JOURNAL OF THE
CHUNGCHEONG MATHEMATICAL SOCIETY
Volume **10**, July 1997

SPLITTING METHOD OF DENSE COLUMNS IN SPARSE LINEAR SYSTEMS AND ITS IMPLEMENTATION

SEYOUNG OH AND SUN JOO KWON

ABSTRACT. It is important to solve the large sparse linear system appeared in many application field such as $AA^Ty = \beta$ efficiently. In solving this linear system, the sparse solver using the splitting method for the relatively dense column is experimentally better than the direct solver using the Cholesky method.

1. Introduction

In the context of interior-point method for solving linear programming efficiently and many other applications, the systems of linear equations of the form

$$AD^2A^Ty = \beta \tag{1}$$

has to be solved where A is a large sparse full row rank matrix having a small number of dense columns and D is a diagonal matrix. The process of solving the system requires the most time consumming in every iteration of each solving technique of the application problems.

For the system (1) many well-behaved methods have been suggested by using Cholesky factorization ([1], [2]) and preconditioned conjugate gradient ([3]). Most of the methods, however, are suffering from the difficulties with the matrix AD^2A^T being very dense when A has a small number (even one) of dense columns and these methods are sensitive to numerical error and rank-deficiency of preconditioners.

Received by the editors on June 30, 1997.
1991 *Mathematics Subject Classifications* : Primary 65F10.
Key words and phrases: Splitting method, Sparse linear system.

A different approach called splitting method for a slightly different problem from (1) has been proposed in [4] without computational results. It is based on making the matrix A sparse with some linking matrices and solving the large sparse matrix. In this paper a slightly modified splitting method with a different parameter is applied to the system (1). We implemented the method and ran the codes with many of randomly generated test problems to obtain the optimal parameter. We used subroutines from SLATEC for sparse solver and LAPACK for Cholesky solver in NetLib.

The numerical results show that the larger the problem size is, the faster the method works and when the dense column is splitting into approximately some proper parameter k sparse columns, the efficiency of this method is obtained.

In the section 2, some application problems of (1) are introduced. And a modified splitting method for (1) is described in the section 3. The comparison between the direct solver of Cholesky factorization and the large sparse conjugate gradient method with incomplete Cholesky preconditioner is implemented and the computational results for the codes are shown.

2. Applications

2.1. Interior-point method. Consider the following general linear program;

$$\begin{array}{rrcl} \min & c^T x & & \\ \text{s.t.} & Ax & \geq & b, \\ & x & \geq & 0, \end{array} \tag{2}$$

where $A \in R^{n \times m}$, $n \leq m$, $c \in R^m$, $b \in R^n$. By adding surplus variables s to the constraints, the standard form of (2) can be written as

$$\begin{array}{rrcl} \min & c^T x & & \\ \text{s.t.} & Ax - s & = & b, \\ & x, s & \geq & 0. \end{array} \tag{3}$$

The interior point algorithm such as like primal-dual path following method considers the dual of (2);

$$
\begin{aligned}
\max \quad & b^T w \\
\text{s.t.} \quad A^T w &\le c, \\
w &\ge 0,
\end{aligned} \tag{4}
$$

and by adding slack variables t to the constraint, the standard form of (4) is obtained as following

$$
\begin{aligned}
\max \quad & b^t w \\
\text{s.t.} \quad A^T w + t &= c, \\
w, t &\ge 0.
\end{aligned} \tag{5}
$$

Denote the diagonal matrices X, W, S and T with the components of x, w, s and t on their diagonals respectively. Then the interior point algorithm always keeps the matrices X, W, S and T positive and the system of equations

$$
\begin{bmatrix} -X^{-1}T & A^T \\ A & W^{-1}S \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta w \end{bmatrix} = \begin{bmatrix} a \\ b \end{bmatrix} \tag{6}
$$

must be solved at every iteration, which is main computational work in the whole algorithm. Because of the indefinite system, it is solved for Δx first in terms of Δw ;

$$\Delta x = -XT^{-1}(a - A^T \Delta w)$$

and substituting this into the second subsystem of (6) leads to

$$(AXT^{-1}A^T + W^{-1}S)\Delta w = (b + AXT^{-1}a).$$

Since $X, T, W,$ and S are diagonal matrices, $AXT^{-1}A^T + W^{-1}S$ is symmetric positive definite matrix and can be considered as the type AD^2A^T which is presented in this paper.

