SIAM J. MAT. ANAL. APPLS
Vol. 0, No. 0, pp. 000–000, Month 1994

012

STABILITY OF LINEAR EQUATIONS SOLVERS IN INTERIOR-POINT METHODS*

STEPHEN J. WRIGHT†

Abstract. Primal-dual interior-point methods for linear complementarity and linear programming problems solve a linear system of equations to obtain a modified Newton step at each iteration. These linear systems become increasingly ill-conditioned in the later stages of the algorithm, but the computed steps are often sufficiently accurate to be useful. We use error analysis techniques tailored to the special structure of these linear systems to explain this observation and examine how theoretically superlinear convergence of a path-following algorithm is affected by the roundoff errors.

Key words. primal-dual interior-point methods, error analysis, stability

AMS(MOS) subject classifications. 65G05, 65F05, 90C33

1. Introduction. The monotone linear complementarity problem (LCP) is the problem of finding a vector pair $(x, y) \in \mathbb{R}^n \times \mathbb{R}^n$ such that

$$y = Mx + q, \qquad (x, y) \geq 0, \qquad x^T y = 0, \tag{1}$$

where M (a real, $n \times n$ positive semidefinite matrix) and q (a real vector with n elements) are given. Note that M need not be symmetric. It is well known that (1) includes the linear programming problem as a special case. Specifically, for the linear programming formulation

$$\min_z c^T z \ \text{ subject to } \ Az \geq b, \ \ z \geq 0, \tag{2}$$

where $A \in \mathbb{R}^{m \times p}$, we can introduce the dual variable $\lambda \in \mathbb{R}^m$ for the constraint $Az \geq b$ and obtain the following necessary and sufficient conditions for optimality of the primal-dual pair (z, λ):

$$\begin{bmatrix} 0 & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} z \\ \lambda \end{bmatrix} + \begin{bmatrix} c \\ -b \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \qquad \begin{bmatrix} z \\ \lambda \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tag{3a}$$

$$z^T(c - A^T\lambda) + \lambda^T(Az - b) = 0. \tag{3b}$$

For appropriate definitions of M and q, (3) has the form (1). Little is lost from either the practical or theoretical point of view by applying interior-point algorithms for (1) to the special cases of linear and convex quadratic programming, provided that the special structure of each problem is exploited in the solution of the linear systems at each iteration.

Interior-point methods for (1) generate a sequence of iterates (x^k, y^k) that are strictly positive. Many such methods require a linear system of the form

$$\begin{bmatrix} M & -I \\ Y & X \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} r \\ -XYe + \sigma\mu e \end{bmatrix}, \tag{4}$$

where

$$X = \operatorname{diag}(x_1, x_2, \ldots, x_n), \quad Y = \operatorname{diag}(y_1, y_2, \ldots, y_n), \quad e = (1, 1, \ldots, 1)^T,$$
$$\mu = x^T y/n, \qquad r = y - Mx - q, \qquad \sigma \in [0, 1],$$

* This work was based on research supported by the Office of Scientific Computing, U.S. Department of Energy, under Contract W-31-109-Eng-38.

† Mathematics and Computer Science Division, Argonne National Laboratory, 9700 South Cass Avenue, Argonne, IL 60439.

to be solved for a search direction (u, v) at each iteration. Affine-scaling methods solve (4) with $\sigma = 0$ to find a search direction, then step a fraction of the distance along this direction to the boundary of the nonnegative orthant defined by $(x, y) \geq 0$. Affine-scaling steps (u, v) are simply Newton steps for the system of nonlinear equations

$$F(x,y) = \begin{bmatrix} Mx + q - y \\ XYe \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{5}$$

Path-following methods (see, for example, Monteiro and Adler [10], Zhang [19], Wright [14]) generate steps by using generally positive values of σ in (4). (As we see later, the algorithm of [14] allows $\sigma = 0$ on some iterates in an attempt to attain the rapid local convergence associated with Newton's method.) Potential-reduction methods (see, for example, Kojima, Mizuno, and Yoshise [6], Kojima, Kurita, and Mizuno [5]) also determine search directions by solving systems like (4), but they refer to a logarithmic potential function to decide how far to move along the computed direction. Predictor-corrector methods (see, for example, Ye and Anstreicher [18], Ji, Potra, and Huang [4], Potra [12]) take steps with either $\sigma = 0$ or $\sigma = 1$.

The system (4) is highly structured; since the diagonals of X and Y are strictly positive, we can rearrange the system to obtain

$$(M + X^{-1}Y)u = r - y + \sigma\mu X^{-1}e, \tag{6a}$$

$$v = -X^{-1}Yu - y + \sigma\mu X^{-1}e. \tag{6b}$$

In the case of linear programming (2), equation (6a) contains even more structure. Its form is

$$\begin{bmatrix} Z^{-1}Y_z & -A^T \\ A & \Lambda^{-1}Y_\lambda \end{bmatrix} \begin{bmatrix} u_z \\ u_\lambda \end{bmatrix} = \begin{bmatrix} A^T\lambda - c + \sigma\mu Z^{-1}e \\ -Az + b + \sigma\mu\Lambda^{-1}e \end{bmatrix}, \tag{7}$$

where Y_z and Y_λ are positive diagonal matrices and

$$Z = \mathrm{diag}(z_1, \ldots, z_p), \qquad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_m).$$

The matrix in (7) can be made symmetric indefinite by multiplying the first block row by -1. This system can be reduced even further by eliminating either u_z or u_λ. For instance, if u_z is eliminated, we obtain

$$(AY_z^{-1}ZA^T + \Lambda^{-1}Y_\lambda)u_\lambda = (b - Az + \sigma\mu\Lambda^{-1}e) - AY_z^{-1}Z(A^T\lambda - c + \sigma\mu Z^{-1}e). \tag{8}$$

Some interior-point codes for linear programming use the formulation (8), with modifications for handling dense columns in A and for dealing with non-standard linear programming formulations (see Lustig, Marsten, and Shanno [7, 8] and Xu, Hung, and Ye [17]). Other codes, notably those of Fourer and Mehrotra [1] and Vanderbei [13], handle the formulation (7). Analysis of algorithms for these formulations are discussed in another preprint [16]. In this paper, we focus on the system arising from general monotone LCP (6), and analyze the behavior of Gaussian elimination with pivoting applied to this system.

Since for any index $i = 1, \ldots, n$, at least one of x_i and y_i is zero at the solution, we would expect some of the diagonal elements of $X^{-1}Y$ to approach zero and some to approach $+\infty$ as the solution set is approached. Hence the coefficient matrix in (6a) tends to become increasingly ill-conditioned during the later stages of the algorithm. From the standard error analysis of linear systems, we might therefore expect that

rounding errors in the step (u, v) make it useless in advancing the algorithm towards convergence. In this paper, we show that while the theoretical superlinear properties suggested by the exact analysis are not generally observed, implementations of the algorithms can still exhibit rapid convergence if the parameters are set to appropriate values. For a particular path-following infeasible-interior-point algorithm with strong theoretical convergence properties, these conclusions are presented in Section 4 and confirmed by computational experiments in Section 5. Section 3 lays the groundwork by deriving bounds on the rounding errors in the computed values of (u, v), for a wide class of algorithms that includes the algorithm of Sections 4 and 5. Section 2 presents the assumptions and a fundamental result from error analysis.

Linear systems that arise in logarithmic barrier methods for constrained optimization methods were analyzed by Ponceleón [11]. The Newton equations for each logarithmic subproblem are similar to (6a) in that the large elements occur only on the diagonal. Despite the apparent ill-conditioning of these systems, Ponceleón showed that their sensitivity to structured perturbations from a certain class is governed by the conditioning of the underlying problem and does not depend on the current value of the barrier parameter. Ponceleón's analysis is somewhat different from that of Section 3 — she looks at the relative error in the components of the solution, rather than starting with the absolute error — but her conclusions are consistent with those obtained in Section 3.

In subsequent sections, subscripts denote components of a vector, while iteration indices (usually k) appear as subscripts on scalars and as superscripts on vectors and matrices. The sets B and N form a partition of the index set $\{1, 2, \ldots, n\}$ defined in Assumption 1 below. If $x \in \mathbb{R}^n$, then

$$x_B = [x_i]_{i \in B}, \qquad X = \operatorname{diag}(x) = \operatorname{diag}(x_1, x_2, \ldots, x_n), \qquad X_B = \operatorname{diag}(x_B),$$

and so on. For the matrix $M \in \mathbb{R}^{n \times n}$, we have

$$M_{BN} = [M_{ij}]_{i \in B, j \in N},$$

and similarly for M_{BB}, M_{NB}, and M_{NN}. Given any matrix $H = [h_{ij}]$, we define $|H| = [|h_{ij}|]$ and denote the j-th column of H by $H_{\cdot j}$. The notation $\|\cdot\|$ denotes the 1-, 2-, or ∞-norm of a vector or matrix, while $\kappa(\cdot)$ denotes the corresponding condition number.

