 then gives

$$\begin{aligned} \omega(K) &\approx \max\{\delta^{-2}\|A^TA\|, \|AH^{-1}A^T\|\} \\ &\le \|A\|^2 \max\{\delta^{-2}, \|H^{-1}\|\} \\ &\approx \max\{\delta^{-2}, \gamma^{-2}\}. \end{aligned}$$

Recalling Theorem 5.1, we have the following.

RESULT 6.1. *Using $PKP^T = LDL^T$, the effective condition number for solving the sqd system* (6.3) *with small $\gamma$ and $\delta$ is $\phi(K) = \max\{\delta^{-2}, \gamma^{-2}\}\kappa_2(K)$.*

On a typical LP problem, the barrier algorithm generates 20 to 30 $K$'s that are increasingly ill-conditioned (even after the large diagonals are scaled to 1). With reasonable values of $\gamma$ and $\delta$, we can expect $PKP^T = LDL^T$ to be stable until the iterates are close to an optimal solution.

**6.2. Numerical experiments.** To confirm this prediction, we applied our barrier code PDQ1 [GMPS91] to some of the more difficult problems in the Netlib collection [Gay85]. Table 6.1 defines some terms and Table 6.2 lists the problem statistics. We requested 6 digits of accuracy in $x$ and $\pi$ on a DEC Alpha 3000/400 workstation with about 16 digits of precision. For regularization we set $\gamma = \delta$ in the range $10^{-3}$ to $10^{-5}$. (Larger values perturb the problem noticeably, while smaller values leave little room for the $LDL^T$ factorization to be stable.)

In PDQ1 Version 1.0, the indefinite solver MA27 [DR82], [DR83] is used to factorize either $K$ itself, or certain reduced matrices $K_B$ (obtained by pivoting on diagonals of

TABLE 6.1
*Definitions associated with the barrier code PDQ1 for solving linear programs.*

| | |
|---|---|
| $K$ | Full KKT system as in (6.3) |
| $K_B$ | Reduced KKT system after pivoting on part of $H$ |
| $nz(K)$ | Number of nonzeros in $K$ |
| PDQ1 | Code for solving sparse LP and QP problems [GMPS91], [GMPS94] |
| MA27 | Code for solving sparse symmetric $Kd = r$ [DR82], [DR83] |
| $LDL^T$ | Sparse factors of permuted $K$ or $K_B$ with $D$ diagonal |
| $LBL^T$ | Sparse factors of permuted $K$ or $K_B$ with $B$ block-diagonal |
| *Htol* | PDQ1's stability tolerance for pivots on $H$ (default = $10^{-6}$) |
| *factol* | MA27's stability tolerance "$u$" (default = 0.01) |
| *ndense* | Nonzeros in a "dense" column of $A$ (default = 10) |
| *residual* | $\|r - K\hat{d}\|/\|r\|$, where $\hat{d}$ is the computed $d$ |
| *restol* | Tolerance for invoking iterative refinement (default = $10^{-5}$) |

TABLE 6.2
*LP test problems: Approximate dimensions of the constraint matrix A, the full KKT matrix K, and a typical reduced KKT matrix $K_B$.*

| | $m$ | $n$ | $nz(A)$ | Size of $K$ | Typical $K_B$ |
|---|---|---|---|---|---|
| *grow22* | 450 | 950 | 6000 | 1400 | 900 |
| *25fv47* | 800 | 1900 | 11000 | 2700 | 1100 |
| *pilotja* | 900 | 2000 | 15000 | 2900 | 1300 |

$H$ that are larger than *Htol* and have fewer than *ndense* entries in the corresponding column of $A$).

The Analyze phase of MA27 typically predicts very sparse $LDL^T$ factors, but to retain stability on indefinite systems, the Factor phase forms $LBL^T$ factors if necessary. These factors grow increasingly dense as the iterations proceed (more so than the combined Analyze/Factor approach used by Fourer and Mehrotra [FM93]).

