

Computing the Strict Chebyshev Solution of Overdetermined Linear Equations

By Nabih N. Abdelmalek

Abstract. A method for calculating the strict Chebyshev solution of overdetermined systems of linear equations using linear programming techniques is described. This method provides: (1) a way to determine, for the majority of cases, all the equations belonging to the characteristic set, (2) an efficient method to obtain the inverse of the matrix needed to calculate the strict Chebyshev solution, and (3) a way of recognizing when an element of the Chebyshev solution equals a corresponding element of the strict Chebyshev solution. As a result, in general, the computational effort is considerably reduced. Also the present method deals with full rank as well as rank deficient cases. Numerical results are given.

1. Introduction. Consider the overdetermined system of linear equations

$$(1) \quad Ca = f,$$

where C is a given real $n \times m$ matrix of rank $k \leq m < n$ and f is a given real n -vector. Let E denote the set of n equations (1). The Chebyshev solution (C.S.) of system (1) is the m -vector $a^* = (a_j^*)$ which minimizes the Chebyshev norm z ,

$$(2) \quad z = \max |r_i(a)|, \quad i \in E,$$

where r_i is the i th residual in (1) and is given by

$$(3) \quad r_i = c_{i1}a_1 + \cdots + c_{im}a_m - f_i, \quad i \in E.$$

Let us denote the C.S. a^* by $(a^*)_{\text{C.S.}}$.

It is known that if C satisfies the Haar condition, the C.S. $(a^*)_{\text{C.S.}}$ is unique. Otherwise it may not be unique. Rice [7] introduced a particular C.S. for functions defined on a finite point set, which is always unique, and called it the strict Chebyshev solution (S.C.S.). See also [8, pp. 237-246]. When the C.S. is not unique, there is a certain degree of freedom for some of the residuals (3). For these residuals, the maximum absolute value is minimized over the C.S. The resulting solution is the S.C.S.

Later, Descloux [2] presented the same S.C.S. to system (1), in a slightly different manner. His presentation is particularly suitable for algorithmic procedures. He proved also the important result that the L_p solution of system (1) converges to the S.C.S. as p tends to infinity. We denote the S.C.S. of (1) by $(a^*)_{\text{S.C.S.}}$.

Received April 15, 1975; revised June 7, 1976.

AMS (MOS) subject classifications (1970). Primary 65D15, 65D20. Secondary 90C05, 90C50.

Key words and phrases. Discrete linear Chebyshev approximation, strict Chebyshev approximation, overdetermined linear equations, linear programming.

Copyright © 1977, American Mathematical Society

Descloux presentation may be summarized as follows. Assume that matrix C is of rank m and let the Chebyshev deviation to system (1) be $z_1^* = z^*$. It is known that $(a^*)_{C.S.}$ and z_1^* are the solution of $(m + 1)$ equations of E known as the reference set (RS). They have the form

$$(4) \quad c_{i1}a_1 + \dots + c_{im}a_m + \delta_i z = f_i, \quad i \in RS,$$

where δ_i is either $+1$ or -1 . For later use we write (4) as

$$(4') \quad c_{i1}a_1 + \dots + c_{im}a_m = f_i - \delta_i z, \quad i \in RS.$$

Assume that the C.S. $(a^*)_{C.S.}$ is not unique and let W_1 be the set of all Chebyshev solutions to system (1). Let R_1 be the collection of all equations $i \in E$ for which $|r_i(a)| = z_1^*$ for all the $(a^*)_{C.S.} \in W_1$. R_1 is denoted by Descloux as the characteristic set of f relative to C . Each of W_1 and R_1 is nonvoid and W_1 is a subset of the solutions of the system .

$$(5) \quad c_{i1}a_1 + \dots + c_{im}a_m = f_i - \delta_i z, \quad i \in R_1,$$

where δ_i is either $+1$ or -1 . System (5) is of rank $s_1 \leq m$.

