STABLE REDUCTION TO KKT SYSTEMS IN BARRIER METHODS FOR LINEAR AND QUADRATIC PROGRAMMING*

MICHAEL A. SAUNDERS† AND JOHN A. TOMLIN‡

Abstract. We discuss methods for solving the key linear equations within primal-dual barrier methods for linear and quadratic programming. Following Freund and Jarre, we explore methods for reducing the Newton equations to 2×2 block systems (KKT systems) in a stable manner. Some methods require partitioning the variables into two or more parts, but a simpler approach is derived and recommended.

To justify symmetrizing the KKT systems, we assume the use of a sparse solver whose numerical properties are independent of row and column scaling. In particular, we regularize the problem and use indefinite Cholesky-type factorizations. An implementation within OSL is tested on the larger NETLIB examples.

Key words. interior methods, barrier methods, linear programming, quadratic programming, KKT systems, Cholesky factors, regularization, stability

AMS subject classifications. 90C05, 90C06, 90C20, 65F05, 65F50

1. Introduction. We consider primal-dual interior methods (barrier methods) for solving sparse linear programs and convex quadratic programs of the form

$$\underset{x}{\text{minimize}} \;\; c^T x + \tfrac{1}{2} x^T Q x \quad \text{subject to} \quad Ax = b, \quad l \le x \le u.$$

Most of the computational work lies in solving large systems of linear equations involving six sets of primal and dual variables (the full Newton system). We focus on techniques for making these solves both stable and efficient.

For problems involving lower bounds $x \ge 0$ and no upper bounds, the Newton system involves only three sets of variables Δx, Δy, Δz. Most authors eliminate Δz and/or Δx without further ado. Freund and Jarre [FJ95] first showed how the 3×3-block Newton system can be reduced to a 2×2 system in a *numerically stable way*. The reduction is viewed as Gaussian elimination with column interchanges, and involves partitioning Δx and Δz into two parts. The resulting system involves an unsymmetric matrix J, but it may be transformed by column scaling into a *KKT system* involving a symmetric matrix K.

We generalize Freund and Jarre's approach to handle upper and lower bounds, and derive several other stable ways of obtaining J and K. Ultimately, we find a method that does not require partitioning. To a large extent it justifies the naive elimination that has been in common use.

In order to make use of sparse Cholesky packages, we regularize the problem via moderate perturbations, as in Gill *et al.* [GMPS94, GSS96] and Saunders [Sau96]. This leads to KKT matrices of the form

$$K = \begin{pmatrix} -H & A^T \\ A & \delta^2 I \end{pmatrix},$$

* February 16, 2000. IBM Almaden Research Report RJ 10039 and Stanford Report SOL 96-3. Presented at the International Symposium on Optimization and Computation, Hayama, Kanagawa, Japan, August 12–16, 1996. This research was partially supported by Office of Naval Research grants N00014-94-C-007 and N00014-96-1-0274 and by National Science Foundation grant DMI-9500668.

† Department of EES & Operations Research, Stanford University, Stanford, CA 94305-4023 (mike@SOL-michael.stanford.edu).

‡ IBM Almaden Research Center, San Jose, CA 95120-6099 (tomlin@almaden.ibm.com).

where H is positive definite, and it allows Cholesky-type LDL^T factorizations of K (where D is diagonal but indefinite), as first advocated by Vanderbei [Van95]. Regularization provides a theoretical and practical guarantee of stability, and helps give sensible solutions in degenerate cases.

The present work complements that of Wright, who gives error analyses for solving the *sequence* of KKT systems [Wri96] or normal equations [Wri95] generated by a typical primal-dual method. By assuming that the iterates remain sufficiently interior and sufficiently close to the central path, Wright provides a global view of the accuracy attainable in each search direction and the final estimate of (x, y), even without the help of regularization.

Our aim here is more simple. As in [FJ95], we wish to take the full Newton system at any iteration and reduce it to a KKT system without introducing unnecessary error, independent of assumptions about the central path. (Regularization helps ensure that the Newton systems are not excessively ill-conditioned, but of course the total number of iterations depends on other things.)

