In L. Adams and J. L. Nazareth (eds.), *Linear and Nonlinear Conjugate Gradient-Related Methods*, SIAM, Philadelphia, 92–100 (1996).

Chapter 8
Cholesky-based Methods for Sparse Least Squares: The Benefits of Regularization*

Michael A. Saunders†

Abstract

We study the use of black-box LDL^T factorizations for solving the augmented systems (KKT systems) associated with least-squares problems and barrier methods for linear programming (LP). With judicious regularization parameters, stability can be achieved for *arbitrary data* and *arbitrary permutations* of the KKT matrix.

This offers improved efficiency compared to implementations based on "pure normal equations" or "pure KKT systems". In particular, the LP matrix may be partitioned arbitrarily as $(\,A_s \;\; A_d\,)$. If $A_s A_s^T$ is unusually sparse, the associated "reduced KKT system" may have very sparse Cholesky factors. Similarly for least-squares problems if a large number of rows of the observation matrix have special structure.

Numerical behavior is illustrated on the villainous Netlib models *greenbea* and *pilots*.

1 Background

The connection between this work and Conjugate-Gradient methods lies in some properties of two CG algorithms, LSQR and CRAIG, for solving linear equations and least-squares problems of various forms. We consider the following problems:

(1)	*Linear equations*:	$Ax = b$
(2)	*Minimum length*:	$\min \|x\|^2$ subject to $Ax = b$
(3)	*Least squares*:	$\min \|Ax - b\|^2$
(4)	*Regularized least squares*:	$\min \|Ax - b\|^2 + \|\delta x\|^2$
(5)	*Regularized min length*:	$\min \|x\|^2 + \|s\|^2$ subject to $Ax + \delta s = b$

where A is a general matrix (square or rectangular) and δ is a scalar ($\delta \geq 0$).

LSQR [17, 18] solves the first four problems, and incidentally the fifth, using essentially the same work and storage per iteration in all cases. The iterates x_k reduce $\|b - Ax_k\|$ monotonically.

CRAIG [4, 17] solves only compatible systems (1)–(2), with $\|x - x_k\|$ decreasing monotonically. Since CRAIG is slightly simpler and more economical than LSQR, it may sometimes be preferred for those problems.

*Partially supported by Department of Energy grant DE-FG03-92ER25117, National Science Foundation grant DMI-9204208, and Office of Naval Research grant N00014-90-J-1242.

†Systems Optimization Laboratory, Dept of Operations Research, Stanford University, Stanford, California 94305-4022 (mike@SOL-michael.stanford.edu).

To extend CRAIG to incompatible systems, we have studied problem (5): a compatible system in the combined variables (x, s). If $\delta > 0$, it is readily confirmed that problems (4) and (5) have the same solution x, and that both are solved by either the normal equations

$$Nx = A^T b, \qquad N \equiv A^T A + \delta^2 I, \tag{6}$$

or the augmented system

$$K \begin{pmatrix} s \\ x \end{pmatrix} = \begin{pmatrix} b \\ 0 \end{pmatrix}, \qquad K \equiv \begin{pmatrix} \delta I & A \\ A^T & -\delta I \end{pmatrix}. \tag{7}$$

The special form of CRAIG developed in [19] does not appear to have advantages over LSQR. However, a side-effect of that research has been to focus attention on system (7). Our aim in the remainder of the paper is to study *direct* methods for solving (6) and (7), with an emphasis on both stability and efficiency.

Some recent references exploring stability matters are Forsgren [7], Gill *et al.* [12], Vavasis [22] and Wright [23].

1.1 Notation

The following terms are used:

sqd	Symmetric quasi-definite, as in (8)
Cholesky factors	$PKP^T = LDL^T$, P a permutation, L lower triangular, D diagonal (possibly indefinite)
$\sigma_{\max}(A) = \|A\|$	Largest singular value of general matrix A
$\lambda_{\max}(K) = \|K\|$	Largest eigenvalue of symmetric matrix K
$\mathrm{Cond}(A) = \sigma_{\max}(A)/\sigma_{\min}(A)$	Condition number of general matrix A
$\mathrm{Cond}(K) = \lambda_{\max}(K)/\lambda_{\min}(K)$	Condition number of symmetric matrix K
$\mathrm{Econd}(K)$	Effective condition number of K when solving with unstable factors
ϵ	Floating-point precision (typically $\approx 10^{-16}$)

2 The Condition of N and K

The normal equations (6) are effective when N is sparse and reasonably conditioned. Since N is positive definite, it is well known that Cholesky factors $PNP^T = LL^T$ or LDL^T can be computed stably for all permutations P, and that P may therefore be chosen to preserve sparsity in L.

