TAMKANG JOURNAL OF MATHEMATICS
Volume 25, Number 1, Spring 1994

A METHOD FOR SOLVING LEAST-SQUARES PROBLEMS ARISING FROM ANGULAR LINEAR PROGRAMS[1]

EUGENE K. YANG AND CHIA-HSIANG CHOU

Abstract. The most costly part of interior point methods for solving linear programming problems is in solving least squares subproblems. If the normal equation matrix of a least-squares problem is not nearly singular, it is well known that LDL^t decomposition is a stable method. However, for the nearly singular case, it can cause numerical difficulties. In this paper, we consider the linear proogram whose constraint matrix B is large, sparse, and with angular structure. We assume that the normal equation matrices arising from such a linear program may be nearly singular. We present a numerically stable block method utilizing LDL^t decomposition with diagonal pivoting for solving such normal equations. Although the method of the diagonal pivoting is old, this paper presents new results when the method is applied to the positive definite but nearly singular case.

1. Introduction

In this paper, we consider the interior (point) methods for solving large angular linear programs, i.e.,

$$\begin{aligned} &\max \quad c^t y \\ &\text{subject to} \\ &\text{and} \quad \begin{cases} By = b, \\ y \geq 0, \end{cases} \end{aligned} \tag{1}$$

where the $m_B \times n_B$ constraint matrix B is of the special from:

$$B = \begin{bmatrix} B_1 & & & & \\ & B_1 & & & \\ & & \ddots & & \\ & & & B_p & \\ F_1 & F_2 & \cdots & F_p & F_{p+1} \end{bmatrix}$$

Received January 23, 1991

[1]This research was supported by National Science Council of the R.O.C. under the grant NSC79-0208M007-76.

Keywords: least squares, angular linear program, nearly singularity, rank revealing.

B is said to be a (block-)angular matrix. B_i is $m_i \times n_i$, $i = 1, \cdots, p$ and F_i is $m_F \times n_i$, $i = 1, \cdots, p+1$. We shall assume that all blocks B_i and F_i are dense (or small) and m_i and m_F are much smaller than m_B and n_B; the effect of this assumption on the efficiency of our algorithm will be discussed at the end of Section 2. Applications of such large scale programs include multi-period manufacturing, allocation problems, two-stage stochastic programming, and various problems in planning and control (see Rosen and Maier [9] for more references). Tomlin [10] notes that the angular linear program (1) is more efficiently solved by interior methods than by the simplex method. Hence, we only consider interior methods for solving (1).

It is well known that, in most of the interior methods for solving large sparse linear program, the time required to perform an interior point iteration is dominated by the solution of a sparse linear system, which is embedded in the solution of a least-squares problem; see Gill et al. [5]. They observe that the efficiency of the interior methods depends critically on fast, stable techniques for solving large-scale least-squares subproblems. In most interior methods the least-squares problem is solved by finding the solution of its normal equation. Interior point methods for solving (1) are iterative methods. Let $y = (y_1, \ldots, y_n)$ be the current interior point iterate in solving (1). The normal equation of the least-squares problem in the barrier method of Gill et al. has the following form:

$$BY^2B^t x = BY^2c, \tag{2}$$

where Y =diag$(y_1, \ldots, y_n)$ is a diagonal matrix and x is the solution to this normal equation; see Gill et al. [5] for further detail. All other interior methods have similar forms. Let $M = BY^2B^t$ and $f = BY^2c$. Then the normal equation matrix

$$M = \begin{bmatrix} A_1 & & & & C_1^t \\ & A_2 & & & C_2^t \\ & & \ddots & & \vdots \\ & & & A_p & C_p^t \\ C_1 & C_2 & \cdots & C_p & C_{p+1} \end{bmatrix}$$

has a bordered angular structure. We assume that M and possibly some A_i are either singular or nearly singular (by "nearly singular" we mean that there is at least one nearly zero eigenvalue regardless whether there are other exactly zero eigenvalues). It is well known that the singularity of M and A_i comes from the degeneracy of the linear program (1) (see Gill et al. [5] or Hooker [7]) and that almost all large, real-world linear programs are degenerate (see Tomlin [10]). In the interior method, if M is (nearly) singular, equation (2) is known to be consistent and there exist solutions to (2); see Hooker [7]. However, it requires a numerically stable method for solving (2).

Since each diagonal block A_i of M is either symmetric positive definite (SPD) or symmetric positive semi-definite (SPSD), it is possible to improve the efficiency by using LDL^t factorization (i.e., symmetric Gaussian elimination or symmetric triangular factorization) instead of LU factorization for each A_i; see Golub and Van Loan [6, pp. 82-90].