2. Regularized QP. We consider regularized linear and quadratic programs of the form

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

where $Q = Q_0 + \gamma^2 I$ and Q_0 is positive semi-definite. The scalars γ and δ are typically "small", in the range 10^{-2} to 10^{-6}. In our numerical experiments, $Q_0 = 0$ and $Q = \gamma^2 I$. As usual, A and x contain entries associated with slack variables on inequality constraints. Slacks on equality rows are excluded. (The perturbation effect of regularization is studied more closely in [ST96].)

ASSUMPTION 1. *The problem is scaled so that* $\|A\| \approx 1$ *and* $\|c\| \approx 1$. (This gives meaning to specific values of γ and δ such as 10^{-4}. Ideally the optimal solution and dual vector should also satisfy $\|x\| \approx 1$ and $\|y\| \approx 1$.)

ASSUMPTION 2. *The upper and lower bounds* u *and* l *contain* $+\infty$ *or* $-\infty$ *entries where necessary, but finite bounds are within a few orders of magnitude of the optimum components of* x. (Ideally there are no "ambiguous" entries like $\pm 10^{10}$. Hence the diagonal matrices S and T below have no excessively large entries.)

ASSUMPTION 3. *The KKT systems are solved by a method whose numerical errors are independent of the row or column scaling of* K. (This allows us to symmetrize K.)

In Assumption 3 we have in mind that when triangular factorization (and forward and backward substitution) are applied to a square system $Kx = b$ with a specified pivot order, the relative errors in the computed solution components x_j are the same for any scaling by powers of 2 (for most machines), and essentially the same for arbitrary scaling as long as overflow or underflow do not occur.

Assumption 3 would be satisfied if K were reduced to $AH^{-1}A^T + \delta^2 I$ as usual and a sparse Cholesky package were applied. More generally, by setting γ and δ sufficiently large, we can use a black-box Cholesky package to form $PKP^T = LDL^T$ for any ordering P, as follows from [GV79, GSS96, Sau96].

2.1. The Newton equations. Following Megiddo [Meg89], Mehrotra [Meh90], Lustig *et al.* [LMS92], Forrest and Tomlin [FT92], Kojima *et al.* [KMM93] and others, we apply an infeasible primal-dual predictor-corrector algorithm to problem (1). The

nonlinear equations defining the central trajectory are $p = \delta y$ and

$$
\begin{aligned}
x - s &= l, \\
x + t &= u, \\
SZe &= \mu e, \qquad && \mu > 0, \\
TWe &= \mu e, && s, t, z, w > 0, \\
A^T y + z - w &= c + Qx, \\
Ax + \delta^2 y &= b,
\end{aligned} \tag{2}
$$

where e is a vector of ones, $S = \mathrm{diag}(s_j)$, and similarly for T, Z, W. (If l and u contain infinite entries, the corresponding equations are omitted.)

The primal-dual algorithm uses Newton's method to generate search directions from equations of the form

$$
\begin{aligned}
\Delta x - \Delta s &= \hat{u} &&= (l + s) - x, \\
\Delta x + \Delta t &= \hat{v} &&= (u - t) - x \\
S\Delta z + Z\Delta s &= g &&= \mu e - Sz, \\
T\Delta w + W\Delta t &= h &&= \mu e - Tw, \\
-Q\Delta x + A^T\Delta y + \Delta z - \Delta w &= d &&= c + Qx - A^T y - z + w, \\
A\Delta x + \delta^2\Delta y &= r &&= b - Ax - \delta^2 y.
\end{aligned} \tag{3}
$$

From the viewpoint of "Gaussian elimination with column interchanges", it is safe to eliminate Δs and Δt using the first two equations (pivoting on $-I$ and I). This gives

$$
\boxed{J_4\,\Delta_4 = r_4 \qquad \equiv \qquad \begin{pmatrix} S & & Z & \\ & T & -W & \\ I & -I & -Q & A^T \\ & & A & \delta^2 I \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta w \\ \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \hat{g} \\ \hat{h} \\ d \\ r \end{pmatrix}}, \tag{4}
$$

where

$$
\begin{aligned}
\hat{g} &= g + Z\hat{u}, \qquad & \Delta s &= \Delta x - \hat{u}, \\
\hat{h} &= h - W\hat{v}, & \Delta t &= \hat{v} - \Delta x.
\end{aligned} \tag{5}
$$

We must solve (4) as stably and efficiently as possible.