For any two nonnegative numbers ν and χ, we write $\nu = O(\chi)$ if there is a moderate constant τ such that $\nu \leq \tau\chi$. When W is a matrix or vector, we write $W = O(\chi)$ to denote $\|W\| = O(\chi)$. We use $\nu = \Omega(\chi)$ to indicate that both $\nu = O(\chi)$ and $\chi = O(\nu)$.

We use $\mathbf{u}$ to denote unit roundoff, which we define implicitly by the statement that when x and y are any two floating point numbers, op denotes $+, -, \times, /$, and $fl(z)$ denotes the floating point representation of any real number z, we have

$$fl(x \operatorname{op} y) = (x \operatorname{op} y)(1 + \delta), \qquad |\delta| \leq \mathbf{u}. \tag{9}$$

(See Golub and Van Loan [2, Section 2.4.2].) We assume throughout that $\mathbf{u}$ is small enough that $O(\mathbf{u}) \ll 1$, where $O(\cdot)$ is the order notation defined above.

2. Assumptions and Basic Results. In the remainder of the paper, we focus on path-following interior-point methods. "Infeasible" variants of these methods are among the most widely used practical algorithms for linear programming and LCPs [7, 8] and have also been the subject of extensive theoretical investigations, which

have shown that they can have strong convergence properties under weak assumptions [19, 15, 14]. The path-following infeasible-interior-point framework stated below also includes the class of predictor-corrector methods, for appropriate choices of the initial point and parameters.

Path-following algorithms restrict their iterates (x^k, y^k) to neighborhoods of the form

$$\mathcal{N}(\gamma) = \{(x,y) > 0 \,|\, x_i y_i \geq \gamma(x^T y/n)\} \;\subset\; \mathbb{R}^n_+ \times \mathbb{R}^n_+, \tag{10}$$

where $\mathbb{R}^n_+$ denotes the nonnegative orthant in $\mathbb{R}^n$. All iterates generated by the algorithms lie in $\mathcal{N}(\gamma_{\min})$, where $\gamma_{\min} \in (0, 1/2)$ is a constant. Other quantities needed to define the general algorithm include

$$\mu_k = (x^k)^T y^k/n, \qquad r^k = y^k - Mx^k - q, \qquad \gamma_{\max} \in (\gamma_{\min}, 1/2],$$
$$X^k = \mathrm{diag}(x^k), \qquad Y^k = \mathrm{diag}(y^k).$$

We define the algorithmic framework as follows:

Algorithm PFI
given $\gamma_0 \in [\gamma_{\min}, \gamma_{\max}]$ and $(x^0, y^0) \in \mathcal{N}(\gamma_0)$

for $k = 0, 1, 2, \ldots$
choose $\sigma_k \in [0, 1]$ and find (u^k, v^k) that satisfies

$$\begin{bmatrix} M & -I \\ Y^k & X^k \end{bmatrix} \begin{bmatrix} u^k \\ v^k \end{bmatrix} = \begin{bmatrix} r^k \\ -X^k Y^k e + \sigma_k \mu_k e \end{bmatrix}; \tag{11}$$

choose $\gamma_{k+1} \in [\gamma_{\min}, \gamma_{\max}]$ and $\alpha_k > 0$ such that

$$(x^{k+1}, y^{k+1}) = (x^k, y^k) + \alpha_k (u^k, v^k) \in \mathcal{N}(\gamma_{k+1})$$

and $\prod_{j=0}^{k}(1 - \alpha_j) \leq K\mu_{k+1}/\mu_0$ when $r^0 \neq 0$, for some constant $K > 0$;
end (for)

The decrease in $\|r^k\|$ at each iteration is linear — in fact, $r^{k+1} = \prod_{j=0}^{k}(1-\alpha_j) r^0$ — so the last condition in Algorithm PFI is equivalent to

$$\|r^{k+1}\|/\|r^0\| \leq K\mu_{k+1}/\mu_0. \tag{12}$$

Hence, $\|r_k\| = O(\mu_k)$ for all k, so the infeasibility is always bounded by a multiple of the complementarity gap μ.

When the initial point is feasible ($r^0 = 0$), predictor-corrector algorithms such as that of Ji, Potra, and Huang [4] are special cases of Algorithm PFI. This framework also includes the infeasible-interior-point algorithms of Zhang [19] and Wright [15, 14]. These algorithms choose γ_{k+1} and σ_k so that a step α_k of nontrivial length can always be taken without violating the required conditions.

In practical implementations of interior-point methods, the framework of Algorithm PFI is usually modified slightly. In linear programming codes, different step lengths are usually chosen for the primal and dual components of x, as experience has shown that this strategy tends to reduce the number of iterations slightly. Moreover, explicit membership of the neighborhood (10) is usually not enforced. (A more common strategy, for which there is no supporting theory, is to find the largest value of

α in $[0, 1]$ that keeps $(x^k, y^k) + \alpha(u^k, v^k)$ in the nonnegative orthant and then choose α_k to be a fixed fraction of this length.) The predictor-corrector strategy of Mehrotra [9], used also in the codes of Lustig, Marsten, and Shanno [8], Vanderbei [13], and Xu, Hung, and Ye [17], adds extra terms to the lower part of the right-hand side on "corrector" iterations. Nevertheless, the coefficient matrices used in these practical algorithms are the same as in (11), and our conclusions about the accuracy of the computed steps continue to hold, with minor modifications to the analysis of Section 3.

For most of our analysis, we make the following assumptions about the data for problem (1) and its solution set:

ASSUMPTION 1.

a) Problem (1) has a unique solution (x^*, y^*) *such that* $x^* + y^* > 0$ *(i.e.* strict complementarity *holds). We can define an associated partition* B, N *of the index set* $\{1, \ldots, n\}$ *such that* $x_i^* > 0$ *for all* $i \in B$ *and* $y_i^* > 0$ *for all* $i \in N$.

b) The quantities

$$\|M\|, \qquad \|M_{BB}^{-1}\|, \qquad \|X_B^*\|, \qquad \|(X_B^*)^{-1}\|, \qquad \|Y_N^*\|, \qquad \|(Y_N^*)^{-1}\|$$

are all moderate in size.

Assumption 1 implies that the coefficient matrix in (11) approaches a nonsingular limit, since there are $2n \times 2n$ permutation matrices P and Π such that

$$P \begin{bmatrix} M & -I \\ Y^* & X^* \end{bmatrix} \Pi = \begin{bmatrix} X_B^* & 0 & 0 & 0 \\ 0 & Y_N^* & 0 & 0 \\ -I & M_{BN} & M_{BB} & 0 \\ 0 & M_{NN} & M_{NB} & -I \end{bmatrix}, \tag{13}$$

and each of the submatrices on the diagonal of (13) is nonsingular.

When the problem (1) is derived from a linear program as in (3), existence of a solution implies existence of a strictly complementary solution. However, for both this special case and the general case of M symmetric positive semidefinite, uniqueness of the solution and well-conditioning of M_{BB} are often not satisfied in practice, so Assumption 1 is quite strong. As we see, however, this Assumption plays an important role in showing that the errors in the computed solutions are not disastrous for the interior-point algorithm, just as well-conditioning of the square coefficient matrix A in a linear system $Az = b$ is needed to ensure that the relative errors in the computed version of z are not too large. Our computational experience (Section 5) tends to indicate that Assumption 1 is necessary as well as sufficient for rapid local convergence of the algorithm.

We make one further assumption on the iterates generated by the basic algorithm.

ASSUMPTION 2. *The iterates generated by Algorithm PFI satisfy*

$$\lim_{k \to \infty} (x^k, y^k) = (x^*, y^*).$$

Of course, it is not necessary to make this assumption for any reasonable instance of Algorithm PFI, since convergence to a solution should be one of the properties implied by the particular schemes for choosing σ_k, α_k, and γ_k. We make this assumption here merely to divorce the error estimates of the next section from any particular variant of Algorithm PFI.

By the implicit function theorem, the nonsingularity of the matrix (13), equation (11), and $\|r^k\| = O(\mu_k)$, we have

$$\|(u^k, v^k)\| = O(\mu_k), \qquad \|(x^k, y^k) - (x^*, y^*)\| = O(\mu_k). \tag{14}$$

We also have the following simple result.