In order to preserve the block structure of M, in this paper we propose a block method which uses LDL^t factorization with diagonal pivoting for each block A_i such that the zero or "nearly zero" pivots are produced at the end of the diagonal factor D of each singular A_i. Here, "nearly zero" means less than $\|A_i\| * eps$, where eps is the machine epsilon (machine precision). The nearly zero pivots are critical in solving (2) stably; see Chan [4] for more detail. We will use capital letters to denote matrices and the corresponding small letters for their elements.

The remaining of this paper is organized as follows. In Section 2, a numerical-rank revealing algorithm utilizing LDL^t factorization with diagonal pivoting for nearly singular A_i is presented. In Section 3, we extend the LDL^t algorithm in Section 2 to factor the block-structured matrix M. We conclude in Section 4.

2. A diagonal pivoting algorithm

Theorem 1. *Suppose that an order n matrix* $A = \begin{bmatrix} \alpha & \nu^t \\ \nu & B \end{bmatrix} \begin{matrix} 1 \\ n-1 \end{matrix}$ *is SPSD with rank $r \leq n$ and $\alpha > 0$. Let the following factors of A be obtained after one step of symmetric Gaussian transformation:*

$$A = \begin{bmatrix} 1 & 0 \\ \frac{\nu}{\alpha} & I_{n-1} \end{bmatrix} \begin{bmatrix} \alpha & 0 \\ 0 & B - \frac{\nu\nu^t}{\alpha} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{\nu}{\alpha} & I_{n-1} \end{bmatrix}^t. \tag{3}$$

Then the submatrix $B - \frac{\nu\nu^t}{\alpha}$ is also SPSD and has the same nullity as A does.

Proof. Let $L = \begin{bmatrix} 1 & 0 \\ \frac{\nu}{\alpha} & I_{n-1} \end{bmatrix}$, which is nonsingular. By Sylvester's Inertia Theorem [8, p.10] on the congruence transformation,

$$L^{-1}AL^{-t} = \begin{bmatrix} \alpha & 0 \\ 0 & B - \frac{\nu\nu^t}{\alpha} \end{bmatrix}$$

has the same inertia as A has and thus is also SPSD. Since $\alpha > 0$, the results follow.

In the following we present an algorithm for solving a consistent system

$$Az = g, \tag{4}$$

where A is $n \times n$, SPSD, and (nearly) singular.

Algorithm DP (Diagonal Pivoting) for factoring a dense matrix A. Set the tolerance *tol* for stopping criterion. (Usually we set $tol \simeq \|A\| * eps$.) Suppose that $A \in \mathbf{R}^{n\times n}$ is SPSD, and for some $k < n$, we have determined the Gaussian transformations $M_1, \ldots, M_{k-1} \in \mathbf{R}^{n\times n}$ (the repetitions of the decomposition (3)) and permutation

matrices $P_1, \ldots, P_{k-1} \in \mathbf{R}^{n \times n}$ such that

$$\begin{aligned} A^{(k-1)} &= M_{k-1} P_{k-1} \ldots M_1 P_1 A P_1^t M_1^t \ldots P_{k-1}^t M_{k-1}^t \\ &= \begin{bmatrix} A_{11}^{(k-1)} & 0 \\ 0 & A_{22}^{(k-1)} \end{bmatrix} \end{aligned} \tag{5}$$

($A^{(0)}$ is defined to be A), where $A_{11}^{(k-1)}$ is a $(k-1) \times (k-1)$ diagonal matrix and

$$A_{22}^{(k-1)} = \begin{bmatrix} a_{kk}^{(k-1)} & \cdots & a_{kn}^{(k-1)} \\ \vdots & \ddots & \vdots \\ a_{kn}^{(k-1)} & \cdots & a_{nn}^{(k-1)} \end{bmatrix}.$$

Herein, $|F| < tol$ is used to denote $f_{ij} < tol$ for every i, j. Next, $A^{(k)}$ is determined as follows:

(i) Determine $\mu^{(k-1)} = a_{pp}^{(k-1)} = \max_{k \le i \le n} a_{ii}^{(k-1)}$, the maximum of the diagonal elements of $A_{22}^{(k-1)}$, which are all *nonnegative* since $A_{22}^{(k-1)}$ is SPSD; see the proof of Theorem 2. If $\mu^{(k-1)} < tol$, (which implies $|A_{22}^{(k-1)}| < tol$ by Theorem 2,) then set $A_{22}^{(k-1)} = O$, assign $D = A^{(k-1)}$, and stop (with the *numerical rank* of A equal to $k-1$; see the definition described before Theorem 2).