If N or its factors are *not* sparse or well-conditioned, the augmented system (7) may be of interest. In particular, it is better conditioned than the normal equations at times of importance—when A is ill-conditioned. By examining the eigenvalues of N and K, we obtain the following result for an arbitrary matrix A.

RESULT 1 ([19, §2]). *If $\delta > \sigma_{\min}(A)$, the condition numbers of N and K in (6)–(7) are as follows:* $\mathrm{Cond}(N) \approx (\|A\|/\delta)^2$, $\mathrm{Cond}(K) \approx \|A\|/\delta$.

If A comes from a sequence of increasingly ill-conditioned matrices, we see that regularization gives essentially *constant condition numbers*, and that K is much better conditioned than N. The implications for linear programming are pursued later.

A word of caution: If the degree of regularization is open to choice, δ should not be chosen "too small", since it could mean that $\|s\| \gg \|x\|$ in (7). Good accuracy in s may be accompanied by poor accuracy in x. With 16-digit precision, we recommend $\delta \geq 10^{-5}\|A\|$.

3 Cholesky on Quasi-definite Systems

In [21], Vanderbei introduced *symmetric quasi-definite* (sqd) matrices of the form

$$K = \begin{pmatrix} H & A^T \\ A & -G \end{pmatrix}, \qquad H \text{ and } G \text{ symmetric positive definite.} \tag{8}$$

and advocated use of sparse Cholesky-type factors, $PKP^T = LDL^T$, for solving linear systems $Kz = d$. (See also [20]. Note that A is now transposed for the remainder of the paper.)

Since the Cholesky factors exist for all permutations, P may be chosen to maintain sparsity, as in the positive definite case. However, the usual excellent stability properties of Cholesky factors do not hold when K is indefinite. We must deal with this issue.

If a *stable* method were used to factorize K and solve $Kz = d$, the relative error in $\hat{z}$ (the computed z) would be bounded by an expression of the form

$$\|\hat{z} - z\|/\|z\| \leq \epsilon\rho \,\mathrm{Cond}(K),$$

where ρ is a slowly-growing function of the dimension of K. If some other method is used to solve $Kz = d$, and if the relative error can be bounded by a similar expression with $\mathrm{Cond}(K)$ replaced by a quantity $\mathrm{Econd}(K)$, we define the latter to be an *effective condition number for* K.

An initial stability analysis of sqd systems follows from some results of Golub and Van Loan [13], as shown by Gill *et al.* [12].

RESULT 2 ([12, §4–5]). *If Cholesky factors are used to solve $Kz = d$, where K is the sqd matrix (8), an effective condition number is given by*

$$\omega(K) = \frac{\max\{\|A^TG^{-1}A\|, \|AH^{-1}A^T\|\}}{\|K\|}, \quad \mathrm{Econd}(K) = \Big(1 + \omega(K)\Big)\,\mathrm{Cond}(K).$$

Typically, $\omega(K) \gg 1$ and we can omit the 1. For the regularized least-squares system (7), this gives the following result.

RESULT 3 ([19, §2.1]). *If Cholesky factors are used to solve $Kz = d$, where K is the matrix in (7), an effective condition number is* $\mathrm{Econd}(K) \approx (\|A\|/\delta)^2$.

Comparing with Result 1, we see that the effective condition of the augmented system K is the same as the true condition of the normal-equations matrix N (when N is ill-conditioned). Hence, if the Cholesky factors of K are sufficiently sparse, they may be preferable to those of N.

3.1 Iterative Refinement

Note that the right-hand side of $Kz = d$ in Result 3 is a general vector d. If iterative refinement is applied, errors in the computed corrections will again be governed by $\mathrm{Econd}(K)$ (and the accuracy of the right-hand side). Refinement is *not* effective with the associated normal equations unless the Cholesky factors are obtained from a QR factorization of A [2], which would usually be less efficient.