3. Simple bounds. When there are no upper bounds, (4) becomes

$$
\boxed{J_3\,\Delta_3 = r_3 \qquad \equiv \qquad \begin{pmatrix} S & Z & \\ I & -Q & A^T \\ & A & \delta^2 I \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \hat{g} \\ d \\ r \end{pmatrix}}. \tag{6}
$$

We examine various ways to eliminate variables.

3.1. Conventional reduction to KKT systems. Pivoting on S gives a symmetric KKT system directly:

$$
\boxed{K\,\Delta_2 = r_2 \qquad \equiv \qquad \begin{pmatrix} -H & A^T \\ A & \delta^2 I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \bar{d} \\ r \end{pmatrix}}, \tag{7}
$$

where

$$(8)\qquad \begin{array}{rcl} H &=& Q + S^{-1}Z, \\ \bar{d} &=& d - S^{-1}\hat{g}, \end{array} \qquad\qquad S\Delta z \;=\; \hat{g} - Z\Delta x.$$

As noted by Vanderbei [Van94] and Freund and Jarre [FJ95], such a reduction is apparently unstable (as the LP solution is approached) because many components of s become very small. Eventually, H has many large diagonals and $\bar{d}$'s components vary greatly in magnitude. Growth of this kind in LU factorization usually leads to significant numerical error.

3.2. Partitions. Here we explore Freund and Jarre's approach of pivoting on just *part* of S (and part of Z). Following [FJ95], we partition s and z into two parts such that $s_1 \geq z_1$ and $s_2 < z_2$. Other quantities are partitioned to match. From the viewpoint of "LU with column interchanges", we may pivot on S_1 and Z_2 in (6) to obtain the systems

$$(9)\qquad \boxed{J12\,\Delta 12 = r12 \qquad \equiv \qquad \begin{pmatrix} -Q_{11} - D_1 & Q_{12}D_2 & A_1^T \\ -Q_{21} & I + Q_{22}D_2 & A_2^T \\ A_1 & -A_2D_2 & \delta^2 I \end{pmatrix} \begin{pmatrix} \Delta x_1 \\ \Delta z_2 \\ \Delta y \end{pmatrix} = r12}$$

and

$$(10)\qquad \begin{array}{rcl} S_1\Delta z_1 &=& \hat{g}_1 - Z_1\Delta x_1, \\ Z_2\Delta x_2 &=& \hat{g}_2 - S_2\Delta z_2, \end{array}$$

where

$$(11)\qquad \begin{array}{rcl} D_1 &=& S_1^{-1}Z_1, \\ D_2 &=& Z_2^{-1}S_2, \end{array} \qquad \begin{array}{rcl} \bar{g}_1 &=& S_1^{-1}\hat{g}_1, \\ \bar{g}_2 &=& Z_2^{-1}\hat{g}_2, \end{array} \qquad r12 \;=\; \begin{pmatrix} d_1 - \bar{g}_1 + Q_{12}\bar{g}_2 \\ d_2 + Q_{22}\bar{g}_2 \\ r - A_2\bar{g}_2 \end{pmatrix}.$$

The matrix $J12$ contains no large numbers. A stable (but unsymmetric) solver on (9), along with (10), would provide a reliable solution to (6). However, we can apply column scaling to obtain the symmetric system

$$(12)\qquad \boxed{K\,\overline{\Delta 12} = r12, \qquad \overline{\Delta 12} = \begin{pmatrix} \Delta x_1 \\ u_2 \\ \Delta y \end{pmatrix} \equiv \begin{pmatrix} \Delta x_1 \\ -D_2\Delta z_2 \\ \Delta y \end{pmatrix}},$$

which should be more efficient to solve. (The matrix K is the same as in the conventional reduction (7).) If Assumption 3 holds, we can say that (12) is equivalent to (9) both *analytically and numerically*. It is then acceptable to recover Δz_2 by unscaling the KKT solution: $\Delta z_2 = -D_2^{-1}u_2$.