(ii) Permute the kth and pth rows and the kth and pth columns of $A_{22}^{(k-1)}$ to obtain the permuted matrix $P_k A^{(k-1)} P_k^t$.

(iii) Use the new maximal $a_{kk}^{(k-1)}$ as the pivot to obtain $A^{(k)} = M_k (P_k A^{(k-1)} P_k^t)\, M_k^t$ by one step of symmetric Gaussian transformation (a repetition of the decomposition (3)). Increase k by 1 and return to (i).

It can be shown that after repeating the above step k decomposition for at most $n-1$ times, the LDL^t decomposition for PAP^t will be completed, where L is $n \times n$ and unit lower-triangular, P is a permutation matrix, and diagonal matrix D will be shown to reveal the rank of A later. Once this decomposition is produced by Algorithm DP, we have done the dominant work for solving (4); the remaining forward and backward solves take only additional $O(n^2)$ flops. Hence, in the rest of this paper we will only discuss the dominant phase of decomposing A (after permutation) into triangular and diagonal factors.

Bunch and Kaufman [2] present several algorithms for solving symmetric indefinite systems. One of their algorithms, Algorithm C, when applied to an SPD matrix, is exactly the same as Algorithm DP presented above. (For the convenience of making a comparison, we use the name Algorithm DP whenever we refer to the context of this paper, i.e., for solving SPSD case.) However, we note that Algorithm DP and the algorithms of Bunch and Kaufman [2] have different purposes. Their algorithms and those proposed in other related papers (e.g., Bunch and Parlett [3]) are designed for solving a linear system with an indefinite but *nonsingular* (by this we mean not nearly singular) matrix. Bunch and Kaufman [2] do not discuss the concept of the *numerical*

rank of a matrix, which is defined to be its number of singular values (or eigenvalues for the SPSD case) that have absolute values larger than the tolerance *tol*. In other words, the numerical rank is the number of singular values that are not nearly zero. On the other hand, Algorithm DP is designed for solving a system with SPSD and *nearly singular* matrix. The next theorem shows in the SPSD case that the diagonal pivoting is equivalent to the complete pivoting.

Theorem 2. *Let A be $n \times n$ and SPSD. Then the diagonal pivoting strategy used in Algorithm DP is equivalent to the complete pivoting.*

Proof. Let k be the step count of Algorithm DP (k is from 1 to n) and let $a_{pp}^{(k-1)}$ denote $\max_{k \le q \le n} a_{pp}^{(k-1)}$. Since A is SPSD, $a_{ii}a_{jj} - a_{ij}^2 \ge 0$, for every $1 \le i, j \le n$. Then, it is easy to see that following formula is true for $k = 1$.

$$\left[a_{pp}^{(k-1)}\right]^2 = \left[\max_{k \le q \le n} a_{qq}^{(k-1)}\right]^2 \ge a_{ii}^{(k-1)} a_{jj}^{(k-1)} \ge \left[a_{ij}^{(k-1)}\right]^2$$

for every i and j such that $k \le i,\ j \le n$,

or,

$$a_{pp}^{(k-1)} \ge |a_{ij}^{(k-1)}| \text{ for every } i \text{ and } j \text{ such that } k \le i,\ j \le n. \tag{6}$$

By repeatedly applying Theorem 1, $A_{22}^{(k-1)}$ at the step $k \ge 1$ (see formula (5)) is also SPSD. Thus inequality (6) is also true for every k such that $1 \le k \le n$ and the conclusion follows.

The Bunch-Kaufman algorithm requires $n^3/6$ flops, $O(n^2)$ comparisons for partial pivoting, and $n^2/2$ storage (compared with that the Gaussian elimination requires $n^3/3$ flops, $O(n^2)$ comparisons for pivoting, and n^2 storage.)Since Algorithm DP is the Bunch-Kaufman algorithm used to suit nearly singular case, it is easy to show (see also Golub and Van Loan [6, pp. 84-85]) that it also requires only $n^3/6$ flops, $n^2/2$ comparisons for diagonal pivoting, and $n^2/2$ storage for solving (4). Since the Bunch-Kaufman alforithm is designed for indefinite matrices, the bound on element growth can only be shown to be $(2.57)^{n-1}$, a result similar to LU with partial pivoting; see Bunch and Kaufman [2]. (A completepivoting variant by Bunch [1] is more stable but unfortunately at a cost of $O(n^3)$ comparisons.) For the SPSD case, the diagonal pivoting takes only $O(n^2)$ comparisons but is equivalent to the complete pivoting; thus, we can apply the bound on element growth established by Wilkinson [11, pp. 213-214] to Algorithm DP. The bound is $n^{1/2}f(n)$, where

$$f(n) = \left(\prod_{k=2}^{n} k^{1/(k-1)}\right)^{1/2} < 1.8n^{(1/4)\log(n)}.$$

In fact, the element growth factor can be shown in following to be one (i.e., there is no growth) for the SPSD case. The next theorem establishes that the elements of D produced by Algorithm DP are in decreasing order.