4 Regularized Linear Programs

Barrier methods for linear programming (e.g., [14]) give rise to sequences of increasingly ill-conditioned least-squares or sqd systems. Here we focus on the regularized LP problem discussed in [10, 11]:

$$\begin{array}{ll} \underset{x,\,s}{\text{minimize}} & c^Tx + \frac{1}{2}\|\gamma x\|^2 + \frac{1}{2}\|p\|^2 \\ \text{subject to} & Ax + \delta p = b, \qquad l \le x \le u. \end{array} \tag{9}$$

We assume that the problem has been scaled to satisfy $\|A\| \approx 1$. The scalars γ and δ are typically "small" ($\approx 10^{-4}$). Like all good regularization parameters, they ensure that an optimal primal and dual solution (x, π) exists and is bounded and unique for any values of the data (assuming $l \le u$!).

Each iteration of an "infeasible primal-dual barrier method" requires the solution of KKT systems of the form

$$K\begin{pmatrix} \Delta x \\ -\Delta\pi \end{pmatrix} = \begin{pmatrix} w \\ r \end{pmatrix}, \qquad K \equiv \begin{pmatrix} H & A^T \\ A & -\delta^2 I \end{pmatrix}, \qquad H \equiv H_\mu + \gamma^2 I, \tag{10}$$

where H_μ is diagonal and positive semi-definite. (Its elements change every iteration.) Zero diagonals of H_μ arise when there are free variables (with infinite bounds in l and u), but setting $\gamma > 0$ removes the common difficulty of forming the normal-equations matrix $N \equiv AH^{-1}A^T + \delta^2 I$.

4.1 Two Scalings of K

In the code PDQ1 [10, 11], we remove artificial ill-conditioning by scaling down the large diagonals of H, using a diagonal matrix D_1 with $(D_1)_{jj} = (\max\{H_{jj}, 1\})^{-1/2}$. This gives an equivalent system

$$K_1\begin{pmatrix} y \\ -\Delta\pi \end{pmatrix} = \begin{pmatrix} D_1 w \\ r \end{pmatrix}, \qquad K_1 \equiv \begin{pmatrix} H_1 & D_1A^T \\ AD_1 & -\delta^2 I \end{pmatrix}, \tag{11}$$

where $\Delta x = D_1 y$, $H_1 = D_1 H D_1$, $\|H_1\| = 1$, $\|H_1^{-1}\| \approx 1/\gamma^2$, $\|D_1\| = 1$, $\|AD_1\| \approx 1$, $\|K_1\| \approx 1$. The accuracy of the solution (and the need for iterative refinement) is based on the residuals for (11). Applying Result 2 to K_1 gives the following effective condition, as previously shown in [12].

RESULT 4 ([12, RESULT 6.1]). *If Cholesky factors are used to solve (11), the effective condition number is* $\text{Econd}(K_1) \approx \max\{1/\gamma^2, 1/\delta^2\}\,\text{Cond}(K_1)$.

On a typical LP problem, the barrier method generates 20 to 50 systems with $\text{Cond}(K_1)$ tending to increase as H changes. With $\gamma = \delta = 10^{-4}$, Result 4 seems to explain the viability of $PK_1P^T = LDL^T$ factorizations within PDQ1, at least until the solution is approached (though it doesn't explain the success of Cholesky factorization in LOQO [20, 21], where P is chosen carefully without the help of regularization).

A difficulty with Result 4 is that $\text{Cond}(K_1)$ is not clearly bounded (though it may be moderate initially). A contribution of this paper is to convert (10) to a regularized least-squares system and obtain an effective condition number that is independent of H and

therefore holds for all iterations of the barrier method. Scaling with $D_2 = H^{-1/2}$ and $1/\delta$ gives the equivalent system

$$K_2 \begin{pmatrix} s \\ -\Delta\pi \end{pmatrix} = \begin{pmatrix} D_2 w \\ (1/\delta) r \end{pmatrix}, \qquad K_2 \equiv \begin{pmatrix} \delta I & D_2 A^T \\ A D_2 & -\delta I \end{pmatrix}, \tag{12}$$

where $\Delta x = \delta D_2 s$, $\|D_2\| \approx 1/\gamma$, $\|AD_2\| \approx 1/\gamma$, $\|K_2\| \approx 1/\gamma$. Applying Result 1 gives $\mathrm{Cond}(K_2) \approx \|D_2 A^T\|/\delta \approx 1/(\gamma\delta)$. Combining with Result 2 gives the following main result.

RESULT 5. *If Cholesky factors are used to solve (12), the effective condition number is* $\mathrm{Econd}(K_2) \approx 1/(\gamma^2\delta^2)$.

We see again that the effective condition of the augmented system K_2 is the same as the true condition of the normal-equations matrix $N = AD_2^2A^T + \delta^2 I$ (when N is ill-conditioned). We may therefore favor K_2 if its factors are more sparse than those of N.