The analogue of (12) is derived in [FJ95] (with $Q = 0$), in the context of solving the system *approximately* by an iterative method. It is noted that the right-hand side $r12$ should involve less cancellation error than $\bar{d}$ in the conventional reduction. Symmetrizing K is partially justified (on the grounds that a sensible preconditioner will undo part of the damage). The vector Δz_2 is recovered from (6) as $\Delta z_2 = d_2 - A_2^T\Delta y$ rather than $-D_2^{-1}u_2$. This must lead to different iterates when the solves are approximate, and may be beneficial in general (see §5).

3.3. Stable reduction without partitions. Here we pursue the viewpoint of "LU with row interchanges and threshold pivoting" to obtain KKT systems without the help of partitions.

Reordering the rows of (6) gives

$$\boxed{\widetilde{J3}\,\Delta 3 = \widetilde{r3} \quad \equiv \quad \begin{pmatrix} I & -Q & A^T \\ S & Z & \\ & A & \delta^2 I \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} d \\ \hat{g} \\ r \end{pmatrix}}. \tag{13}$$

By Assumption 2 we can pivot on I with reasonable safety to eliminate S (because $\|S\|$ is not enormously larger than 1), thereby obtaining

$$\boxed{J2\,\Delta 2 = r2 \quad \equiv \quad \begin{pmatrix} (SQ+Z) & -SA^T \\ A & \delta^2 I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \hat{g} - Sd \\ r \end{pmatrix}} \tag{14}$$

and

$$\Delta z = d + Q\Delta x - A^T \Delta y. \tag{15}$$

This time we symmetrize $J2$ by *row* scaling (with $-S^{-1}$). The resulting KKT system proves to be exactly the same as (7). Thus, as long as Assumptions 2 and 3 are satisfied, we see that the conventional reduction to (7) is stable after all. The only difference is that Δz should be obtained from (15) rather than the cheaper system $S\Delta z = \hat{g} - Z\Delta x$ in (8).

4. Double bounds. Returning to the full system $J4\,\Delta 4 = r4$ (4), we examine similar options for eliminating variables to obtain KKT systems.

4.1. Conventional reduction. Pivoting on S and T gives the symmetric KKT system

$$\boxed{K\,\Delta 2 = r2 \quad \equiv \quad \begin{pmatrix} -H & A^T \\ A & \delta^2 I \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \bar{d} \\ r \end{pmatrix}}, \tag{16}$$

where

$$\begin{array}{rclcrcl} H & = & Q + S^{-1}Z + T^{-1}W, & & S\Delta z & = & \hat{g} - Z\Delta x, \\ \bar{d} & = & d - S^{-1}\hat{g} + T^{-1}\hat{h}, & & T\Delta w & = & \hat{h} + W\Delta x. \end{array} \tag{17}$$

As before, the reduction is apparently unstable because many components of s and t become very small.

4.2. Four partitions. The approach of Freund and Jarre can be followed if we partition the variables into four parts according to

$$\begin{array}{lll} s_1 \geq z_1 & \text{and} & t_1 \geq w_1, \\ s_2 < z_2 & \text{and} & t_2 \geq w_2, \\ s_3 \geq z_3 & \text{and} & t_3 < w_3, \\ s_4 < z_4 & \text{and} & t_4 < w_4. \end{array} \tag{18}$$

As in §3.2, "LU factorization with column interchanges" leads with maximum stability to an unsymmetric system and thence to a KKT system via column scaling.

Table 1

Stable factorizations for triangularizing the first two rows of J_4, allowing for the relative sizes of S, Z, T and W. In all cases, $\|D\| \le 1$ and $\|E\| \le 1$. In the last case, $\bar{T} = T + WD$ and $F = D(D+E)^{-1}$.