Theorem 3. *If $A \in \mathbf{R}^{n\times n}$ is SPSD with rank $(A) = r \leq n$, then Algorithm DP will produce a diagonal matrix $D = diag(d_1, \ldots, d_n)$ whose elements are in decreasing order. Moreover, if A is exactly singular with $r < n$, the zero diagonal elements are forced to the end of D (i.e., $d_{r+1} = d_{r+2} = \ldots = d_n = 0$).*

Proof. The part that $d_{r+1} = d_{r+2} = \ldots = d_n = 0$ is easy to show. Since, by Theorem 2, Algorithm DP uses a complete pivoting strategy for Gaussian elimination, it is well known that $A_{22}^{(r)} = O$ and $a_{kk}^{(k-1)} \neq 0$ for $k \leq r$.

For $k \leq r$ and at the substep (iii) of Algorithm DP, $A_{22}^{(k-1)}$ is SPD (with rank > 0) and thus the maximal pivot $a_{kk}^{(k-1)} > 0$. Hence, $0 \leq a_{ii}^{(k)} = a_{ii}^{(k-1)} - [a_{ki}^{(k-1)}]^2/a_{kk}^{(k-1)} \leq a_{ii}^{(k-1)} \leq a_{kk}^{(k-1)}$, for $k = 1, \cdots, r$ and $k+1 \leq i \leq n$, where the last inequality is implied by the diagonal pivoting at the step (i) of Algorthm DP. By $d_k = a_{kk}^{(k-1)}$ and by applying the above inequality recursively (i.e., $a_{11}^{(0)} \geq a_{22}^{(1)}$, $a_{22}^{(1)} \geq a_{33}^{(2)}$, etc.), we have $d_1 \geq d_2 \geq \ldots \geq d_r > d_{r+1} = d_{r+2} = \ldots = d_n = 0$.

By Theorem 3, d_1 (which is $a_{11}^{(0)}$ after pivoting) is the maximal element of both D and A. Then by $|L| \leq 1$, the growth factor is one. In the remaining of this section, we comment on the *numerical-rank* revealing feature (i.e., the ability to find nearly zero pivots) of Algorithm DP, which is critical in stably solving (4) and (2). For a discussion of rank-revealing LU factorization, see Chan [4]. We can only analyze the case when the numerical rank of A (i.e., the number of eigenvalues greater than *tol*) is $n-1$. For complete pivoting, both L and L^{-1} have norms of order unity (see [11, pp. 364-365]). Let λ_n denote the smallest eigenvalue of A. From $A^{-1} = L^{-t}D^{-1}L^{-1}$, we have $\lambda_n^{-1} = \|A^{-1}\|_2 \leq \|L^{-1}\|_2^2 \cdot \|D^{-1}\|_2 \simeq d_n^{-1}$. Similarly, it can be shown that $d_n^{-1} \lesssim \lambda_n^{-1}$ and thus $O(d_n) \simeq O(\lambda_n)$, or, the smallest pivot of D has the same order of magnitude as the smallest eigenvalue of A. This shows that, when the numerical rank of A is $n-1$, Algorithm DP reveals the numerical rank of A by the result that the last element of D is nearly zero. Furthermore, the numerical tests given below confirm this rank-revealing property even if the numerical rank of A is less than $n-1$.

In the following, we first present the numerical results of Algorithm DP on factoring Hilbert matrix A, (i.e., $a_{ij} = 1/(i+j-1)$ for $1 \leq i, j \leq n$,) which is a well-known ill-conditioned SPD matrix. All of numerical tests were performed on an IBM-compatible 386 PC. Algorithm DP was compiled on a Microsoft Fortran 4.1 compiler with double precision calculation. The standard software package MATLAB was used for computing eigenvalues. We tested on two Hilbert matrices of the order $n = 15$ and $n = 20$, respectively. When the Cholesky method without pivoting was utilized to factor these Hilbert matrices, the method failed because a negative pivot was produced at the 14^{th} step for both matrices. Algorithm DP successfully runs to the completion for these two Hilbert matrices as well as all the other nearly singular matrices that we have tested. Table 1 given below shows the eigenvalues λ_i's of the Hilbert matrices and the diagonal elements d_i's of D produced by Algorithm DP with the tolerance *tol* set to 1.0E-13. (The machine epsilon of 386 PC is approximately 1.0E-15.) We observe that the numerical ranks of A

and D are exactly the same for both matrices. In other words, the numerical rank of A is revealed in the LDL^t factors and thus (4) can be solved stably; see Chan [4] for other applications of the rank-revealing factorization.