2.2. Projecting the direction of steepest descent. In the optimization problem with linear constraints, i.e.,

$$\begin{aligned} \min\ \varphi(x) \\ \text{s.t}\quad Ax &= b, \\ x &\geq 0, \end{aligned}$$

the direction of steepest descent is projected into the feasible region. At current feasible point x_0, the projected steepest direction p into feasible region is

$$p = P(\nabla\varphi(x_o)) = [I - A^T(AA^T)^{-1}A](\nabla\varphi(x_0)).$$

p is computed by the two step procedures :

1. Solve $AA^Ty = A \nabla \varphi(x_0)$.
2. $p = \nabla\varphi(x_0) - A^Ty$.

The first step of the above precedures is one of the applications for our discussing problem where D is an identity matrix.

2.3. Least square problem with a design matrix A^T. The least square problem with A^T, i.e.,

$$\min_x ||A^Tx - c||,$$

is reduced to the positive definite system;

$$AA^Tx = Ac.$$

Use of the Cholesky factorization to get a solution is often efficient. The QR factorization of A^T can also be used since it is more stable where the condition number of AA^T is the square of the condition number of A^T.

3. Splitting method of the dense columns

Suppose that the matrix A is a large sparse with some of the number of dense columns and have full row rank and that the number of dense columns is small and have been identified. Consider the system of equations

$$AD^2A^Tx = b,$$

where D is a diagonal matrix. Partition the matrix A into

$$A = \begin{bmatrix} A_S & A_D \end{bmatrix},$$

where A_S and A_D represent the sparse part and the dense part of A respectively.

By partitioning the diagonal matrix D into two blocks as

$$D = \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix},$$

where the sizes of D_1 and D_2 correspond to those of A_S and A_D, then AD can be written as follows:

$$AD = \begin{bmatrix} A_S & A_D \end{bmatrix} \begin{bmatrix} D_1 & 0 \\ 0 & D_2 \end{bmatrix} = \begin{bmatrix} A_SD_1 & A_DD_2 \end{bmatrix}.$$

Then the submatrices A_SD_1 and A_DD_2 of AD are still sparse part and dense part respectively. Note that

$$\begin{aligned} AD^2A^T &= AD(AD)^T \\ &= \begin{bmatrix} A_SD_1 & A_DD_2 \end{bmatrix} \begin{bmatrix} A_SD_1 & A_DD_2 \end{bmatrix}^T \\ &= (A_SD_1)(A_SD_1)^T + (A_DD_2)(A_DD_2)^T. \end{aligned} \tag{7}$$

Now we split A_DD_2 into a sparse matrix having the same number of rows but more columns. It satisfies

$$A_DD_2 = \Delta E,$$

where

$$E = \frac{1}{\sqrt{k}} \begin{bmatrix} e_k & & & & \\ & e_k & & & \\ & & e_k & & \\ & & & \ddots & \\ & & & & e_k \end{bmatrix}, \quad kl \times l \text{ matrix.}$$

Here l is the number of columns of $A_D D_2$ and $e_k = \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}^T$ is a k-vector for some constant k. All the columns of $A_D D_2$ is expanded into kl new sparse columns. This distributes the column entries of $A_D D_2$ into Δ, so Δ is related to $A_D D_2$ and becomes a sparse matrix. When θ is a threshold parameter and any i-th column of $A_D D_2$ is split into a set of k columns, each column contains exactly θ elements of i-th column of $A_D D_2$ except for the last column in some case. Thus $k = \lceil \frac{n}{\theta} \rceil$.