4.2 Scale Invariance

In practice, the numerical solution of $Kz = d$ using Cholesky factors with a given ordering is the *same* for all symmetric scalings of K (assuming overflow and underflow do not occur). Hence, Results 4 and 5 apply equally well to the original KKT system (10). The *best* of those results serves as a stability indicator.

Thus, we are free to choose *any ordering* P for the Cholesky factors of K in (10), as long as γ and δ are sufficiently large.

5 LBL^T Factorizations

Since the true condition of K_2 in (12) is only $1/(\gamma\delta) \approx 10^8$, implementations based on a *stable* factorization of K_2 should experience few numerical difficulties *regardless of the data*.

In PDQ1 we currently use the sparse indefinite solver MA47 [6], which performs somewhat better than its predecessor MA27 [5] in this context. These packages can form both LDL^T and LBL^T factorizations (with B block diagonal). The latter provide stability in the conventional sense, but tend to be less sparse. In general we request LDL^T as long as possible, and fall back on LBL^T factors with loose but increasingly strict tolerances if the KKT systems are not solved with sufficient precision (e.g., if γ and δ are too small). An alternative would be to continue with Cholesky factors after increasing γ and δ.

Fourer and Mehrotra [8] have implemented their own LBL^T factorizer and applied it to KKT systems closely related to K_2, using very loose stability tolerances. They would probably achieve similar success on K_2 itself, with LDL^T factors resulting if γ and δ are not too small.

6 Reduced KKT Systems

The KKT system (10) is often solved by forcing a block pivot on all of H and allowing a black-box Cholesky package to choose an ordering for the resulting normal equations. This is clearly stable if γ and δ are sufficiently large. However, several real-world models in [1, 15] illustrate the need for alternatives when AA^T or L are excessively dense.

Reduced KKT systems are formed by pivoting on *part* of H (say H_s). In PDQ1, an element of H_s is required to be "sufficiently large", and the associated column of A must be "sufficiently sparse". When the regularization parameters are large enough, the partition can be based solely on sparsity, as described next.

6.1 Dense Columns or Dense Factors

Let A be partitioned as $(A_s \; A_d)$, where the columns of A_d contain *ndense* or more nonzeros. Pivoting on the first part of H gives a *reduced KKT system* of the form

$$K_r \begin{pmatrix} \Delta\pi \\ \Delta x_d \end{pmatrix} = \begin{pmatrix} r \\ -w_d \end{pmatrix}, \qquad K_r \equiv \begin{pmatrix} A_s H_s^{-1} A_s^T + \delta^2 I & A_d^T \\ A_d & -H_d \end{pmatrix}. \tag{13}$$

We then form a black-box factorization $PK_rP^T = LDL^T$. Acceptable values for *ndense* and P can be determined symbolically prior to the barrier iterations. For example, *ndense* $= 100$ might be successful in many cases (treating most columns of A as sparse), but if K_r or L exceed available storage, values such as $50, 20, 15, 10, 5, 1$ could be tried in turn. An intermediate value will probably be optimal.

6.2 Special Structures

In particular applications, A_s could be a large part of A with the property that $A_sA_s^T$ is unusually sparse. A benefit of regularization is that we are free to choose any such partition and then apply a black-box Cholesky package to the reduced matrix K_r. (Similarly for least-squares problems if many rows of the observation matrix have special structure.)

7 Numerical Results

To illustrate some effects, we report results from running the barrier code PDQ1 on two "eminent" LP problems from the Netlib collection [9]. The problems were scaled and then regularized (γ, $\delta > 0$). We requested 6 digits of accuracy in the regularized solution (x, π). (Iteration counts are about 10% greater when 8 digits are required.) Times are CPU seconds on a DEC Alpha 3000/400 workstation with about 16 digits of precision. MA47 was instructed to compute indefinite Cholesky factors of reduced KKT systems (13).

Table 1 shows the effect of varying γ and δ on problem *greenbea*, which can cause numerical difficulties in barrier methods without regularization (e.g., [21]).

1. With excessive regularization ($\gamma = \delta = 10^{-3}$), the final objective value is rather different from the optimum for the original problem. This is probably due to $\|x\|$ being unusually large.

2. With too little regularization ($\gamma = \delta = 10^{-6}$), Cholesky factorization of K_r becomes unstable. The last two iterations proceeded satisfactorily after MA47 switched to slightly more dense LBL^T factors (with stability tolerance 10^{-8}).