$n = 15$			$n = 20$		
i	λ_i	d_i	i	λ_i	d_i
1	1.84E+00	1.00E+00	1	1.90E+00	1.00E+00
2	4.26E−01	8.89E−02	2	4.87E−01	8.89E−02
3	5.72E−02	1.51E−02	3	7.55E−02	1.51E−02
4	5.63E−03	3.22E−03	4	8.96E−03	3.22E−03
5	4.36E−04	4.38E−04	5	8.67E−04	4.86E−04
6	2.71E−05	1.13E−05	6	7.03E−05	1.37E−04
7	1.36E−06	9.69E−07	7	4.83E−06	2.70E−06
8	5.52E−08	2.79E−07	8	2.82E−07	3.02E−07
9	1.80E−09	1.32E−09	9	1.41E−08	1.45E−08
10	4.65E−11	7.02E−11	10	6.03E−10	6.06E−10
11	9.32E−13	1.28E−12	11	2.19E−11	1.22E−11
12	1.39E−14	0.00E+00	12	6.74E−13	4.55E−13
13	1.44E−16	0.00E+00	13	1.73E−14	0.00E+00
14	1.02E−17	0.00E+00	14	3.77E−16	0.00E+00
15	5.61E−18	0.00E+00	15	1.42E−17	0.00E+00
			16	1.02E−17	0.00E+00
			17	6.83E−18	0.00E+00
			18	4.86E−18	0.00E+00
			19	4.86E−18	0.00E+00
			20	2.36E−18	0.00E+00

Table 1. Hilbert matrices

Next, we present the numerical results of two more examples. Algorithm DP again reveals the numerical ranks. Chan [4] gives a well-known nearly singular (but not SPD) matrix with only one nearly zero singular value. Let H denote such a matrix with the order $n = 20$ and let $T = HH^t$. Obviously, T is SPD with only one nearly zero eigenvalue. The resulting T is as follows:

$$T = \begin{bmatrix} 20 & 17 & 16 & \cdots & 2 & 1 & 0 & -1 \\ 17 & 19 & 16 & \cdots & 2 & 1 & 0 & -1 \\ 16 & 16 & 18 & \cdots & 2 & 1 & 0 & -1 \\ \vdots & & & \ddots & \vdots & \vdots & \vdots & \vdots \\ 2 & 2 & 2 & \cdots & 4 & 1 & 0 & -1 \\ 1 & 1 & 1 & \cdots & 1 & 3 & 0 & -1 \\ 0 & 0 & 0 & \cdots & 0 & 0 & 2 & -1 \\ -1 & -1 & -1 & \cdots & -1 & -1 & -1 & 1 \end{bmatrix}.$$

After applying Algorithm DP to factor T, we obtain the following diagonal elements d_i of matrix D:

i	d_i	i	d_i
1	.200000000E+02	11	.311683457E+01
2	.700000000E+01	12	.310242423E+01
3	.491428571E+01	13	.306097830E+01
4	.486046512E+01	14	.301479326E+01
5	.385645933E+01	15	.300366986E+01
6	.385607940E+01	16	.300004864E+01
7	.367310167E+01	17	.300000000E+01
8	.362158374E+01	18	.266666667E+01
9	.328115325E+01	19	.200000000E+01
10	.328115233E+01	20	.109139364E−10

The last pivot d_n is indeed close to λ_n, which is 8.18E−12. Chan [4] also describes another matrix with one nearly zero singular value, which is first given by Wilkinson [11]. Let W denote such a matrix with the order $n = 21$ and again let $S = WW^t$. Then, the resulting S is as follows:

$$S = \begin{bmatrix}
101 & 19 & 1 & & & & & & & & & & & \\
19 & 83 & 17 & 1 & & & & & & & & & & \\
1 & 17 & 66 & 15 & 1 & & & & & & & & & \\
 & 1 & 15 & 51 & 13 & 1 & & & & & & & & \\
 & & 1 & 13 & 38 & 11 & 1 & & & & & & & \\
 & & & 1 & 11 & 27 & 9 & 1 & & & & & & \\
 & & & & 1 & 9 & 18 & 7 & 1 & & & & & \\
 & & & & & 1 & 7 & 11 & 5 & 1 & & & & \\
 & & & & & & 1 & 5 & 6 & 3 & 1 & & & \\
 & & & & & & & 1 & 3 & 3 & 1 & 1 & & \\
 & & & & & & & & 1 & 1 & 2 & −1 & 1 & \\
 & & & & & & & & & $\ddots$ & $\ddots$ & $\ddots$ & & \\
 & & & & & & & & & & & 1 & −19 & 101
\end{bmatrix}.$$