LEMMA 2.1. *There are positive constants C_1 and C_2 such that for all k sufficiently large, we have*

$$C_1\mu_k \le x_i^k \le C_2\mu_k, \qquad \forall i \in N, \tag{15a}$$

$$C_1\mu_k \le y_i^k \le C_2\mu_k, \qquad \forall i \in B. \tag{15b}$$

Proof. Because of Assumption 2, we can define an index K and positive constants $\bar{C}_1$ and $\bar{C}_2$ such that

$$x_i^k \in [\bar{C}_2, \bar{C}_1], \qquad \forall k \ge K, \ \forall i \in B,$$
$$y_i^k \in [\bar{C}_2, \bar{C}_1], \qquad \forall k \ge K, \ \forall i \in N.$$

Therefore, since $(x^k, y^k) > 0$, we have for $k \ge K$, $i \in N$, that

$$x_i^k y_i^k < (x^k)^T y^k = n\mu_k \quad \Rightarrow \quad x_i^k < \frac{n\mu_k}{y_i^k} \le \frac{n\mu_k}{\bar{C}_2}.$$

Also,

$$x_i^k y_i^k \ge \gamma_k \mu_k \ge \gamma_{\min}\mu_k \quad \Rightarrow \quad x_i^k \ge \frac{\gamma_{\min}\mu_k}{y_i^k} \ge \frac{\gamma_{\min}\mu_k}{\bar{C}_1}.$$

Therefore (15a) holds with $C_1 = \gamma_{\min}/\bar{C}_1$ and $C_2 = n/\bar{C}_2$. The proof of (15b) is similar. ∎

Finally, we state a result from the roundoff error analysis of Gaussian elimination, for reference in the next section.

THEOREM 2.2. *Suppose that the $m \times m$ linear system $Az = b$ is solved by using Gaussian elimination, possibly with row and/or column pivoting. Let us denote the row permutation matrix by P, the column permutation matrices by Π, the computed unit lower triangular factor by $\hat{L}$, and the computed upper triangular factor by $\hat{U}$. Then the computed solution $\hat{z}$ solves the perturbed system $(A + H)\hat{z} = b$, where*

$$|PH\Pi| \le \epsilon_m(2 + \epsilon_m)|\hat{L}||\hat{U}|, \tag{16}$$

and $\epsilon_m = m\mathbf{u}/(1 - m\mathbf{u}) = O(\mathbf{u})$.

Proof. Follows immediately from Theorem 6.4 of Higham [3]. ∎

During Gaussian elimination, the size of the largest element in each column of the remaining submatrix may grow as multiples of the pivot rows are added to later rows in the matrix. We quantify this growth by the growth factor ρ, defined as the smallest positive number such that

$$\max_{i=1,\ldots,m} |\hat{U}_{ij}| \le \rho \max_{i=1,\ldots,m} |(PA\Pi)_{ij}|, \qquad \text{for all } j = 1, 2, \ldots, m. \tag{17}$$

We then have the following simple corollary of Theorem 2.2.

COROLLARY 2.3. *Let the system $Az = b$ be as in Theorem 2.2, and suppose the pivots $\hat{U}_{jj}$ are chosen so that $|\hat{L}_{ij}| \le 1$ for all $i = 2, \ldots, m$, $j = 1, \ldots, i-1$. Then*

$$\|H_{\cdot j}\| \le m\epsilon_m(2 + \epsilon_m)\rho\|A_{\cdot j}\|\mathbf{u}. \tag{18}$$

3. Error Bounds for the Steps. In this section, we derive estimates for the difference between the step actually computed by solving (6), which we denote by $(\hat{u}, \hat{v})$, and the corresponding exact values, denoted by (u, v). We treat the cases in which $(\hat{u}, \hat{v})$ is determined by Gaussian elimination with row partial pivoting and with complete pivoting.

We start with a purely technical result.

LEMMA 3.1. *Let G be a square matrix partitioned as*

$$G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix},$$

where G_{11} and G_{22} are also square. Suppose that G_{11} and $G_{22} - G_{21}G_{11}^{-1}G_{12}$ are nonsingular. Then G is nonsingular and G^{-1} has the form

$$\begin{bmatrix} G_{11}^{-1} + G_{11}^{-1}G_{12}(G_{22} - G_{21}G_{11}^{-1}G_{12})^{-1}G_{21}G_{11}^{-1} & -G_{11}^{-1}G_{12}(G_{22} - G_{21}G_{11}^{-1}G_{12})^{-1} \\ -(G_{22} - G_{21}G_{11}^{-1}G_{12})^{-1}G_{21}G_{11}^{-1} & (G_{22} - G_{21}G_{11}^{-1}G_{12})^{-1} \end{bmatrix}.$$

Our first main result concerning components of $\hat{u}$ is the following.

THEOREM 3.2. *Let $\hat{u}$ be computed by applying Gaussian elimination with row partial pivoting to (6a), and suppose that the growth factor ρ is not too large. Then for all μ sufficiently small, we have*

$$\|\hat{u}_B\| = O(\mathbf{u} + \mu), \qquad \|\hat{u}_N\| = O(\mu).$$

Proof. We assume that μ is much smaller than all the quantities in Assumption 1(b), so that $\mu \ll 1$. We retain only the lowest-order terms in $\mathbf{u}$ and μ in the analysis since, by our assumptions, higher order terms are small enough to be absorbed into lower-order terms with minor perturbations of the coefficients.

From Theorem 2.2 and (18), we have, by permuting the rows and columns of (6a), that

$$\begin{bmatrix} M_{BB} + X_B^{-1}Y_B + E_{BB} & M_{BN} + E_{BN} \\ M_{NB} + E_{NB} & M_{NN} + X_N^{-1}Y_N + E_{NN} \end{bmatrix} \begin{bmatrix} \hat{u}_B \\ \hat{u}_N \end{bmatrix} = \begin{bmatrix} r_B - y_B - \sigma\mu X_B^{-1}e \\ r_N - y_N - \sigma\mu X_N^{-1}e \end{bmatrix}, \tag{19}$$

where

$$\|E_{BB}\| \leq \rho O(\|M_{BB} + X_B^{-1}Y_B\| + \|M_{NB}\|)\mathbf{u}, \tag{20a}$$

$$\|E_{NB}\| \leq \rho O(\|M_{BB} + X_B^{-1}Y_B\| + \|M_{NB}\|)\mathbf{u}, \tag{20b}$$

$$\|E_{BN}\| \leq \rho O(\|M_{NN} + X_N^{-1}Y_N\| + \|M_{BN}\|)\mathbf{u}, \tag{20c}$$

$$\|E_{NN}\| \leq \rho O(\|M_{NN} + X_N^{-1}Y_N\| + \|M_{BN}\|)\mathbf{u}. \tag{20d}$$

Now from Assumption 1(b) and Lemma 2.1, we have

$$\|X_B^{-1}Y_B\| = O(\mu), \qquad \|X_N^{-1}Y_N\| = O(\mu^{-1}),$$

and $\|M_{BB}\|$, $\|M_{BN}\|$, $\|M_{NB}\|$, and $\|M_{NN}\|$ are all $O(1)$. Combining these observations with (20), we obtain

$$\|E_{BB}\| = O(\mathbf{u}), \qquad \|E_{NB}\| = O(\mathbf{u}), \qquad \|E_{BN}\| = O(\mu^{-1}\mathbf{u}), \qquad \|E_{NN}\| = O(\mu^{-1}\mathbf{u}).$$

Therefore (19) can be rewritten as

$$\begin{bmatrix} M_{BB}+\bar{E}_{BB} & M_{BN}+\bar{E}_{BN} \\ M_{NB}+\bar{E}_{NB} & M_{NN}+X_N^{-1}Y_N+\bar{E}_{NN} \end{bmatrix} \begin{bmatrix} \hat{u}_B \\ \hat{u}_N \end{bmatrix} = \begin{bmatrix} b_B \\ b_N \end{bmatrix}, \tag{21}$$

where

$$\begin{aligned} &\|\bar{E}_{BB}\| = O(\mu+\mathbf{u}), && \|\bar{E}_{NB}\| = O(\mathbf{u}), \\ &\|\bar{E}_{BN}\| = O(\mu^{-1}\mathbf{u}), && \|\bar{E}_{NN}\| = O(\mu^{-1}\mathbf{u}), \end{aligned} \tag{22}$$

and

$$b_B = r_B - y_B - \sigma\mu X_B^{-1}e, \qquad b_N = r_N - y_N - \sigma\mu X_N^{-1}e.$$

If we denote the coefficient matrix in (21) by G, with $G_{11} = M_{BB} + \bar{E}_{BB}$ and so on, we have from Assumption 1(b) and Lemma 3.1 that

$$G_{11}^{-1} = [I + M_{BB}^{-1}\bar{E}_{BB}]^{-1}M_{BB}^{-1} = M_{BB}^{-1} + O(\mu+\mathbf{u}).$$

Since

$$\|G_{21}\| = O(1), \qquad \|G_{12}\| = O(1+\mu^{-1}\mathbf{u}), \qquad \|G_{11}^{-1}\| = O(1),$$

and since $\|X_N Y_N^{-1}\| = O(\mu)$ from (15a), we have

$$\begin{aligned} G_{22} - G_{21}G_{11}^{-1}G_{12} &= X_N^{-1}Y_N\left[I + X_NY_N^{-1}(M_{NN}+\bar{E}_{NN}) - X_NY_N^{-1}G_{21}G_{11}^{-1}G_{12}\right] \\ &= X_N^{-1}Y_N\left[I + O(\mu)O(1+\mu^{-1}\mathbf{u}) + O(\mu)O(1)O(1)O(1+\mu^{-1}\mathbf{u})\right] \\ &= X_N^{-1}Y_N[I + O(\mu+\mathbf{u})]. \end{aligned}$$