Stability is measured by testing residuals after the factors of $K$ are used to solve $Kd = r$. If *residual* > *restol*, one step of iterative refinement is performed to correct $\hat{d}$. (The effects of refinement with an unstable factorization are analyzed in [ADD89].) If *residual* still exceeds *restol*, the factors are considered unreliable and *factol* is increased in stages towards 1. In the experiments cited here, once the $LDL^T$ factors were abandoned, the remaining $LBL^T$ solves were performed reliably with *factol* = 0.01.

**6.3. Factorizing $K$.** We first caused the full $K$ to be used every iteration (*Htol* = $10^{20}$). With the default stability tolerance (*factol* = 0.01), MA27 computed $LBL^T$ factors at all iterations except the first few. Iterative refinement was seldom needed, but the factors were two to four times as dense as Analyze predicted. On problem *grow22*, $nz(LBL^T)$ increased steadily from 20000 to 80000 over 18 iterations, giving a relatively long runtime.

With *factol* = 0.001 (a little more dangerous), the $LBL^T$ solves were again reliable, and the factors somewhat more sparse. The values of $\gamma$ and $\delta$ had little effect on the sparsity of the factors.

We then allowed MA27 to compute $LDL^T$ factors as long as possible (*factol* = $10^{-20}$). Table 6.3 shows the number of iterations for which the Cholesky solves were reliable, for various values of $\gamma$ and $\delta$. Times are in cpu seconds. With the larger regularizations, most Cholesky factorizations were stable and efficient. On problem *grow22*, $nz(LDL^T)$ was 20000. With regularizations $10^{-5}$, $10^{-4}$, $10^{-3}$, refinement was first requested at iterations 15, 16, 17, and first failed at iterations 16, 17, 17.

TABLE 6.3

*Performance of PDQ1 with various regularizations $(\gamma, \delta)$, factorizing full KKT systems. The column labeled $LDL^T$ shows how many iterations were performed reliably with (indefinite) Cholesky factors of $K$. The remaining iterations used $LBL^T$ factors, which become increasingly dense.*

| | $\gamma, \delta$ | *factol* | Analyze | $LDL^T$ | $LBL^T$ | time |
|---|---|---|---|---|---|---|
| *grow22* | $10^{-5}$ | 0.01 | 1 | 0 | 18 | 13.4 |
| | $10^{-5}$ | $10^{-20}$ | 1 | 15 | 3 | 6.7 |
| | $10^{-4}$ | $10^{-20}$ | 1 | 16 | 2 | 5.6 |
| | $10^{-3}$ | $10^{-20}$ | 1 | 16 | 2 | 5.5 |
| *25fv47* | $10^{-5}$ | 0.01 | 1 | 0 | 23 | 23.0 |
| | $10^{-5}$ | $10^{-20}$ | 1 | 5 | 18 | 24.2 |
| | $10^{-4}$ | $10^{-20}$ | 1 | 17 | 6 | 20.4 |
| | $10^{-3}$ | $10^{-20}$ | 1 | 21 | 2 | 16.8 |
| *pilotja* | $10^{-5}$ | 0.01 | 1 | 0 | 27 | 37.3 |
| | $10^{-5}$ | $10^{-20}$ | 1 | 5 | 22 | 38.6 |
| | $10^{-4}$ | $10^{-20}$ | 1 | 18 | 9 | 33.5 |
| | $10^{-3}$ | $10^{-20}$ | 1 | 23 | 3 | 26.6 |

For the last two or three iterations, $nz(LBL^T)$ jumped to 80000.

In general, iterative refinement saved several Cholesky factorizations before a switch was made to $LBL^T$. The larger the regularization, the later the need for refinement (and the later the switch to $LBL^T$). The best performance was obtained with the largest regularization, $10^{-3}$.