After using Algorithm DP to factor S, we obtain the following diagonal elements of D:

i	d_i	i	d_i
1	.101000000E+03	12	.236461619E+02
2	.101000000E+03	13	.147778892E+02
3	.794257426E+02	14	.147778892E+02
4	.794257426E+02	15	.798094132E+01
5	.624315632E+02	16	.798094132E+01
6	.624315632E+02	17	.333698653E+01
7	.474844577E+02	18	.333698653E+01
8	.474844577E+02	19	.140065685E+01
9	.345534406E+02	20	.105277241E+01
10	.345534406E+02	21	.444089210E−15
11	.236461619E+02		

d_n is again close to λ_n, which is 6.06E−16.

In summary, Algorithm DP (for the SPSD case) provides a better bound on the element growth and reveals the numerical rank. Although there exist unsymmetric matrices that LU factorization with even complete pivoting fails to reveal numerical ranks, we have not found any SPSD matrix that Algorithm DP fails to reveal the rank. Since the rank-revealing LU factorization of Chan [4] costs at least twice as many the flop count as Algorithm DP does for the SPSD case, Chan's method should not be used in this case. Furthermore, there is no efficient extension of Chan's method to the general case that the numerical rank is less than $n-1$. Finally, we comment on a limitation of Algorithm DP. If each diagonal blck A_i of M is large and sparse, then the diagonal pivoting is in conflict with the pivoting techniques aimed at reducing fill-ins, such as the minimum defree ordering (see Gill et al. [5] for the role of such techniques in solving the normal equations arising from a generally sparse linear program). This conflict between efficiency (i.e., sparsity retaining) and stability seems unavoidable for generally sparse matrices.

3. A block method

In this section we will utilize the algorithm in Section 2 to decompose the sparse, block-structured M which is defined in Section 1:

$$M = \begin{bmatrix} A_1 & & & & C_1^t \\ & A_2 & & & C_2^t \\ & & \ddots & & \vdots \\ & & & A_p & C_p^t \\ C_1 & C_2 & & C_p & C_{p+1} \end{bmatrix}.$$

Recall that M is SPSD and nearly singular. Let r_i =rank(A_i), where from now on rank(X) denotes the *numerical rank* of matrix X for convenience. Let η denote $\sum_{i=1}^{p}(m_i-$

r_i) and $k = m_F + \eta$. We will factor A_i, for each i at a time, by Algorithm DP. If $r_i < m_i$, we obtain a zero matrix of order $m_i - r_i$, denoted by $A_{i\sigma}$, at the end of the factor D for the block A_i. As explained in Section 2, we really have $|A_{i\sigma}| < \text{tol}$ mathematically. In the following we shall describe an efficient method for factoring M that exploits block structure of M. Henceforth, this method is called the block method. Let

$$A_1^* = \left[\begin{array}{c|c} A_1 & C_1^t \\ \hline C_1 & C_{p+1} \end{array}\right].$$

Applying Algorithm DP to A_1^* with some restrictions implied by the structures of the following matrices, we have

$$M_1^* P_1^* A_1^* P_1^{*'} M_1^{*'} = \left[\begin{array}{c|c|c} D_{1r} & & C_{1r}^t \\ \hline & A_{1\sigma} & C_{1\sigma}^t \\ \hline C_{1r} & C_{1\sigma} & C_{p+1} \end{array}\right],$$

where the $r_1 \times r_1$ matrix D_{1r} =diag$(d_1, \ldots, d_{r_1})$, $P_1^* = \left[\begin{array}{c|c} \widetilde{P}_1 & \\ \hline & I \end{array}\right]$ $\begin{matrix} m_1 \\ m_F \end{matrix}$, and $\widetilde{P}_1$ is the permutation matrix corresponding to the diagonal pivoting applied to A_1. By the block structure, M_1^* has the same sparse structure as $M_1^{*^{-1}}$:

$$M_1^{*^{-1}} = \left[\begin{array}{c|c} L_{11} & \\ \hline O & I \end{array}\right] = \left[\begin{array}{c|c|c} L_{1r} & & \\ \hline L_{1\sigma} & I & \\ \hline O & & I \end{array}\right] \begin{matrix} \Big\} m_1 \\ \Big\} m_F \end{matrix} \tag{7}$$