Hence

$$(G_{22} - G_{21}G_{11}^{-1}G_{12})^{-1} = (I + O(\mu+\mathbf{u}))\,Y_N^{-1}X_N = O(\mu).$$

Substitution in Lemma 3.1, together with Assumption 1(b) and some manipulation, yields

$$\begin{aligned} &(G^{-1})_{11} = M_{BB}^{-1} + O(\mu+\mathbf{u}), && (G^{-1})_{12} = O(\mu+\mathbf{u}), \\ &(G^{-1})_{21} = O(\mu), && (G^{-1})_{22} = O(\mu). \end{aligned}$$

We have $\|r\| = O(\mu)$ in exact arithmetic, but roundoff errors in the calculation of $r = y - Mx - q$ restrict us to assuming that $\|r\| = O(\mu+\mathbf{u})$. Using this fact together with Assumption 1(b), formula (14), and Lemma 2.1, we have

$$\|b_B\| = O(\mu+\mathbf{u}), \qquad \|b_N\| = O(1).$$

Therefore, from (21), we have

$$\begin{aligned} \hat{u}_B &= (G^{-1})_{11}b_B + (G^{-1})_{12}b_N = O(\mu+\mathbf{u}), \\ \hat{u}_N &= (G^{-1})_{21}b_B + (G^{-1})_{22}b_N = O(\mu), \end{aligned}$$

as required. ∎

Our next result bounds the difference between u and $\hat{u}$.

THEOREM 3.3. *Suppose the assumptions of Theorem 3.2 hold. Then for all μ sufficiently small, we have*

$$\|\hat{u}_B - u_B\| = O(\mu + \mathbf{u}), \tag{23a}$$

$$\|\hat{u}_N - u_N\| = O(\mu(\mu + \mathbf{u})), \tag{23b}$$

and, for all $i \in N$,

$$u_i/x_i = -1 + O(\sigma + \mu), \tag{24a}$$

$$\hat{u}_i/x_i = -1 + O(\sigma + \mu + \mathbf{u}). \tag{24b}$$

Proof. The expression (23a) follows immediately from (14) and Theorem 3.2, since

$$\|\hat{u}_B - u_B\| \le \|\hat{u}_B\| + \|u_B\| = O(\mu + \mathbf{u}).$$

For (23b), note first from (6) and (21) that

$$M_{NB}u_B + (M_{NN} + X_N^{-1}Y_N)u_N = b_N = (M_{NB} + \bar{E}_{NB})\hat{u}_B + (M_{NN} + X_N^{-1}Y_N + \bar{E}_{NN})\hat{u}_N.$$

Hence, from (14), (22), (23a), Assumption 1(b), and Theorem 3.2, we have

$$\begin{aligned} X_N^{-1}Y_N(u_N - \hat{u}_N) &= M_{NB}(\hat{u}_B - u_B) + M_{NN}(\hat{u}_N - u_N) + \bar{E}_{NB}\hat{u}_B + \bar{E}_{NN}\hat{u}_N \\ &= O(\mu + \mathbf{u}) + O(\mu) + O(\mu\mathbf{u}) + O(\mathbf{u}) = O(\mu + \mathbf{u}). \end{aligned}$$

From (15a), we have $X_N Y_N^{-1} = O(\mu)$ and so (23b) is proved.

From (4), we have

$$y_i u_i + x_i v_i = -x_i y_i + \sigma\mu$$

and therefore

$$\frac{u_i}{x_i} = -1 - \frac{v_i}{y_i} + \frac{\sigma\mu}{x_i y_i}. \tag{25}$$

Because $x_i y_i \ge \gamma_{\min}\mu$, we have $\mu/(x_i y_i) = O(1)$. Also from Assumption 2 and (14), we have for $i \in N$ that $v_i/y_i = O(\mu)$. Hence, (24a) is obtained by using these estimates in (25). For (24b), we have from (15a), (23b), and (24a) that for all $i \in N$,

$$\frac{\hat{u}_i}{x_i} = \frac{u_i}{x_i} + \frac{1}{x_i}O(\mu(\mu + \mathbf{u})) = -1 + O(\mu + \sigma) + O(\mu + \mathbf{u}).$$

■

Note that in Theorems 3.2 and 3.3, we have ignored possible errors that are introduced into the computation during the formation of the right-hand side b from the vectors r, x, and y, and the scalars σ and μ. Since the formation process introduces a relative perturbation of $O(\mathbf{u})$ into each component of b, we lose nothing by ignoring the perturbations.

We now turn to recovery of the step $\hat{v}$. From the exact formula (6b), we have

$$v_i = -y_i(1 + u_i/x_i) + \sigma\mu/x_i. \tag{26}$$

In the actual computation of $\hat{v}$, we have only the computed value $\hat{u}$ available to us. Moreover, errors are introduced when each of the five or six floating-point operations

on the right-hand side of (26) are performed. The exact nature of these errors will depend on the order in which the operations in (26) are performed. Two possibilities are suggested by the parentheses in the expressions

$$-y_i[1+(u_i/x_i)]+(\sigma\mu)/x_i, \qquad [-y_i-(y_iu_i)/x_i]+\sigma(\mu/x_i).$$

We can, however, perform an analysis that takes all the possibilities into account, as we show in the following theorem.

THEOREM 3.4. *Suppose the assumptions of Theorem 3.3 hold and that $\hat{v}$ is computed from the formula (6b) (equivalently, (26)) with $\hat{u}$ replacing u. Then we have*

$$\|\hat{v}_B - v_B\| = O(\mu(\mu+\mathbf{u})), \tag{27a}$$

$$\|\hat{v}_N - v_N\| = O(\mu+\mathbf{u}), \tag{27b}$$

and, for all $i \in B$,

$$v_i/y_i = -1+O(\sigma+\mu), \tag{28a}$$

$$\hat{v}_i/y_i = -1+O(\sigma+\mu+\mathbf{u}). \tag{28b}$$

Proof. In all the formulae of this proof, we use the notation δ_j, $j = 1,2,\ldots$, to represent scalar quantities of order $\mathbf{u}$ (We certainly have $\delta_j \leq 10\mathbf{u}$ throughout the proof.) Recall that relative errors of $O(\mathbf{u})$ are incurred whenever a real number is approximated by a floating-point number and when an arithmetic operation involving two floating-point numbers is performed (cf. (9)). Therefore, regardless of the order in which the operations required to recover $\hat{v}_i$ are performed, we have

$$\hat{v}_i = \left[-y_i\left(1+\delta_1+\frac{\hat{u}_i}{x_i}(1+\delta_2)\right)(1+\delta_3)+\frac{\sigma\mu}{x_i}(1+\delta_4)\right](1+\delta_5). \tag{29}$$

By rearranging and combining terms in (29), we obtain

$$\begin{aligned}\hat{v}_i &= -y_i(1+\delta_6)-\frac{y_i\hat{u}_i}{x_i}(1+\delta_7)+\frac{\sigma\mu}{x_i}(1+\delta_8)\\ &= -y_i-\frac{y_iu_i}{x_i}+\frac{\sigma\mu}{x_i}+\frac{y_i}{x_i}(u_i-\hat{u}_i)-\delta_6y_i-\delta_7\frac{y_i\hat{u}_i}{x_i}-\delta_8\frac{\sigma\mu}{x_i}.\end{aligned}$$

By substituting from (26), we obtain

$$|v_i-\hat{v}_i| = \left|\frac{y_i}{x_i}\right||u_i-\hat{u}_i|+O(\mathbf{u})\left[|y_i|+\left|\frac{y_i\hat{u}_i}{x_i}\right|+\left|\frac{\sigma\mu}{x_i}\right|\right]. \tag{30}$$

Consider first the case of $i \in B$. From (30) together with Assumption 2, Theorem 3.2, expressions (15) and (23a), and $\delta_j = O(\mathbf{u})$, we have

$$|v_i-\hat{v}_i| = O(\mu(\mu+\mathbf{u}))+O(\mathbf{u})\left[O(\mu)+O(\mu(\mu+\mathbf{u}))+O(\sigma\mu)\right] = O(\mu(\mu+\mathbf{u})),$$

proving (27a). For $i \in N$, we have from Assumption 2, Theorem 3.2, and expressions (15) and (23b) that

$$\left|\frac{y_i}{x_i}\right| = O(\mu^{-1}), \qquad |u_i-\hat{u}_i| = O(\mu(\mu+\mathbf{u})), \qquad \left|\frac{y_i\hat{u}_i}{x_i}\right| = O(1), \qquad \left|\frac{\sigma\mu}{x_i}\right| = O(1).$$

Therefore we obtain from (30) that

$$|v_i - \hat{v}_i| = O(\mu + \mathbf{u}), \qquad \forall i \in N,$$

as required.