3. Most problems in the Netlib set have $\|x\| \approx 1$ after scaling, and give satisfactory solutions with $\gamma = \delta = 10^{-4}$. Implementations with a crossover to the simplex method should require relatively few simplex iterations to solve the original problem.

Table 2 shows the effect of varying the partition of A in forming reduced KKT systems (13) and their Cholesky factors. The values $\gamma = \delta = 10^{-4}$ gave reliable performance in all cases. $|K_r|$ is the number of terms summed to form K_r ($\times 1000$). $|L|$ is the number of nonzeros in L ($\times 1000$).

1. Problem *greenbea* is typical of "sparse" problems. Cholesky factors of N are significantly more sparse than for the full KKT system K.

2. Problem *pilots* contains a large number of *moderately* dense columns. The normal equations are reasonably efficient, but there is evidently scope for improvement with reduced KKT systems of various size (notably, *ndense*$= 5$).

TABLE 1

Barrier code PDQ1 on problem greenbea with various regularizations.

γ, δ	Itns	Final objective	$\|x\|$	$\|\pi\|$
10^{-3}	43	-6.948877×10^7	2700	25
10^{-4}	43	-7.246302×10^7	2800	54
10^{-5}	42	-7.246243×10^7	2800	1300
10^{-6}	44	-7.246264×10^7	6200	1100

TABLE 2

PDQ1 using Cholesky factors of various reduced KKT systems (13), including normal equations N and full system K. MA47 computes $PK_rP^T = LDL^T$. Normal equations are often efficient for sparse problems like greenbea. One of the reduced KKT systems (ndense = 5) is noticeably better for pilots.

	ndense	Cols in A_d	$\|K_r\|$	$\|L\|$	time	
greenbea	1000	0	102	113	54	N
	50	2	101	113	53	
	20	2	101	113	53	
	15	204	81	122	56	
	10	465	66	173	93	
	5	3833	37	272	131	
	1	5495	39	295	140	K
pilots	1000	0	530	230	187	N
	50	77	352	235	169	
	20	679	113	280	174	
	15	975	79	276	162	
	10	1432	53	290	160	
	5	2121	44	300	148	
	1	4657	48	371	170	K

8 Least Squares with Bounds

As an example of augmented systems that are regularized "naturally", we consider least-squares problems with bounded variables. Barrier methods generate systems that are increasingly ill-conditioned (as in the LP case), but again the systems can be solved with LDL^T factors, as we now show.

For simplicity, let the problem be

$$\min \|Ax - b\|^2 \quad \text{subject to} \quad x \geq 0 \tag{14}$$

and consider the function $F(x, \mu) = \frac{1}{2}\|Ax-b\|^2 - \mu \sum \ln x_j$, where μ is the barrier parameter ($\mu > 0$). A primal barrier method applies Newton's method to minimize $F(x, \mu)$ as a function of x. Each iteration requires the solution of

$$(A^TA + \mu X^{-2})\Delta x \;=\; A^Tr + \mu X^{-1}e, \tag{15}$$

where x is the current estimate ($x > 0$), $X = \text{diag}(x_j)$, $r = b - Ax$, and e is a vector of 1's. In some applications, it may be best to treat this system directly. Otherwise, we may

write it as the least-squares problem

$$\min_t \left\| \begin{pmatrix} AX \\ \delta I \end{pmatrix} t - \begin{pmatrix} r \\ \delta e \end{pmatrix} \right\|^2, \tag{16}$$

where $\delta = \sqrt{\mu}$ and $\Delta x = Xt$. The solution is given by the augmented system

$$K \begin{pmatrix} s \\ t \end{pmatrix} = \begin{pmatrix} r \\ -\delta e \end{pmatrix}, \qquad K \equiv \begin{pmatrix} \delta I & AX \\ XA^T & -\delta I \end{pmatrix}, \tag{17}$$

whose regularization parameter is *prescribed* in terms of the barrier parameter μ. Although μ tends to zero as the barrier method converges, it should comfortably satisfy $\mu \geq \|b\|^2 \epsilon$. From Result 3, we have

$$\text{Econd}(K) \approx (\|AX\|/\delta)^2 \approx (\|Ax\|/\delta)^2 \leq \|b\|^2/\mu \leq 1/\epsilon.$$

Hence, LDL^T factors of K should be sufficiently stable throughout.