Some sensitivity was noted regarding the test for refinement. Earlier experience with PDQ1 on the first 70 Netlib problems suggested using *restol* $= 10^{-4}$, but the present experiments with Cholesky factors revealed an occasional increase in total iterations, indicating some unnoticed instability. With *restol* $= 10^{-5}$, the results here err on the side of "fewer iterations at the expense of earlier refinement, and hence possible earlier switch to $LBL^T$ factors." Perhaps the tests in [ADD89] would increase the number of iterations for which Cholesky factors could be safely used.

**6.4. Reduced KKT systems.** We next followed the original PDQ1 strategy of pivoting on most of the diagonals of $H$ (*Htol* $= 10^{-6}$, *ndense* $= 10$). Partitioning $H = \text{diag}(H_N, H_B)$, $A = (N \;\; B)$ and pivoting on $H_N$ gives a reduced matrix of the form

$$K_B = \begin{pmatrix} H_B & B^T \\ B & -NH_N^{-1}N^T - \delta^2 I \end{pmatrix}. \tag{6.4}$$

The aim is to help the Factor phase of MA27, since $K_B$ is smaller and "less indefinite" than $K$. A penalty is that a new Analyze is needed whenever the makeup of $K_B$ changes.

Note that Result 6.1 still applies, since we still have a Cholesky factorization of the full $K$, permuted by a different $P$. Table 6.4 therefore shows qualitatively similar results. The best performance was obtained with $\gamma = \delta = 10^{-3}$ as before, because Analyze was needed only once, and most iterations survived with $LDL^T$ factors.

**6.5. Fully reduced systems.** Table 6.5 gives results when $K$ was fully reduced to $-(AH^{-1}A^T + \delta^2 I)$ via *Htol* $= 10^{-20}$, *factol* $= 0.0$, *ndense* $= 100$. We write this matrix as $AH^{-1}A^T$ for short. It is the one used in most barrier implementations, such as OB1 [LMS92]. A single Analyze is sufficient for the Cholesky factorizations.

TABLE 6.4

*Performance of PDQ1 with various regularizations, factorizing reduced KKT systems $K_B$. Htol is $10^{-20}$ initially, but is increased to $10^{-6}$ after Cholesky factors become unstable. A new Analyze is then needed each time the size of $K_B$ changes. Best results are obtained with maximum regularization ($\gamma = \delta = 10^{-3}$) because the size of $K_B$ depends only on ndense; a single Analyze suffices.*

| | $\gamma, \delta$ | *factol* | Analyze | $LDL^T$ | $LBL^T$ | time |
|---|---|---|---|---|---|---|
| *grow22* | $10^{-5}$ | 0.01 | 11 | 0 | 18 | 19.6 |
| | $10^{-5}$ | $10^{-20}$ | 5 | 14 | 4 | 11.2 |
| | $10^{-4}$ | $10^{-20}$ | 4 | 15 | 3 | 8.6 |
| | $10^{-3}$ | $10^{-20}$ | 1 | 18 | 1 | 5.7 |
| *25fv47* | $10^{-5}$ | 0.01 | 14 | 0 | 23 | 18.9 |
| | $10^{-5}$ | $10^{-20}$ | 4 | 20 | 3 | 15.3 |
| | $10^{-4}$ | $10^{-20}$ | 3 | 21 | 2 | 14.2 |
| | $10^{-3}$ | $10^{-20}$ | 1 | 21 | 2 | 12.5 |
| *pilotja* | $10^{-5}$ | 0.01 | 15 | 0 | 27 | 38.0 |
| | $10^{-5}$ | $10^{-20}$ | 15 | 5 | 22 | 38.7 |
| | $10^{-4}$ | $10^{-20}$ | 3 | 25 | 2 | 25.7 |
| | $10^{-3}$ | $10^{-20}$ | 1 | 25 | 1 | 20.9 |