D	E	$\begin{pmatrix} S & & Z \\ & T & -W \end{pmatrix}$
$S^{-1}Z$	$T^{-1}W$	$\begin{pmatrix} S & & \\ & T & \end{pmatrix} \begin{pmatrix} I & & D \\ & I & -E \\ & & I \end{pmatrix}$
SZ^{-1}	$T^{-1}W$	$\begin{pmatrix} & & Z \\ & T & -W \end{pmatrix} \begin{pmatrix} I & & \\ DE & I & \\ D & & I \end{pmatrix}$
$S^{-1}Z$	TW^{-1}	$\begin{pmatrix} S & & \\ & & -W \end{pmatrix} \begin{pmatrix} I & & D \\ & I & \\ & -E & I \end{pmatrix}$
SZ^{-1}	TW^{-1}	$\begin{pmatrix} & & Z \\ & \bar{T} & -W \end{pmatrix} \begin{pmatrix} I & & \\ F & I & \\ & & I \end{pmatrix} \begin{pmatrix} I & -I & \\ & I & \\ D & & I \end{pmatrix}$

In essence, we need four stable factorizations to triangularize the first two rows of J_4. All cases in (18) can be dealt with in terms of diagonal matrices D_i and E_i satisfying $\|D_i\| \le 1$ and $\|E_i\| \le 1$. For example, the variables in partition 1 satisfy

$$\begin{pmatrix} S_1 & & Z_1 \\ & T_1 & -W_1 \end{pmatrix} = \begin{pmatrix} S_1 & & \\ & T_1 & \end{pmatrix} \begin{pmatrix} I & & D_1 \\ & I & -E_1 \\ & & I \end{pmatrix}.$$

Table 1 shows this factorization and the other three (with subscripts omitted for simplicity). The last factorization is nontrivial, but it is easily verified by multiplying the factors.

We have implemented this four-partition method and obtained reliable performance as expected. We now view it as unnecessarily complex, but it remains the only method that uses "LU with maximal pivoting".

4.3. Two partitions. As in §3.3 we may adopt the viewpoint of "LU with row interchanges and threshold pivoting". Reordering the rows and columns of (4) gives the system

$$\widetilde{J_4}\,\widetilde{\Delta_4} = \widetilde{r_4} \quad \equiv \quad \begin{pmatrix} -I & I & -Q & A^T \\ & S & Z & \\ T & & -W & \\ & & A & \delta^2 I \end{pmatrix} \begin{pmatrix} \Delta w \\ \Delta z \\ \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} d \\ \hat{g} \\ \hat{h} \\ r \end{pmatrix}, \tag{19}$$

and by Assumption 2 we may pivot on $-I$ to eliminate T. Thus,

$$\boxed{\overline{J_3}\Delta_3 = \overline{r_3} \quad \equiv \quad \begin{pmatrix} S & Z & \\ T & -(W+TQ) & TA^T \\ & A & \delta^2 I \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \hat{g} \\ \hat{h} + Td \\ r \end{pmatrix}} \tag{20}$$

and $\Delta w = \Delta z - Q\Delta x + A^T\Delta y - d$ provide a reasonably stable reduction.

We now have two options for applying the [FJ95] approach to $\overline{J_3}$:

- (LU with column interchanges) Partition s and z as for (6) and reduce to a KKT system with the help of column *and* row scaling.
- (LU with row interchanges) Partition s and t and reduce to a KKT system via column and row scaling.

We have implemented the first option successfully, but again we now view it as unnecessarily complex.

4.4. No partitions. The transition from (19) to (20) is stable so far. To create another suitable pivot, a simple device is to add the second row of (20) to the first row:

$$\boxed{\begin{pmatrix} S+T & (Z-W-TQ) & TA^T \\ T & -(W+TQ) & TA^T \\ & A & \delta^2 I \end{pmatrix} \begin{pmatrix} \Delta z \\ \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} \hat{g} + \hat{h} + Td \\ \hat{h} + Td \\ r \end{pmatrix}}. \tag{21}$$

Since S and T have positive entries, we can now pivot on $S+T$ to eliminate T. After some elaborate algebra and appropriate row scaling, we obtain exactly the KKT system (16) that arises in the conventional reduction. The only difference is that we should recover Δz and Δw from

$$\begin{aligned} q &= d + Q\Delta x - A^T\Delta y, \\ (S+T)\Delta z &= \hat{g} + \hat{h} + Tq + (W-Z)\Delta x, \\ \Delta w &= \Delta z - q, \end{aligned} \tag{22}$$

rather than from (17).