For example, suppose

$$A_D D_2 = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \\ p_{31} & p_{32} \\ p_{41} & p_{42} \\ p_{51} & p_{52} \end{bmatrix} \text{ and } l = 2,\ \theta = 2. \tag{8}$$

Then $k = \lceil \frac{5}{\theta} \rceil = \lceil \frac{5}{2} \rceil = 3$. By the definition of E,

$$E = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

and then

$$\Delta = \sqrt{3} \begin{bmatrix} p_{11} & 0 & 0 & p_{12} & 0 & 0 \\ p_{21} & 0 & 0 & p_{22} & 0 & 0 \\ 0 & p_{31} & 0 & 0 & p_{32} & 0 \\ 0 & p_{41} & 0 & 0 & p_{42} & 0 \\ 0 & 0 & p_{51} & 0 & 0 & p_{52} \end{bmatrix}.$$

The equality

$$\Delta E = A_D D_2$$

is achieved, where Δ is quite sparse.

Let

$$L = \begin{bmatrix} L_1 & & & & \\ & L_2 & & & \\ & & L_3 & & \\ & & & \ddots & \\ & & & & L_l \end{bmatrix},$$

where L_i is $(k-1) \times k$ matrix with the shape such as

$$L_i = \begin{bmatrix} 1 & -1 & & & & \\ & 1 & -1 & & & \\ & & 1 & -1 & & \\ & & & \ddots & \ddots & \\ & & & & \ddots & \ddots \\ & & & & 1 & -1 \end{bmatrix}.$$

Now, a new matrix G from Δ and $A_S D_1$ is defined as

$$G = \begin{bmatrix} A_S D_1 & \Delta \\ 0 & L \end{bmatrix}.$$

It can be used to obtain the solution of the original system.

For the example (8), G can be written as

$$G = \begin{bmatrix} & \sqrt{3}p_{11} & 0 & 0 & \sqrt{3}p_{12} & 0 & 0 \\ & \sqrt{3}p_{21} & 0 & 0 & \sqrt{3}p_{22} & 0 & 0 \\ A_S D_1 & 0 & \sqrt{3}p_{31} & 0 & 0 & \sqrt{3}p_{32} & 0 \\ & 0 & \sqrt{3}p_{41} & 0 & 0 & \sqrt{3}p_{42} & 0 \\ & 0 & 0 & \sqrt{3}p_{51} & 0 & 0 & \sqrt{3}p_{52} \\ & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 1 & -1 & 0 \\ & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}.$$

The following theorem implies that the solution $(x,\ y)^T$ of the system of equations

$$GG^T \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \tag{9}$$

represents x as the solution of $AD^2A^Tx = b$, i.e., $x = (AD^2A^T)^{-1}b$.

THEOREM 3.1. *G has full row rank if and only if A has full row rank. Let $(x,\ y)^T$ be the solution to (9) for G having full row rank. Then $x = (AD^2A^T)^{-1}b$.*

Proof. Note that the number of dense column A_DD_2 is l and Δ has kl columns and n rows. Then G has $n + l(k-1)$ rows and the column operations on G, without effecting the rank of G, can be performed to have a same row rank matrix as follows

$$\begin{aligned} G &= \begin{bmatrix} A_SD_1 & \Delta \\ 0 & L \end{bmatrix} \\ &= \begin{bmatrix} A_SD_1 & \Delta_1 & \Delta_2 & \cdots & \Delta_l \\ & L_1 & & & \\ 0 & & L_2 & & \\ & & & \ddots & \\ & & & & L_l \end{bmatrix}, \end{aligned}$$

where Δ_i = split block of i-th column of A_DD_2. Apply column operations to G then

$$\begin{bmatrix} A_SD_1 & A_DD_2 & \Delta_1' & \Delta_2' & \cdots & \Delta_l' \\ & & L_1' & & & \\ 0 & 0 & & L_2' & & \\ & & & & \ddots & \\ & & & & & L_l' \end{bmatrix}$$

$$= \begin{bmatrix} AD & \Delta_1' & \Delta_2' & \cdots & \Delta_l' \\ & L_1' & & & \\ 0 & & L_2' & & \\ & & & \ddots & \\ & & & & L_l' \end{bmatrix},$$

where Δ_i' is the submatrix of Δ_i by dissecting the first column of Δ_i and L_i' is obtained from L_i for the same way. The last matrix has the same row rank as that of G and G has full row rank if and only if A or AD does since the matrix

$$\begin{bmatrix} L_1' & & & \\ & L_2' & & \\ & & \ddots & \\ & & & L_l' \end{bmatrix}$$

is an invertible $(k-1)l \times kl$ matrix.