where L_{1r} is an $r_1 \times r_1$ unit lower triangular matrix, $L_{1\sigma}$ is $(m_1 - r_1) \times r_1$ and $[C_{1r}|C_{1\sigma}] = C_1 \widetilde{P}_1^t (L_{11}^t)^{-1}$. Pepeating the above process to factor A_q^* (similarly defined as A_1^*) for $q = 2, \cdots, p$, we obtain

$$M_p P_p \cdots (M_2 P_2 (M_1 P_1 M P_1^t M_1^t) P_2^t M_2^t) \cdots P_p^t M_p^t$$

$$= D' = \begin{bmatrix} D_{1r} & & & & & C_{1r}^t \\ & A_{1\sigma} & & & & C_{1\sigma}^t \\ & & \ddots & & & \vdots \\ & & & D_{pt} & & C_{pr}^t \\ & & & & A_{p\sigma} & C_{p\sigma}^t \\ C_{1r} & C_{1\sigma} & \cdots & C_{pr} & C_{p\sigma} & C_{p+1} \end{bmatrix},$$

where $M_i^{-1} = \text{diag}(I, \cdots, I, L_{ii}, I, \cdots, I)$ and $P_i = \text{diag}(I, \cdots, I, \widetilde{P}_i, I, \cdots, I)$. Since D' is SPSD and $A_{i\sigma} = O$, we have $C_{i\sigma} = O$ too. Let

$$P' = P_p \ldots P_1$$

and

$$L' = P'(M_p P_p \cdots M_1 P_1)^{-1}.$$

By following Golub and Van Loan [6, p.66] and by the special block-diagonal structures of M_i, P_i, and (7), L' can be shown to have the form:

$$L' = \left[\begin{array}{ccccc|c} L_{1r} & & & & & \\ L_{1\sigma} & I & & & & \\ & & \ddots & & & \\ & & & L_{pr} & & \\ & & & L_{p\sigma} & I & \\ \hline & & & & & I \end{array}\right]$$

and the decomposition so far is simplified to

$$P'MP'^t = L'D'L'^t.$$

Next, we permute zero blocks $A_{1\sigma}, \ldots, A_{p\sigma}$ (together with zero blocks $C_{i\sigma}$'s) to the last position of D'. Let $\tilde{D} = \tilde{P}D'\tilde{P}^t$ denote the resulting permuted matrix. Then, we have the following identities:

$$\begin{aligned} P'MP'^t &= L'\tilde{P}^t(\tilde{P}D'\tilde{P}^t)\tilde{P}L'^t = L'\tilde{P}^t\tilde{D}\tilde{P}L'^t \\ &= L'\tilde{P}^t \left[\begin{array}{ccccccccc} D_{1r} & & & & C_{1r}^t & & & & \\ & \ddots & & & \vdots & & & & \\ & & & D_{pr} & C_{pr}^t & & & & \\ C_{1r} & \cdots & C_{pr} & & C_{p+1} & C_{1\sigma} & \cdots & C_{p\sigma} & \\ & & & & C_{1\sigma}^t & A_{1\sigma} & & & \\ & & & & \vdots & & \ddots & & \\ & & & & C_{p\sigma} & & & A_{p\sigma} & \end{array}\right] \tilde{P}L'^t. \end{aligned} \tag{8}$$

Next, by using D_{ir} to eliminate C_{ir} and C_{ir}^t for $i = 1, \cdots, p$, (8) becomes

$$P'MP'^t = L'\tilde{P}^t(\tilde{L}D\tilde{L}^t)\tilde{P}L'^t, \tag{9}$$

where

$$\tilde{L} = \left[\begin{array}{ccc|cc} I & & & & \\ & \ddots & & & \\ & & I & & \\ \hline C_{1r}D_{1r}^{-1/2} & \cdots & C_{pr}D_{pr}^{-1/2} & I & \\ & O & & & I \end{array}\right] \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \end{array}\right\} \sum_{i=1}^{p} r_i \\ \left.\begin{array}{l} \\ \end{array}\right\} m_F \\ \left.\begin{array}{l} \\ \end{array}\right\} \eta \end{array},$$

$$D = \left[\begin{array}{ccc|cccc} D_{1r} & & & & & & \\ & \ddots & & & & & \\ & & D_{pr} & & & & \\ \hline & & & \widetilde{C}_{p+1} & C_{1\sigma} & \cdots & C_{p\sigma} \\ & & & C_{1\sigma} & A_{1\sigma} & & \\ & & & \vdots & & \ddots & \\ & & & C_{p\sigma}^t & & & A_{p\sigma} \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c}a\\a\\a\end{array}}\right\} \sum_{i=1}^{p} r_i \\ \left.\vphantom{\begin{array}{c}a\\a\\a\\a\end{array}}\right\} k = m_F + \eta \end{array},$$

and $\widetilde{C}_{p+1} = C_{p+1} - \sum_{i=1}^{p} C_{ir} D_{ir}^{-1} C_{ir}^t$. Now, (9) implies that

$\widetilde{P}(P'MP'^t)\widetilde{P}^t = \widetilde{P}L'\widetilde{P}^t(\widetilde{L}D\widetilde{L}^t)\widetilde{P}L'^t\widetilde{P}^t$.