9 Conclusions

Although sparse LBL^T packages are available for indefinite systems, they are inevitably more complex than Cholesky codes. Regularization expands the latter's applicability. Some final comments follow:

1. Sparse Cholesky codes are often implemented to compute LL^T factors on the assumption that they will be applied to positive definite systems. With little change they could produce LDL^T factors and allow D to have both positive and negative elements. MA27 and MA47 already do so.

2. We advocate the use of such black-box packages (and any new ones that come along) for solving sparse least-squares problems with regularization. Barrier methods for linear programming are a natural application. The parallelized Cholesky solver used in [16] is a promising candidate for wider use.

3. The same LDL^T packages may be applied to normal equations N, to full KKT systems K (10), or to the spectrum of reduced KKT matrices K_r (13). Regularization ensures adequate stability in all cases, allowing the choice to be based solely on the sparsity of K_r and its factors.

4. Reduced KKT systems promise efficiency in special cases where $A = (A_s \; A_d)$, if A_s is a large part of A and $A_s A_s^T$ is unusually sparse.

5. Similar techniques apply to bound-constrained least-squares problems. If a linear program is suspected of being infeasible, the approach of Section 8 might provide a useful "best" solution. (Alternatively, the primal-dual barrier method of Section 4 could be applied with $\delta = 1$. This has proved more effective in recent experiments on large, dense, infeasible LP problems [3].)

6. With today's 64-bit machines, the range of permissible regularization parameters is somewhat narrow. If higher precision becomes commonplace, the solution of quasi-definite systems (via Cholesky factorization) will be an important beneficiary.

7. So too will the solution of *unsymmetric* systems $Ax = b$ (via Cholesky factorization of system (7), with iterative refinement to minimize the effect of δ).

References

[1] R. E. Bixby, *Progress in linear programming*, ORSA J. on Computing, 6(1) (1994), pp. 15–22.

[2] Å. Björck, *Stability analysis of the method of seminormal equations for linear least squares problems*, Linear Alg. and its Appl., 88/89 (1987), pp. 31–48.

[3] S. Chen (private communication), Computational experiments on infeasible LP problems arising from Basis Pursuit methods for de-noising images, Dept of Statistics, Stanford University, Stanford, CA, 1995.

[4] J. E. Craig, *The N-step iteration procedures*, J. Math. and Phys., 34(1) (1955), pp. 64–73.

[5] I. S. Duff and J. K. Reid, *The multifrontal solution of indefinite sparse symmetric linear systems*, ACM Trans. Math. Softw., 9 (1983), pp. 302–325.

[6] I. S. Duff and J. K. Reid, *Exploiting zeros on the diagonal in the direct solution of indefinite sparse symmetric linear systems*, ACM Trans. Math. Softw., to appear.

[7] A. Forsgren, *On linear least-squares problems with diagonally dominant weight matrices*, Report TRITA-MAT-1995-OS2, Dept of Mathematics, Royal Institute of Technology, Stockholm, Sweden, 1995.

[8] R. Fourer and S. Mehrotra, *Solving symmetric indefinite systems in an interior-point method for linear programming*, Math. Prog., 62 (1993), pp. 15–39.

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

[10] 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, Dept of Operations Research, Stanford University, CA, 1991.

[11] ——, *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.

[12] P. E. Gill, M. A. Saunders and J. R. Shinnerl, *On the stability of Cholesky factorization for quasi-definite systems*, SIAM J. Mat. Anal., to appear.

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

[14] 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.

[15] ——, *Interior point methods for linear programming: Computational state of the art*, ORSA J. on Computing, 6(1) (1994), pp. 1–14.

[16] I. J. Lustig and E. Rothberg, *Gigaflops in linear programming*, Technical Report, 1995.

[17] C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Softw., 8(1) (1982), pp. 43–71.

[18] ——, *Algorithm 583. LSQR: Sparse linear equations and least squares problems*, ACM Trans. Math. Softw., 8(2) (1982), pp. 195–209.

[19] M. A. Saunders, *Solution of sparse rectangular systems using LSQR and CRAIG*, Report SOL 94-4, Dept of Operations Research, Stanford University, CA, 1994. To appear in BIT.

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

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

[22] S. A. Vavasis, *Stable numerical algorithms for equilibrium systems*, SIAM J. Mat. Anal., 15 (1994), pp. 1108–1131.

[23] S. Wright, *Stability of linear algebra computations in interior-point methods for linear programming*, Report MCS-P446-0694, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, 1994.