From the definition of G, we obtain

$$\begin{aligned} GG^T &= \begin{pmatrix} A_S D_1 & \Delta \\ 0 & L \end{pmatrix} \begin{pmatrix} (A_S D_1)^T & 0 \\ \Delta^T & L^T \end{pmatrix} \\ &= \begin{pmatrix} (A_S D_1)(A_S D_1)^T + \Delta\Delta^T & \Delta L^T \\ L\Delta^T & LL^T \end{pmatrix}. \end{aligned}$$

Rewriting the system (9) as

$$\begin{aligned} ((A_S D_1)(A_S D_1)^T + \Delta\Delta^T)x + \Delta L^T y &= b, \\ L\Delta^T x + LL^T y &= 0, \end{aligned} \tag{10}$$

we can solve the second system for y first to get

$$y = -(LL^T)^{-1} L \Delta^T x$$

since LL^T is always invertible. Substitute y into the first system of (10) and solve for x. Then x is

$$\begin{aligned} x &= [A_S D_1 (A_S D_1)^T + \Delta\Delta^T - \Delta L^T (LL^T)^{-1} L \Delta^T]^{-1} b \\ &= [A_S D_1 (A_S D_1)^T + \Delta(I - L^T (LL^T)^{-1} L)\Delta^T]^{-1} b. \end{aligned} \tag{11}$$

The matrix $I - L^T(LL^T)^{-1}L$ is the projection onto the nullspace of L. Since for the matrix L_i the linear system $L_i z = 0$ gives $z = ce_k$ for any constant c, the nullspace of L is the $n-$dimensional space spanned by the n orthonormal vectors;

$$\frac{1}{\sqrt{k}}\begin{bmatrix} e_k \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \frac{1}{\sqrt{k}}\begin{bmatrix} 0 \\ e_k \\ \vdots \\ 0 \end{bmatrix}, \cdots, \frac{1}{\sqrt{k}}\begin{bmatrix} 0 \\ \vdots \\ 0 \\ e_k \end{bmatrix}.$$

From the simple computation, we see that EE^T is the projection onto the nullspace of L which is identical to $I - L^T(LL^T)^{-1}L$. Using that equality and (7), the equation (11) is reduced to

$$\begin{aligned} x &= [A_S D_1 (A_S D_1)^T + \Delta(I - L^T(LL^T)^{-1}L)\Delta^T]^{-1}b \\ &= [A_S D_1 (A_S D_1)^T + \Delta EE^T \Delta^T]^{-1} \\ &= [A_S D_1 (A_S D_1)^T + A_D D_2 (A_D D_2)^T]^{-1}b \\ &= [AD^2A^T]^{-1}b. \end{aligned}$$

□

4. Implementations and Experimental Results

By the theorem 3.1, the expanded linear sparse system

$$GG^T \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \tag{12}$$

must be solved to obtain x which is the solution of the original system $AD^2A^Tx = b$.

For the comparison purposes, we implement the direct solver for $AD^2A^Tx = b$ using Cholesky factorization and the sparse solver for (12) using conjugate gradient with incomplete Cholesky preconditioner. The matrix A, D and b are randomly generated with normal distribution. The size of A matrices moves in $101 \leq m \leq 2010$ and $100 \leq n \leq 2000$.