5. Numerical results. Our implementation of the above methods is based on OSL, the IBM Optimization Subroutine Library [OSL]. The implementation took place in two phases. The first required modification of the existing primal-dual predictor-corrector barrier code [FT92] and the Cholesky factorization routines to allow for the following items:

- Solving regularized LP problems of the form (1) with $Q = \gamma^2 I$.
- Solving KKT systems (or reduced KKT systems [GMPS94]) rather than the normal equations used in [FT92].
- Computing sparse LDL^T factors rather than LL^T, with D diagonal but indefinite.

Note that the regularization process for LPs can be accommodated while retaining the normal equations approach. However, the KKT formulation and indefinite Cholesky factorization are not only essential for QP problems, but they provide other recognized advantages (such as direct handling of free variables, and better exploitation of sparsity, especially in the presence of dense columns).

As test problems we used a subset of the NETLIB collection [Gay85], namely all problems with more than 1000 equations (with the exception of *stocfor3*, which

is in inconvenient form). The subset includes *80bau3b*, *bnl2*, *cycle*, *d2q06c*, *degen3*, *dfl001*, *fit2p*, *ganges*, *greenbea*, *greenbeb*, *pilots*, *pilot87*, *sctap2*, *sctap3*, *ship12l*, *ship12s*, *stocfor2* and *woodw*. All models were scaled (see Assumption 1) but not presolved, since that makes it difficult to reproduce results in other implementations.

The tests were performed on an IBM RS/6000 workstation with machine precision $\epsilon \approx 10^{-16}$. Experiments with various values of γ and δ are described in a companion paper [ST96]. Here we set $\gamma = \delta = 4.0 \times 10^{-4}$. The "effective condition" of the KKT systems [Sau96] is then $(\|A\|/(\gamma\delta))^2 \approx 0.01/\epsilon$, so that at least 2 digits of accuracy can be expected in each search direction. Iterations were terminated when the relative gap between the (regularized) primal and dual objectives was below 10^{-7}, or the total complementarity $s^T z + t^T w$ (suitably normalized) was below 10^{-6}.

The second implementation phase involved coding the various reduction schemes. As an accuracy check, we evaluated the "relative residuals" for the Newton system (4); i.e., the norm of the residual on each row relative to $\|r_4\|$. The standard method of §4.1 served as a benchmark. In general, the relative residuals increase as the solution is approached, but are never larger than $O(10^{-2})$.

For the four-partition approach of §4.2—the most elaborate alternative—it was initially disconcerting to find that the results were almost identical to those from the standard reduction. The relative residuals for any given iteration were very similar, as were the actual sequences of iterates. This was true even for the traditionally more badly behaved NETLIB models such as *80bau3b*.

The two-partition method of §4.3 (with s and z partitioned) also gave similar results.

Finally, for the no-partitions method of §4.4 we again saw the now customary behavior. However, this is the method we recommend, and we could observe one benefit in terms of the residuals. From (17), we see that the conventional reduction recovers Δz and Δw directly from the first two rows of (4), while Δx and Δy are subject to errors in the KKT solve. Thus, the relative residuals for the first two rows are $O(\epsilon)$, while those for the last two rows are much larger.

On the other hand, when Δz and Δw are recovered via (22), the third equation in (4) is satisfied to $O(\epsilon)$ and the unavoidable error in the KKT solve now manifests itself in the residuals for the first two equations. It seems preferable to us that the search direction should closely satisfy the Newton equation requiring dual feasibility, rather than the complementarity equations, which we know are satisfied only approximately in large-step algorithms (with μ changing markedly at each iteration).

To test the need for Assumption 2, we changed all infinite bounds to $\pm 10^{10}$ and found that the number of barrier iterations increased substantially in many cases. However, there was no significant difference among the various reduction methods. (The increase in computation time is one more reason for avoiding the bad modeling practice of using large bounds for essentially unbounded quantities.)

6. Conclusions. For several years, the Newton equations for primal-dual barrier methods have been reduced to KKT systems or normal equations by "naively" pivoting on diagonal matrices regardless of their numerical values. Vanderbei [Van94] commented on the apparent dangers. Freund and Jarre [FJ95] removed most doubts by applying "LU factorization with column interchanges (and maximal pivoting)".