The inequalities (28) are proved in the same way as (24). ∎

Gaussian elimination with complete pivoting is possibly more relevant to practical algorithms, since sparse elimination algorithms rearrange both rows and columns and hence can be regarded as approximations to the complete pivoting strategy. The main error results for complete pivoting are the same as those for partial pivoting. To justify this claim, we note first that the nonbasic indices will eventually be used as pivots before any of the basic indices are used, because of the large sizes of y_i/x_i, $i \in N$. Moreover, the error matrices E_{BN} and E_{NN} are $O(\mathbf{u})$ rather than $O(\mu^{-1}\mathbf{u})$, because the elements y_i/x_i, $i \in N$ cannot appear in a pivot row except on the diagonal, so they cannot "contaminate" other elements in the nonbasic columns of $M + X^{-1}Y$. In other words, $\hat{u}$ actually solves the system

$$\begin{bmatrix} M_{BB} + X_B^{-1}Y_B + E_{BB} & M_{BN} + E_{BN} \\ M_{NB} + E_{NB} & M_{NN} + X_N^{-1}Y_N + E_{NN} \end{bmatrix} \begin{bmatrix} \hat{u}_B \\ \hat{u}_N \end{bmatrix} = \begin{bmatrix} r_B - y_B - \sigma\mu X_B^{-1}e \\ r_N - y_N - \sigma\mu X_N^{-1}e \end{bmatrix}, \tag{31}$$

where

$$\|E_{BB}\| \leq \rho O(\|M_{BB} + X_B^{-1}Y_B\| + \|M_{NB}\|)\mathbf{u} = O(\mathbf{u}), \tag{32a}$$
$$\|E_{NB}\| \leq \rho O(\|M_{BB} + X_B^{-1}Y_B\| + \|M_{NB}\|)\mathbf{u} = O(\mathbf{u}), \tag{32b}$$
$$\|E_{BN}\| \leq \rho O(\|M_{NN}\| + \|M_{BN}\|)\mathbf{u} = O(\mathbf{u}), \tag{32c}$$
$$\|E_{NN}\| \leq \rho O(\|M_{NN}\| + \|M_{BN}\|)\mathbf{u} = O(\mathbf{u}). \tag{32d}$$

By defining G as the coefficient matrix in (31) and partitioning as before, we obtain after some manipulation that

$$(G^{-1})_{11} = M_{BB}^{-1} + O(\mu + \mathbf{u}), \qquad (G^{-1})_{12} = O(\mu),$$
$$(G^{-1})_{21} = O(\mu), \qquad (G^{-1})_{22} = O(\mu).$$

Therefore, using $\|b_B\| = O(\mu + \mathbf{u})$ and $\|b_N\| = O(1)$, we have

$$\hat{u}_B = (G^{-1})_{11}b_B + (G^{-1})_{12}b_N = O(\mu + \mathbf{u}),$$
$$\hat{u}_N = (G^{-1})_{21}b_B + (G^{-1})_{22}b_N = O(\mu),$$

which is the same error result as the one obtained for partial pivoting in Theorem 3.2. The other results in the section also hold for complete pivoting, with minor modifications to the proofs.

4. Effect of Roundoff Error on Local Convergence. We now consider the algorithm in [14], which can be described as follows. Given parameters $\gamma_{k+1} \in (\gamma_{\min}, \gamma_k]$ and $\beta_k \in [0, 1)$, the step α_k is chosen as

$$\alpha_k = \arg\min_{\alpha \in (0,1]} \mu_k(\alpha) \stackrel{\triangle}{=} (x^k + \alpha u^k)^T(y^k + \alpha v^k)/n, \tag{33}$$

subject to

$$(x^k, y^k) + \alpha(u^k, v^k) \in \mathcal{N}(\gamma_{k+1}), \tag{34a}$$

$$\mu_k(\alpha) \geq (1-\alpha)(1-\beta_k)\mu_k, \qquad \text{if } r^k \neq 0, \tag{34b}$$

for all $\alpha \in [0, \alpha_k]$. The choices of σ_k, γ_{k+1}, and β_k at each iteration are made according to the following scheme.

given $\bar{\gamma} \in (0,1)$, $\gamma_{\min}$, $\gamma_{\max}$ with $0 < \gamma_{\min} < \gamma_{\max} < 1/2$, $\bar{\sigma} \in (0, 1/2)$,
$\rho \in (0, \bar{\gamma})$, and (x^0, y^0) with $x_i^0 y_i^0 \geq \gamma_{\max}\mu_0 > 0$;

$t_0 \leftarrow 1$, $\gamma_0 \leftarrow \gamma_{\max}$;

for $k = 0, 1, 2, \ldots$

if $\mu_k = 0$ **then** stop;

Find (u^k, v^k) and α_k from (11), (33), and (34) with
$\sigma_k = 0$, $\beta_k = \bar{\gamma}^{t_k}$, $\gamma_{k+1} = \gamma_{\min} + \bar{\gamma}^{t_k}(\gamma_{\max} - \gamma_{\min})$;
if $\mu_k(\alpha_k) \leq \rho\mu_k$
then accept this step; $t_{k+1} \leftarrow t_k + 1$; go to next k;
end if

Find (u^k, v^k) and α_k from (11), (33), and (34) with
$\sigma_k \in [\bar{\sigma}, 1/2]$, $\beta_k = 0$, $\gamma_{k+1} = \gamma_k$;
accept this step; $t_{k+1} \leftarrow t_k$; go to next k;

end for.

This algorithm takes two types of steps — "safe" steps, for which $\sigma_k \geq \bar{\sigma}$, and "fast" steps, for which $\sigma_k = 0$. Theoretically, the safe steps ensure good global convergence properties and complexity, while the fast steps ensure asymptotic superlinear convergence. The counter t_k keeps track of the number of fast steps taken prior to iteration k.

The choice of step length α_k ensures that $\|r^{k+1}\| = O(\mu_{k+1})$. To see this, note from condition (34b) that

$$\mu_{k+1} = (1-\alpha_k)(1-\beta_k)\mu_k = \left[\prod_{j=0}^{k}(1-\alpha_j)(1-\beta_j)\right]\mu_0.$$

Since

$$\prod_{j=0}^{k}(1-\beta_j) \geq \prod_{j=0}^{\infty}(1-\bar{\gamma}^j) = \hat{\beta} > 0, \qquad r^{k+1} = \prod_{j=0}^{k}(1-\alpha_j)r^0,$$

we have

$$\mu_{k+1}/\mu_0 \geq \hat{\beta}\|r^{k+1}\|/\|r^0\|,$$

so condition (12) holds with $K = \hat{\beta}^{-1}$.

We focus on this algorithm because of its strong theoretical properties, namely, global convergence from any positive starting point (x^0, y^0), polynomial complexity when properly initialized, and superlinear local convergence. Also, the method performs well in computational tests and is quite similar (at least in its "non-superlinear" phase where $\sigma_k \geq \bar{\sigma}$) to the algorithm implemented by Lustig, Marsten, and Shanno [8]. We assume throughout that finite termination does not occur, that is, the algorithm generates an infinite sequence of strictly positive iterates (x^k, y^k).

In this section, we examine how the behavior of this algorithm is affected when the computed steps $(\hat{u}^k, \hat{v}^k)$ are used in place of the exact steps (u^k, v^k). We start by showing that near-unit steplengths can eventually be taken by this algorithm without violating the positivity condition $(x^k, y^k) > 0$. Consequently, there exists the possibility of rapid convergence of the sequence of complementarity gaps μ_k to zero, even in the presence of roundoff error. We refine the results to show that for the safe steps $(\sigma_k \geq \bar{\sigma})$, we actually have $\alpha_k = 1$ when μ_k is sufficiently small.

In all the analysis below, our convention is to use the iteration index k in the statement of each result, but omit it in the proofs.