TABLE 6.5

*Performance of PDQ1, factorizing $AH^{-1}A^T$. This is often the most effective method, but $AH^{-1}A^T$ must be formed efficiently. Not applicable if A contains dense columns.*

| | $\gamma, \delta$ | Analyze | $LDL^T$ | time |
|---|---|---|---|---|
| *grow22* | $10^{-3}$ | 1 | 17 | 5.5 |
| *25fv47* | $10^{-3}$ | 1 | 23 | 12.4 |
| *pilotja* | $10^{-3}$ | 1 | 27 | 29.6 |

Regularization is essential, given the way "free variables" are handled. (If $x_j$ has infinite bounds, $(H_0)_{jj} = 0$. Problem *pilotja* has 88 free variables.) We used $\gamma = \delta = 10^{-3}$ to match the best results in the other tables.

Somewhat surprisingly, $AH^{-1}A^T$ was not a clear winner. Since $A$ had no dense columns in these examples, the Cholesky factors of $AH^{-1}A^T$ were more sparse than the $LDL^T$ or $LBL^T$ factors in Tables 6.3 and 6.4, yet the factorization times were slightly greater. A possible explanation is that the off-diagonals of $AH^{-1}A^T$ are formed as a long list of entries from the sparse rank-one matrices $(1/H_{jj})a_j a_j^T$, which MA27 must accumulate before commencing the factorization. (The same accumulation is used for partially reduced KKT systems, but to a lesser degree.)

**6.6. Use of MA47.** We have recently implemented PDQ1 Version 2.0, in which MA27 is replaced by the new indefinite solver MA47 [DGR91], [DR94]. Following [FM93], we have also experimented with looser pivot tolerances in both codes to improve the sparsity of the numerical factors. In particular, we have initialized *factol* at $10^{-8}$ (increasing it by a factor of 10 whenever refinement fails), and we have run a larger set of test problems.

With MA27, we do obtain significantly improved performance, though iterative refinement and tolerance increases are frequently needed as before. In some cases, *factol* reaches 0.01 or even 0.1.

With MA47, we have found unexpectedly that refinement is *almost never needed*. Reduced KKT systems again give the best performance ($Htol = 10^{-8}$), and milder regularization seems adequate ($\gamma = \delta = 10^{-4}$). The first 53 Netlib problems solved to 8 digits of accuracy with a total of only three refinements, two of which caused *Htol*

and *factol* to be raised to $10^{-7}$. With tolerances of this nature, most factorizations are simply $LDL^T$ with the Analyze ordering. Any $LDL^T$ or $LBL^T$ factorizations with revised orderings are almost equally sparse. The ability to *do* the reordering provides stability at negligible cost.

It appears that two features are contributing to MA47's performance: new stability tests [DGR91], and the default strategy of amalgamating tree nodes to reduce indirect addressing. (By themselves, MA27 *with* amalgamation and MA47 *without* amalgamation were not equally successful.) We hope to give fuller results elsewhere.

**7. Conclusions.** Diverse techniques have been combined here to obtain some new theoretical and practical results. In the context of barrier methods for linear programming, full KKT matrices $K$ are known to have advantages over $AH^{-1}A^T$ in the presence of dense columns and free variables. In [GMPS91] we attempted to improve the performance of MA27's $LBL^T$ factorizations on severely indefinite systems, but with limited success. Regularization was included there for "numerical analysis" reasons, ensuring uniqueness and boundedness of solutions.

Around the same time, Vanderbei introduced quasidefinite systems and exploited the efficiency of $LDL^T$ factors on KKT-like matrices. Recognizing that regularized KKT systems are quasidefinite, and that a closely related system is positive definite, we were led to the results of Golub and Van Loan on $LU$ factorization without interchanges. From these, we established an effective condition number $\phi(K)$ (5.4) for Cholesky factorization of sqd systems. Result 6.1 justifies $LDL^T$ factorization of sqd matrices $K$ for the special case of barrier methods for linear programming.