Let $P = \widetilde{P}P'$ and $L = (\widetilde{P}L'\widetilde{P}^t)\widetilde{L}$. Then the above equation becomes

$$PMP^t = LDL^t,$$

where L can be shown to have the form:

$$L = \left[\begin{array}{ccc|c} L_{1r} & & & \\ & L_{2r} & & \\ & & \ddots & \\ & & L_{pr} & \\ \hline C_{1r}D_{1r}^{-1/2} & C_{2r}D_{2r}^{-1/2} & \cdots\ C_{pr}D_{pr}^{-1/2} & \\ L_{1\sigma} & & & \\ & L_{2\sigma} & & I \\ & & \ddots & \\ & & L_{p\sigma} & \end{array}\right] \begin{array}{l} \left.\vphantom{\begin{array}{c}a\\a\\a\end{array}}\right\} \sum_{i=1}^{p} r_i \\ \left.\vphantom{\begin{array}{c}a\\a\\a\\a\end{array}}\right\} k \end{array}.$$

Note that $A_{i\sigma}$ and $C_{i\sigma}$ are zeros, L has a similar sparse structure as that of the lower triangular part of M (after some permutations), and D is diagoal except for the block $\widetilde{C}_{p+1}$. Next, applying Algorithm DP to $\widetilde{C}_{p+1}$ will complete the LDL^t factorization and we have a method for solving (2).

Since this block method creates no fill-ins (assuming each block is dense), it is the most efficient sparse method one can design. Also, it can be easily implemented on multiprocessor parallel machines. Moreover, by using Algorithm DP for each block A_i, the block method is also stable.

4. Conclusion

It is important to find out a stable, efficient method for solving the normal equations arising from the interior methods. Large, sparse, angular linear programs have so may different applications that they merit special attentions. By exploiting the angular special structure, we have found a stable, efficient block method for solving nearly singular systems (2) arising from degenerate linear programs with large, sparse, and angular-structured constraint matrices. In the heart of the block method, we adopt the efficient and stable Algorithm DP which takes only $\frac{n^3}{6}$ flop count for factoring a nearly singular SPSD matrix. Algorithm DP uses an equivalent complete pivoting strategy and, in theory, can reveal the numerical rank of each diagonal block for the case when the numerical rank-deficiency is one. Moreover, it also reveals the numerical rank in all of our numerical tests even if the rank-deficiency is more than one.

Acknowledgements

The authors are grateful to the anonymous referee for his many helpful suggestions on improving the presentation of this paper.

References

[1] J. R. Bunch, "Analysis of the diagonal pivoting method", *SIAM J. Numer. Anal.* 8 (1971), 656-680.

[2] J. R. Bunch and L. Kaufman, "Some stable methods for calculating inertia and solving symmetric linear systems", *Mathematics of Computation* 31, No. 137 (1977), 163-179.

[3] J. R. Bunch and B. N. Parlett, "Direct methods for solving symmetric indefinite systems of linear equations", *SIAM J. Numer. Anal.* 8, No. 4 (1971), 639-655.

[4] T. F. Chan, "On the existence and computation of LU-factorizations with small pivots", *Mathematics of Computation* 42 (1984), 535-547.

[5] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin and M. H. Wright, "On projective Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method", *Mathematical Programming*, 36 (1986), 183-209.

[6] G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD (1983).

[7] J. N. Hooker, "Karmarkar's linear programming algorithm", *Interfaces* 16 (1986), 75-90.

[8] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall Press (1980).

[9] J. B. Rosen and Robert S. Maier, "Parallel solution of large-scale, block angular linear program", *Annals of Operations Research* 22 (1990), 23-41.

[10] J. A. Tomlin, "A note on comparing simplex and interior methods for linear programming", in *Progress in Mathematical Programming* (N. Megiddo ed.), Spbing-Verlag, New York (1989), 91-103.

[11] J. H. Wilkinson, *Algebraic Eigenvalue Problem*, Clarendon Press, Oxford (1965).

Institute of Applied Mathematics, National Tsing Hua University, Hsinchu, 30043, Taiwan.