LEMMA 4.1. *For all k sufficiently large, we have*

$$(x^k + \alpha \hat{u}^k, y^k + \alpha \hat{v}^k) \geq 0,$$

for all $\alpha \in [0, \bar{\alpha}_k]$, where

$$|1 - \bar{\alpha}_k| = O(\sigma_k + \mu_k + \mathbf{u}). \tag{35}$$

Proof. We consider first the indices $i \in N$. From (27b), we have

$$|\hat{v}_i| = |v_i| + O(\mu + \mathbf{u}) = O(\mu + \mathbf{u}),$$

while from Assumption 2 we have for large k that $y_i^k \approx y_i^* > 0$. Hence $y_i^k + \alpha \hat{v}_i^k > 0$ for all $\alpha \in [0, 1]$ and all k sufficiently large. On the other hand, we have from (24b) that

$$x_i + \alpha \hat{u}_i = x_i + \alpha x_i(-1 + O(\sigma + \mu + \mathbf{u})).$$

Therefore, if $x_i + \alpha_i \hat{u}_i = 0$ for some index i, then we must have

$$1 - \alpha + \alpha O(\sigma + \mu + \mathbf{u}) = 0 \quad \Rightarrow \quad |1 - \alpha| = O(\sigma + \mu + \mathbf{u}).$$

Hence $x_i + \alpha \hat{u}_i \geq 0$ for all $\alpha \in [0, \bar{\alpha}]$, for $\bar{\alpha}$ satisfying (35).

The case of $i \in B$ is proved in a similar way by using (28b). ∎

We now show that near-unit steps produce fast linear convergence of the complementarity gap to zero.

THEOREM 4.2. *If k is sufficiently large, then*

$$(x^k + \alpha_k \hat{u}^k)^T (y^k + \alpha_k \hat{v}^k) = [1 - \alpha_k(1 - \sigma_k) + O(\mu_k + \mathbf{u})](x^k)^T y^k.$$

Proof. For any $i = 1, \ldots, n$, we have from the second part of equation (4) that

$$\begin{aligned} & (x_i + \alpha \hat{u}_i)(y_i + \alpha \hat{v}_i) \\ &= x_i y_i + \alpha y_i u_i + \alpha x_i v_i + \alpha y_i(\hat{u}_i - u_i) + \alpha x_i(\hat{v}_i - v_i) + \alpha^2 \hat{u}_i \hat{v}_i \\ &= (1 - \alpha) x_i y_i + \alpha \sigma \mu + \alpha y_i(\hat{u}_i - u_i) + \alpha x_i(\hat{v}_i - v_i) + \alpha^2 \hat{u}_i \hat{v}_i. \end{aligned} \tag{36}$$

Now, by Assumption 2 and relations (15a) and (27), we have

$$\begin{aligned} i \in N \quad &\Rightarrow \quad |x_i(\hat{v}_i - v_i)| = O(\mu)O(\mu + \mathbf{u}), \\ i \in B \quad &\Rightarrow \quad |x_i(\hat{v}_i - v_i)| = O(1)O(\mu(\mu + \mathbf{u})). \end{aligned}$$

A similar result holds for $|y_i(\hat{u}_i - u_i)|$. For the last term in (36), we have from (14), (27), and Theorem 3.2 that

$$\begin{aligned} i \in B \quad &\Rightarrow \quad |\hat{u}_i\hat{v}_i| \leq O(\mu + \mathbf{u})(|v_i| + |\hat{v}_i - v_i|) = O(\mu(\mu + \mathbf{u})) \\ i \in N \quad &\Rightarrow \quad |\hat{u}_i\hat{v}_i| = O(\mu)(|v_i| + |\hat{v}_i - v_i|) = O(\mu(\mu + \mathbf{u})). \end{aligned}$$

We also have $\mu/(x_i y_i) \leq 1/\gamma_{\min} = O(1)$. Using these estimates in (36), we obtain

$$\begin{aligned} (x_i + \alpha\hat{u}_i)(y_i + \alpha\hat{v}_i) &= x_i y_i(1-\alpha) + \alpha\sigma\mu + \alpha O(\mu(\mu + \mathbf{u})) \\ &= x_i y_i(1-\alpha) + \alpha\mu\left[\sigma + O(\mu + \mathbf{u})\right]. \end{aligned} \tag{37}$$

By summing over i, we obtain

$$(x + \alpha\hat{u})^T(y + \alpha\hat{v}) = x^T y\left[(1 - \alpha + \alpha\sigma) + \alpha O(\mu + \mathbf{u})\right], \tag{38}$$

which yields the desired result. ∎

We now examine the safe steps, for which $\sigma_k \geq \bar{\sigma}$, and show that $\alpha_k = 1$ satisfies the criteria (33) and (34) for large enough k, even when the computed search direction $(\hat{u}^k, \hat{v}^k)$ is used in place of the exact direction (u^k, v^k). That is, a unit step is taken.

THEOREM 4.3. *Suppose that $\bar{\sigma}$ is substantially larger than $\mathbf{u}$, in a sense to be defined below. Then for all sufficiently large k, if a safe step (with $\sigma_k \geq \bar{\sigma}$, $\gamma_{k+1} = \gamma_k$, and $\beta_k = 0$) is computed, the step length parameter satisfying (33) and (34) will be $\alpha_k = 1$.*

Proof. From (38), we have

$$\frac{(x + \alpha\hat{u})^T(y + \alpha\hat{v})}{x^T y} = (1 - \alpha) + \alpha\left[\sigma + O(\mu + \mathbf{u})\right]. \tag{39}$$

Therefore (34b) will hold for all $\alpha \in [0, 1]$ (with (u, v) replaced by $(\hat{u}, \hat{v})$ and $\beta_k = 0$), provided that the term in square brackets in (39) is nonnegative. But nonnegativity is guaranteed for $\mathbf{u} \ll \bar{\sigma}$ and μ sufficiently small, so $\alpha_k = 1$ satisfies this inequality.

Consider now (34a), with $\gamma_{k+1} = \gamma_k$. From (37) and (38), we have

$$\begin{aligned} \frac{(x_i + \alpha\hat{u}_i)(y_i + \alpha\hat{v}_i)}{(x + \alpha\hat{u})^T(y + \alpha\hat{v})/n} &= \frac{(1-\alpha)x_i y_i + \alpha\mu[\sigma + O(\mu + \mathbf{u})]}{(1-\alpha)\mu + \alpha\mu[\sigma + O(\mu + \mathbf{u})]} \\ &\geq \frac{x_i y_i/\mu + [\sigma - C_{10}(\mu + \mathbf{u})]\alpha/(1-\alpha)}{1 + [\sigma + C_{11}(\mu + \mathbf{u})]\alpha/(1-\alpha)}, \end{aligned} \tag{40}$$

for some positive constants C_{10} and C_{11}. Since $x_i y_i \geq \gamma_k\mu$ for all $i = 1, \ldots, n$, we find that (34a) is satisfied if

$$\frac{\sigma - C_{10}(\mu + \mathbf{u})}{\sigma + C_{11}(\mu + \mathbf{u})} \geq \gamma_{\max},$$

or, equivalently,

$$\frac{\mu + \mathbf{u}}{\sigma} \leq \frac{1 - \gamma_{\max}}{C_{10} + C_{11}}.$$

This last inequality, and therefore (34a), holds provided that μ and $\mathbf{u}$ are small enough with respect to $\bar{\sigma}$, as we have assumed.

Finally, we show that

$$\hat{\mu}(\alpha) \triangleq (x + \alpha\hat{u})^T(y + \alpha\hat{v})/n$$

is decreasing on the interval $\alpha \in [0, 1]$. Since from (38), we have

$$\frac{\hat{\mu}(\alpha)}{\mu} = 1 - \alpha + \alpha\sigma + \alpha O(\mu + \mathbf{u}),$$

the derivative $\hat{\mu}'(\alpha)$ is nonpositive provided that

$$-1 + \sigma + O(\mu + \mathbf{u}) \leq 0. \tag{41}$$

Since $\bar{\sigma} \leq \sigma \leq 1/2$ and, by our assumptions in the first part of this proof, $\bar{\sigma}$ dominates the $O(\mu + \mathbf{u})$ term, we have that (41) holds.

We have shown that all $\alpha \in [0, 1]$ satisfy the conditions (34) with (u, v) replaced by $(\hat{u}, \hat{v})$. Moreover, the function in (33) is decreasing over this interval. We conclude that $\alpha_k = 1$ is the step chosen by the line search procedure, giving the result. ∎

We turn now to fast step, for which $\sigma_k = 0$, $\beta_k = \bar{\gamma}^{t_k}$, and $\gamma_{k+1} = \gamma_{\min} + \bar{\gamma}^{t_k}(\gamma_{\max} - \gamma_{\min})$. The (exact) analysis in Wright [14] shows that fast steps are eventually always taken by the algorithm. Note that if the fast step is accepted, we have

$$\gamma_k - \gamma_{k+1} = (\bar{\gamma}^{t_k - 1} - \bar{\gamma}^{t_k})(\gamma_{\max} - \gamma_{\min}) = \bar{\gamma}^{t_k}(\bar{\gamma}^{-1} - 1)(\gamma_{\max} - \gamma_{\min}) = \Omega(\beta_k). \tag{42}$$

The following theorem gives an estimate for the length of a fast step.

THEOREM 4.4. *If a fast step is attempted at iteration k, where k is sufficiently large and $\mathbf{u} \ll 1$, then*

$$\alpha_k \geq \left[1 + \eta_k \frac{\mu_k + \mathbf{u}}{\bar{\gamma}^{t_k}}\right]^{-1},$$

where η_k satisfies $0 \leq \eta_k \leq O(1)$.