Here we have explored Freund and Jarre's approach, extending it to LP and QP problems with upper and lower bounds. Under certain assumptions (§2), we derived a simpler method based on "LU with threshold pivoting" and found that the resulting KKT systems are the same as those obtained by the naive approach. Thus,

the conventional method for obtaining the search vectors Δx and Δy is seen to be reasonably stable after all.

The same viewpoint suggests that the other search vectors Δz and Δw should be obtained differently from the naive approach, i.e., from (15) and (22) rather than (8) or (17), although our numerical results do not reveal a significant effect on the overall convergence of the barrier algorithm. (Far greater effects are obtained from presolving, scaling, and avoiding large bounds.) Reliability concerns are largely swept away by the judicious use of regularization.

REFERENCES

[FJ95] R. W. Freund and F. Jarre, *A QMR-based interior-point algorithm for solving linear programs*, Math. Prog., to appear.

[FT92] J. J. H. Forrest and J. A. Tomlin, *Implementing interior point linear programming methods in the Optimization Subroutine Library*, IBM Systems J., 31(1), pp. 26–38, 1992.

[Gay85] D. M. Gay, *Electronic mail distribution of linear programming test problems*, Math. Programming Society COAL Newsletter, 13, pp. 10–12, 1985.

[GMPS94] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. Saunders, *Solving reduced KKT systems in barrier methods for linear programming*, in G. A. Watson and D. Griffiths (eds.), *Numerical Analysis 1993*, Pitman Research Notes in Mathematics 303, Longmans Press, pp. 89–104, 1994.

[GSS96] P. E. Gill, M. A. Saunders and J. R. Shinnerl, *On the stability of Cholesky factorization for quasi-definite systems*, SIAM J. Mat. Anal., 17(1) (1996), pp. 35–46.

[GV79] G. H. Golub and C. F. Van Loan, *Unsymmetric positive definite linear systems*, Linear Alg. and its Appl., 28 (1979), pp. 85–98.

[KMM93] M. Kojima, N. Megiddo and S. Mizuno, *A primal-dual infeasible-interior-point algorithm for linear programming*, Math. Programming 61 (1993), pp. 263–280.

[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.

[Meg89] N. Megiddo, *Pathways to the optimal set in linear programming*, in N. Megiddo (ed.), *Progress in Mathematical Programming, Interior-Point and Related Methods*, Springer, NY (1989), pp. 131–158.

[Meh90] S. Mehrotra, *On the implementation of a (primal-dual) interior point method*, Report 90-03, Dept of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 1990.

[OSL] IBM Corporation, *Optimization Subroutine Library Guide and Reference*, Document SC23-0519-2 (1991), available through IBM branch offices.

[Sau96] M. A. Saunders, *Cholesky-based methods for sparse least squares: The benefits of regularization*, in L. Adams and J. L. Nazareth (eds.), *Linear and Nonlinear Conjugate Gradient-Related Methods*, Proceedings of AMS-IMS-SIAM Joint Summer Research Conference, University of Washington, Seattle, WA (July 9–13, 1995), SIAM, Philadelphia, pp. 92–100, 1996.

[ST96] M. A. Saunders and J. A. Tomlin, *Solving regularized KKT systems in barrier methods for linear programming*, Report SOL 96-4, Dept of EES&OR, Stanford University, Stanford, CA, 1996.

[Van94] R. J. Vanderbei, *LOQO: An interior point code for quadratic programming*, Report SOR-94-15, Dept of Civil Engineering and Operations Research, Princeton University, Princeton, NJ, 1994.

[Van95] ———, *Symmetric quasi-definite matrices*, SIAM J. Optim., 5(1) (1995), pp. 100–113.

[Wri95] S. J. Wright, *Stability of linear equations solvers in interior-point methods*, SIAM J. Mat. Anal. Appls., 16 (1995), pp. 1287–1307.

[Wri96] ———, *Stability of augmented system factorizations in interior-point methods*, SIAM J. Mat. Anal. Appls., to appear.