Note that our analysis does *not* explain the remarkable success that Vanderbei has had with his $LDL^T$ factors of sqd systems. In particular, Vanderbei does not resort to regularization. Instead, some innovative problem formulation and partitioning gives a multilevel ordering scheme in which certain diagonal pivots are deferred (notably zeros). An sqd principal submatrix is chosen and factored as $LDL^T$. The Schur complement then has an sqd principal submatrix, and so on. We hope that a direct analysis will eventuate.

Meanwhile, the numerical results obtained here suggest the following approach to systems $Kd = r$ of the form (6.3): Choose the regularizing parameters $\gamma$, $\delta$ reasonably large (e.g., $10^{-3}$ or $10^{-4}$) and pivot on all entries of $H$ for which the column of $A$ is not too dense. A single Analyze will then suffice, and $LDL^T$ factorization should be efficient and reliable until a good estimate of the solution is reached.

For higher accuracy, we must not forget that implementations based on $AH^{-1}A^T$ are surprisingly reliable and efficient on most real-world problems [Lus94]. Otherwise, Vanderbei's indefinite Cholesky approach is an answer to dense columns and free variables, as are the $LBL^T$ factors in [FM93], [GMPS91], with MA47 now providing a very welcome boost.

**Acknowledgments.** We are grateful to Nicholas Higham and a second referee for many helpful suggestions.

## REFERENCES

[ADD89] M. Arioli, J. W. Demmel, and I. S. Duff, *Solving sparse linear systems with sparse backward error*, SIAM J. Matrix Anal. Appl., 10 (1989), pp. 165–190.

[DGR91] 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.

[DR82] I. S. DUFF AND J. K. REID, MA27: *A set of Fortran subroutines for solving sparse symmetric sets of linear equations*, Report R-10533, Computer Science and Systems Division, AERE Harwell, Oxford, England, 1982.

[DR83] ———, *The multifrontal solution of indefinite sparse symmetric linear equations*, ACM Trans. Math. Software, 9 (1983), pp. 302–325.

[DR94] ———, MA47: *A Fortran code for direct solution of indefinite sparse symmetric linear systems*, 1994, manuscript.

[FM93] R. FOURER AND S. MEHROTRA, *Solving symmetric indefinite systems in an interior-point method for linear programming*, Math. Prog., 62 (1993), pp. 15–39.

[Gay85] D. M. GAY, *Electronic mail distribution of linear programming test problems*, Math. Programming Society COAL Newsletter, December 1985.

[GMPS91] P. E. GILL, W. MURRAY, D. B. PONCELEÓN, AND M. A. SAUNDERS, *Solving reduced* KKT *systems in barrier methods for linear and quadratic programming*, Report SOL 91-7, Department of Operations Research, Stanford University, Stanford, CA, 1991.

[GMPS94] ———, *Solving reduced* KKT *systems in barrier methods for linear programming*, in Numerical Analysis 1993, G. A. Watson and D. Griffiths, eds., Pitman Research Notes in Mathematics 303, Longmans Press, 1994, pp. 89–104.

[GV79] G. H. GOLUB AND C. F. VAN LOAN, *Unsymmetric positive definite linear systems*, Linear Algebra Appl., 28 (1979), pp. 85–98.

[GV89] ———, *Matrix Computations*, second edition, The Johns Hopkins University Press, Baltimore, 1989.

[Lus94] I. J. LUSTIG, *Comments on the performance of the* CPLEX *barrier algorithm*, private communication, 1994.

[LMS92] I. J. LUSTIG, R. E. MARSTEN, AND D. F. SHANNO, *On implementing Mehrotra's predictor-corrector interior point method for linear programming*, SIAM J. Optim., 2 (1992), pp. 435–449.

[Van91] R. J. VANDERBEI, *Symmetric quasi-definite matrices*, Report SOR 91-10, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ, 1991.

[Van94] ———, *Symmetric quasi-definite matrices*, SIAM J. Optim., 5 (1995), pp. 100–113.