Proof. As in (37), we have for $\sigma_k = 0$ that

$$(x_i + \alpha\hat{u}_i)(y_i + \alpha\hat{v}_i) = x_i y_i(1 - \alpha) + \mu\alpha O(\mu + \mathbf{u}) \geq \gamma_k\mu(1 - \alpha) - \mu\alpha(\mu + \mathbf{u})\eta_k^{(1)}, \tag{43}$$

where $0 \leq \eta_k^{(1)} \leq O(1)$. From (38) we have

$$(x + \alpha\hat{u})^T(y + \alpha\hat{v})/n = \mu\left[1 - \alpha + \alpha O(\mu + \mathbf{u})\right] \leq \mu\left[1 - \alpha + \alpha(\mu + \mathbf{u})\eta_k^{(2)}\right], \tag{44}$$

where $0 \leq \eta_k^{(2)} \leq O(1)$. Putting (43) and (44) together, we deduce that (34a) holds provided that

$$\gamma_k(1 - \alpha) - \alpha(\mu + \mathbf{u})\eta_k^{(1)} \geq \gamma_{k+1}[1 - \alpha + \alpha(\mu + \mathbf{u})\eta_k^{(2)}],$$

which in turn is true if

$$(\gamma_k - \gamma_{k+1})(1 - \alpha) \geq \alpha(\mu + \mathbf{u})(\eta_k^{(1)} + \eta_k^{(2)}).$$

This last inequality is implied by the following bound on α:

$$\alpha \leq \left[1 + \frac{(\mu+\mathbf{u})(\eta_k^{(1)} + \eta_k^{(2)})}{\gamma_k - \gamma_{k+1}}\right]^{-1} = \left[1 + \frac{(\mu+\mathbf{u})(\eta_k^{(1)} + \eta_k^{(2)})}{\bar{\gamma}^{t_k}(\bar{\gamma}^{-1}-1)(\gamma_{\max}-\gamma_{\min})}\right]^{-1}, \tag{45}$$

where we have used (42) to derive the final inequality.

Using (44) again, we have

$$(x+\alpha\hat{u})^T(y+\alpha\hat{v})/n = \mu\left[1-\alpha+\alpha O(\mu+\mathbf{u})\right] \geq \mu\left[1-\alpha-\alpha(\mu+\mathbf{u})\eta_k^{(3)}\right],$$

where $0 \leq \eta_k^{(3)} \leq O(1)$, so the inequality (34b) is satisfied when

$$1-\alpha-\alpha(\mu+\mathbf{u})\eta_k^{(3)} \geq (1-\alpha)(1-\beta_k),$$

that is, when

$$\alpha \leq \left[1 + \frac{(\mu+\mathbf{u})\eta_k^{(3)}}{\beta_k}\right]^{-1} = \left[1 + \frac{(\mu+\mathbf{u})\eta_k^{(3)}}{\bar{\gamma}^{t_k}}\right]^{-1}. \tag{46}$$

Providing we can show that $\hat{\mu}_k(\alpha)$ is decreasing on $[0,1]$, we have from (45) and (46) that the result holds for η_k defined by

$$\eta_k = \max\left(\eta_k^{(3)}, \frac{\eta_k^{(1)}+\eta_k^{(2)}}{(\bar{\gamma}^{-1}-1)(\gamma_{\max}-\gamma_{\min})}\right).$$

However,

$$\hat{\mu}_k'(\alpha) \leq -\mu\left[1-O(\mu+\mathbf{u})\right] < 0,$$

so $\hat{\mu}_k(\alpha)$ is certainly decreasing on $[0,1]$, and the result is proved. ∎

This result accurately indicates the behavior of fast steps on later iterations of the algorithm. The quantity η_k is typically either extremely small or else quite significant (that is, $\eta_k = \Omega(1)$), depending on the sign of certain products such as $\hat{u}^T\hat{v}$, $\hat{u}_i\hat{v}_i$, and so on. When η_k is tiny and $\mu_k + \mathbf{u} \ll \bar{\gamma}^{t_k}$, the value of α_k is very close to 1, and the fast step is accepted with a large reduction in μ. When η_k is larger, or when $\bar{\gamma}^{t_k} = O(\mathbf{u})$, the fast step may not lead to a very large decrease in μ and may even be rejected in favor of a safe step.

We summarize the results of this section in the following theorem.

THEOREM 4.5. *Suppose that* $\mathbf{u}$ *is much smaller than* $\bar{\sigma}$*, in the sense of Theorem 4.3. Then for all sufficiently large* k *we have that either*

(i) a fast step is taken, and

$$\mu_{k+1} \leq \rho\mu_k, \tag{47}$$

with

$$\mu_{k+1} = O\left(\frac{\mu_k+\mathbf{u}}{\bar{\gamma}^{t_k}}\right)\mu_k, \tag{48}$$

or

(ii) a safe step is taken, with

$$\mu_{k+1} = [\sigma_k + O(\mu_k+\mathbf{u})]\mu_k. \tag{49}$$

Proof. The condition for fast-step acceptance yields (47). The estimate (48) follows from Theorems 4.2 and 4.4 and the identity $\sigma_k = 0$. The safe-step estimate (49) follows from Theorem 4.2 when we use $\alpha_k = 1$. ∎

5. Computational Results. The algorithm of Section 4 was implemented in double-precision Fortran, using the LAPACK routines dgetrf and dgetrs to solve the linear system (6a). Our test problems are of two types.

(i) The matrix M has the form $M = ADA^T$, where A is $n \times n_r$ dense with all elements drawn from a uniform distribution on $[-1, 1]$, while D is diagonal with diagonal elements of the form 10^τ, where τ is drawn from a uniform distribution on $[0, 1]$. Since we choose $n_r \leq n$, the rank of M is n_r. (Rank-deficiency of M is not an artificial feature; in certain applications of (1), including (3), M is structurally rank-deficient.) The solutions x^* and y^* are chosen so that the even-numbered components of x^* and the odd-numbered components of y^* are zero, while the remaining components are uniform on $[0, 1]$.

(ii) The matrix M has the form (3a), where the matrix A is dense with elements of the form $\tau_1 10^{\tau_2}$, where τ_1 and τ_2 are drawn from uniform distributions on $[-\frac{1}{2}, \frac{1}{2}]$ and $[0, 1]$, respectively. If p and m denote the dimensions of of z and λ, respectively, we choose the even-numbered components $2, 4, \ldots, min(p, m)$ of both z^* and λ^* to be nonzero. (It is a consequence of nondegeneracy of the solution of (2) that the same number of components of z^* and λ^* be nonzero. This requirement is also necessary for nonsingularity of M_{BB}.) The nonzero components of z^*, λ^* and the complementary vector pair in (3) are all drawn from $[0, 1]$.

We use the following values for the constants:

$$\gamma_{\min} = 10^{-5}, \qquad \gamma_{\max} = 10^{-2}, \qquad \rho = \bar{\gamma}/2, \qquad \bar{\gamma} = 10^{-1}. \tag{50}$$

We choose the centering parameter σ_k at each safe iteration according to the formula

$$\sigma_k = \text{mid}(\bar{\sigma}, \mu_k/\sqrt{n}, .2), \qquad \text{where } \bar{\sigma} = .01. \tag{51}$$

Though we made no special effort to tune these constants to their optimal values, our experience indicates that the choices (50), (51) are efficient for these and other types of problems.

In Tables 1–4 we tabulate the behavior of the algorithm of Section 4. Many of the uninteresting safe iterates are omitted. An asterisk in the last column indicates that a fast step was taken from this iterate. We terminate the algorithm when μ falls below 10^{-20}.

These tables indicate rapid convergence of the algorithm during its final stages. Typically, the algorithm takes only fast steps after it has decreased μ below a certain threshold. (The experience of the author and others indicates that this threshold is quite small for linear and quadratic problems, so that superlinear convergence does not set in until quite late in the process. Preliminary experience with nonlinear problems indicates that fast steps are typically taken at an earlier stage, that is, the threshold is not so small.)