The codes for the Cholesky solver, DPOTRF and DPOTRS from LAPACK in NetLib, and incomplete Cholesky conjugate gradient codes from SLATEC in NetLib were used to solve the original system (1) and (12), respectively. The data of A, D and b are generated as CSC-form

Table 1. Execution Time Comparision

n	ndc	Cholesky for $AD^2A^Ty = \beta$	IC Preconditioning CG for $GG^T \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$	
		Time	Optimal theta	Time (sec)
100	1	1.12	7	0.10
	2	1.13	27	0.13
	5	1.21	25	0.26
	10	1.26	25	0.56
200	1	10.22	20	0.21
	2	10.31	25	0.33
	5	10.44	40	0.84
	10	10.64	50	1.89
500	1	179.12	40	1.01
	2	179.59	50	1.97
	5	180.95	105	4.84
	10	182.53	110	9.58
700	1	510.80	30	1.60
	2	511.74	60	3.83
	5	512.80	90	9.55
	10	518.34	140	20.04
1000	1		30	3.32
	2		60	7.59
	5		80	20.74
	10		100	44.41
2000	1		50	12.40
	2		100	32.60
	5		70	96.60

which is sparse data structure for SLATEC. The dense part of A has at most 10% density of sparse column for most of the problems. The incomplete Cholesky factorization is one of the most efficient preconditioner for iterative methods of solving large sparse symmetric positive definite linear systems.

The weakness is the failure of the factorization due to nonpositive pivots. Several methods have been suggested to overcome this problem

([5], [6]). To escape this trouble in our problem, the problems that do not occur the failure of the factorization are only examined. The modified incomplete Cholesky factorization as preconditioner will be considered in the future research. All the codes for our test problems are derived by using Fortran-77 on HP755/90 workstation.

The codes were tested on A which has 100, 200, 500, 700, 1000, 2000 sparse columns and 1, 2, 5 , 10 dense columns for each case. The number of row is the same as that of sparse columns. For each case the parameter θ varies in at most 50% of the number of sparse columns. That is, each dense column is splitted into 2 - 50 columns. Table 1 represents the average execution time(sec) of more than 4 problems for each case that A has 100 - 2000 sparse columns and 1, 2, 5 or 10 dense columns. *ndc* is the number of dense columns. Experimentally, in conclusion, the sparse solver used splitting method for the relatively dense columns is efficient.

REFERENCES

1. R. E. Marsten, M. J. Saltzman, et. al., *Implementation of a dual interior point algorithm for linear programming,* ORSA Journal on Computing, 1(1989), pp287-297.
2. R. J. Vanderbei, *A brief description of ALPO,* OR Letters, (1991), pp531-534.
3. S. Mehrotra, *Implementations of affine scaling methods; Approximate solutions of systems of linear equations using preconditioned conjugate gradient methods,* Technical Report 89-04, Dept. Ind. Eng. and Mgmt. Sci., Northwestern Univ., Evanston, IL, (1989).
4. Vanderbei, *Splitting dense columns in sparse linear systems. Linear algebra and its Applications,* 152(1991), pp107-117.
5. X. Wang, et. al., *Incomplete Cholesky factorization with sparsity pattern modification,* private letter, (1993).
6. G. Wittum and F. Liebau, *On truncated incomplete decompositions,* BIT, 29(1989), pp719-740.
7. Jack Dongarra, et. al., *LAPACK User's Guide,* SIAM press, Philadephia, (1995).

8. R. Vanderbei, *Symmetric Quasi-definite Matrices,* private letter, (1996).
9. Gene H. Golub and Charles F. Van Loan, *Matrix Computation,* (1983).
10. Gilbert Strang, *Linear algebra and its Application,* HBJ, (1988).
11. James P. Ignizio and Tom M. Cavalier, *Linear Programming,* Prentice-Hall, (1994).
12. Youcef Saad, *SPARSKIT: A basic tool kit for sparse matrix computations, version 2,* (1994).
13. Richard L. Burden and J. Douglas Faires, *Numerical Analysis,* PWS, (1993).

Seyoung Oh
Department of Mathematics
Chungnam National University
Taejon 305-764, Korea
E-mail: `soh@math.chungnam.ac.kr`

Sun Joo Kwon
Department of Mathematics
Chungnam National University
Taejon 305-764, Korea
E-mail: `sjkwon@math.chungnam.ac.kr`