The behavior observed in Tables 1–4 certainly confirms the efficacy of Gaussian elimination with partial pivoting in the context of this interior-point method. The linear algebra continues to produce good steps even when μ is extremely small. The convergence of μ to zero appears to be superlinear in each case (even quadratic, in the case of Table 4). These tables do not, however, show the asymptotic behavior suggested by Theorem 4.5. To see it, we must continue to run the algorithm past the point of convergence. Table 5 shows what happens when we continue to iterate on the problem of Table 3 until μ is reduced below 10^{-100}. (The late asymptotic

TABLE 1
Convergence of the algorithm: Problem type (i), $n = n_r = 20$

k	$\log_{10} \mu_k$	$\log_{10} \|r^k\|_1$	Fast step?
0	4.0	4.7	
1	3.7	4.3	
2	3.2	3.3	
3	2.7	-11.7	
⋮	⋮	⋮	
15	-3.7	-13.4	
16	-4.8	-13.3	
17	-5.7	-13.2	*
18	-7.0	-13.1	*
19	-9.5	-13.2	*
20	-14.5	-13.5	*
21	-23.8	-13.4	terminate

TABLE 2
Convergence of the algorithm: Problem type (i), $n = n_r = 100$

k	$\log_{10} \mu_k$	$\log_{10} \|r^k\|_1$	Fast step?
0	4.9	6.5	
1	4.6	6.2	
2	4.1	5.5	
3	3.8	4.9	
4	3.3	-9.8	
⋮	⋮	⋮	
21	-5.8	-11.6	
22	-6.7	-11.6	
23	-7.8	-11.6	
24	-9.0	-11.6	*
25	-10.9	-11.5	*
26	-14.7	-11.6	*
27	-22.1	-11.8	terminate

TABLE 3
Convergence of the algorithm: Problem type (i), $n = 100$, $n_r = 60$.

k	$\log_{10} \mu_k$	$\log_{10} \|r^k\|_1$	Fast step?
0	4.5	6.1	
1	4.3	5.8	
2	4.1	5.5	
3	3.7	5.0	
4	3.2	3.9	
5	2.8	-9.5	
⋮	⋮	⋮	
24	-2.8	-11.8	
25	-3.5	-11.7	
26	-4.4	-11.7	
27	-5.6	-11.7	
28	-6.7	-11.7	*
29	-8.1	-11.7	*
30	-10.1	-11.8	*
31	-13.8	-11.8	*
32	-20.8	-11.8	terminate

TABLE 4
Convergence of the algorithm: Problem type (ii), $n = 200$, *matrix* A *is* 40×160

k	$\log_{10} \mu_k$	$\log_{10} \|r^k\|_1$	Fast step?
0	4.0	5.2	
1	3.8	4.9	
2	3.5	4.7	
3	3.1	4.2	
4	2.1	3.1	
5	1.7	2.6	
6	1.4	2.2	
7	1.0	1.7	
8	0.5	1.2	
9	0.2	0.8	
10	-0.7	-0.3	
11	-2.1	-1.8	*
12	-4.4	-4.1	*
13	-8.3	-7.9	*
14	-15.6	-10.2	*
15	-28.2	-10.2	terminate

TABLE 5
Later iterates on the problem of Table 3

k	$\log_{10}\mu_k$	$\log_{10}\|r^k\|_1$	Fast step?
⋮	⋮	⋮	
32	-20.8	-11.8	*
33	-31.4	-11.9	*
34	-42.1	-11.9	*
35	-46.8	-11.9	*
36	-50.9	-11.8	*
37	-61.7	-11.9	*
38	-71.9	-11.9	
39	-73.9	-11.9	*
40	-78.4	-11.9	*
41	-90.5	-11.8	*
42	-102.4	-11.9	terminate

convergence was qualitatively similar on all the problems we tried, so we report just this one instance.) Note that fast steps are taken on each iteration with decrease factors between 10^{-4} and 10^{-12}, except for one iteration — the 38th — on which a safe step is taken with a decrease ratio of almost exactly $\sigma_k = 10^{-2}$. The existence of these two kinds of steps and their effects on μ_k are in close accord with the predictions of Theorem 4.5.

Note that in all the tables the residual norm $\|r_k\|$ decreases to $O(\mathbf{u})$ but no further. As discussed in the proof of Theorem 3.2, this behavior is due to roundoff error in the calculation of r^k via the formula $r^k = y^k - Mx^k - q$.

We experimented with a version of the code in which a modified complete pivoting strategy was used for solving (6a). The columns of the coefficient matrix were ordered by decreasing value of $\|\cdot\|_\infty$ before Gaussian elimination with partial pivoting was applied. Asymptotically, this strategy has the effect of ordering the nonbasic columns first, so the analysis at the end of Section 3 still applies. As predicted in that analysis, this version of the algorithm behaves only slightly differently from the partial pivoting version described above.

The assumption that M_{BB} is nonsingular (indeed, well conditioned) plays an important role in the analysis of Sections 3 and 4. Theoretically, the algorithm of Section 4 is known to have fast local convergence even when M_{BB} is singular and the solution is not unique. We tested to see whether fast convergence was attainable in practice by forming a problem from the class (i) with $n_r = 25$. Since B contains 50 indices, the submatrix M_{BB} is certainly rank deficient. The result of this run is summarized in Table 6. It is clear that the behavior indicated in Theorem 4.5 does not occur. After taking two fast steps and converging to $\mu_k \approx 10^{-7}$ by iteration 14, the algorithm stalls and makes very little progress from that point on. This and other similar examples suggest that the assumption of M_{BB} nonsingular probably cannot be relaxed.

Acknowledgement. I thank the editor and referees for their insightful comments on an earlier draft.

TABLE 6
Convergence in the case of M_{BB} rank-deficient: Problem type (i), $n = 100$, $n_r = 25$

k	$\log_{10}\mu_k$	$\log_{10}\|r^k\|_1$	Fast step?
0	4.2	5.9	
1	4.0	5.5	
2	3.6	5.1	
3	2.7	4.2	
4	2.1	3.4	
⋮	⋮	⋮	
11	-0.2	0.2	
12	-1.2	-0.9	*
13	-3.7	-3.5	*
14	-7.3	-7.0	
15	-7.4	-7.1	
16	-7.4	-7.2	
17	-7.6	-7.4	
⋮	⋮	⋮	
98	-8.4	-8.2	
99	-8.5	-8.2	
100	-8.5	-8.3	
⋮	⋮	⋮	

REFERENCES

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

[2] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, 2nd ed., 1989.

[3] N. J. HIGHAM, *Accuracy and Stability of Numerical Algorithms (provisional title)*, 1994. In preparation.

[4] J. JI, F. A. POTRA, AND S. HUANG, *A predictor-corrector method for linear complementarity problems with polynomial complexity and superlinear convergence*, Technical Report 18, Department of Mathematics, University of Iowa, Iowa City, Iowa, August 1991.

[5] M. KOJIMA, Y. KURITA, AND S. MIZUNO, *Large–step interior point algorithms for linear complementarity problems*, SIAM Journal on Optimization, 3 (1993), pp. 398–412.

[6] M. KOJIMA, S. MIZUNO, AND A. YOSHISE, *An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems*, Mathematical Programming, 50 (1991), pp. 331–342.

[7] I. J. LUSTIG, R. E. MARSTEN, AND D. F. SHANNO, *Computational experience with a primal-dual interior point method for linear programming*, Linear Algebra and Its Applications, 152 (1991), pp. 191–222.

[8] ———, *Computational experience with a globally convergent primal-dual predictor-corrector algorithm for linear programming*, Technical Report SOR 92–10, Program in Statistics and Operations Research, Princeton University, Princeton, N. J., 1992.

[9] S. MEHROTRA, *On the implementation of a primal-dual interior point method*, SIAM Journal on Optimization, 2 (1992), pp. 575–601.

[10] R. D. C. MONTEIRO AND I. ADLER, *Interior path-following primal-dual algorithms. part II: Convex quadratic programming*, Mathematical Programming, 44 (1989), pp. 43–66.

[11] D. B. PONCELEÓN, *Barrier methods for large-scale quadratic programming*, PhD thesis, Stanford University, 1990.

[12] F. A. POTRA, *An $o(nl)$ infeasible-interior-point algorithm for LCP with quadratic convergence*, Report on Computational Mathematics 50, Department of Mathematics, University of Iowa, Iowa City, Iowa, January 1994.

[13] R. J. Vanderbei, *LOQO User's Manual*, Technical Report SOR 92–5, Program in Statistics and Operations Research, Princeton University, Princeton, N. J., 1992.

[14] S. J. Wright, *A path-following interior-point algorithm for linear and quadratic optimization problems*, Preprint MCS–P401–1293, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., December 1993.

[15] ———, *An infeasible-interior-point algorithm for linear complementarity problems*, Mathematical Programming, 67 (1994), pp. 29–52.

[16] ———, *Stability of linear algebra computations in interior-point methods for linear programming*, Preprint MCS–P446–0694, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Ill., June 1994.

[17] X. Xu, P. Hung, and Y. Ye, *A simplified homogeneous and self-dual linear programming algorithm and its implementation.* Manuscript, September 1993.

[18] Y. Ye and K. Anstreicher, *On quadratic and $O(\sqrt{n}L)$ convergence of a predictor-corrector algorithm for LCP*, Mathematical Programming, Series A, 62 (1993), pp. 537–551.

[19] Y. Zhang, *On the convergence of a class of infeasible-interior-point methods for the horizontal linear complementarity problem*, SIAM Journal on Optimization, 4 (1994), pp. 208–227.