The original problem may now be viewed as follows. It is required to obtain the C.S. of the system $(E - R_1)$ subject to the s_1 conditions (5). This may be done by eliminating s_1 appropriate elements of the vector a , from $(E - R_1)$ using (5) and then calculating the C.S. of the obtained reduced system.

System (1) thus reduces to the system of $(E - R_1)$ equations in $(m - s_1)$ unknowns of the form

$$(6) \quad C^{(2)}a^{(2)} = f^{(2)},$$

where $C^{(2)}$ is of rank $(m - s_1)$.

The same procedure is now repeated for system (6). If the C.S. of (6) is not unique, the characteristic set R_2 of (6) is obtained. Again, we eliminate s_2 appropriate elements of $a^{(2)}$ from $(E - R_1 - R_2)$ and obtain the C.S. of the further reduced system.

The above process is repeated if necessary a finite number of times, until the C.S. of the most reduced system is unique. A nonsingular system of m equations then presents itself, the solution of which is the S.C.S. of (1). This system consists of s_1 equations of R_1 plus s_2 equations of R_2 plus Let this system be

$$(7) \quad Da = d.$$

For later use, from (7), we write the S.C.S. as

$$(7') \quad (a^*)_{S.C.S.}^T = d^T(D^T)^{-1},$$

where the T refers to the transpose.

The first attempt to describe an algorithm for calculating the S.C.S. of (1), following the presentation of Descloux was given by Duris and Temple [4].

The present algorithm is essentially that of Duris and Temple. However, our computational scheme differs in several significant respects with the result that,

normally, the computational effort is reduced considerably. The features of this scheme may be summarized as follows.

(a) In obtaining the C.S. of (1), we solve the linear programming formulation of the C.S. problem. By examining the final tableau of the programming problem, we give a simple procedure by which, for the majority of cases, we determine all the equations belonging to a characteristic set R_1 . If this procedure is not followed, several major iterations may be needed to obtain system (6) from system (1). (b) In the present work, matrix $(D^T)^{-1}$ of (7') needed for the S.C.S. is obtained in an efficient way. This is done by successively modifying the inverse of the basis matrix B^{-1} of the programming problem. Normally, the effort in obtaining $(D^T)^{-1}$ from B^{-1} is small. (c) The present method provides a way to recognize when an element of $(a^*)_{\text{S.C.S.}}$ equals a corresponding element of $(a^*)_{\text{C.S.}}$. In some cases this results in considerable saving of the computing effort. Example 2 in Section 2.4 and Table 1 in Section 4 below demonstrate the efficiency of the present method. (d) The present method also deals with rank deficient as well as full rank cases.

2. Analysis and Description of the New Method. The C.S. of system (1) is obtained by solving the corresponding linear programming problem, using the algorithm described in [1]. Without loss of generality, assume that $\text{rank}(C) = m$. Again let the $(m + 1)$ square matrix B denote the basis matrix for the optimal C.S. and B^{-1} be its inverse. Let also b_B be the optimal basic solution and $\{z_i - f_i\}, i = 1, \dots, 2n$, be the marginal costs in the final tableau. It is known [6] that B is the transpose of the coefficient matrix on the l.h.s. of (4). Also the residuals (3) are given by $r_i = \pm [(z_i - f_i) - z^*]$. So that if $(z_i - f_i) = 0$ or $2z^*$, r_i is given by $|r_i| = z^*$. From the final tableau of the programming problem we find out whether the C.S. of (1) is unique. We may also determine the characteristic set R_1 .

2.1. The Characteristic Set R_1 .

LEMMA 1. *If b_B has no zero components, the C.S. of (1) is unique.*

Proof. It is known that the elements b_{B_i} of b_B and the elements w_i of the vector w introduced in [4, p. 692] are given by $b_{B_i} = \pm \alpha w_i$, where α is a constant. The lemma is thus proved from Lemma 2.3 in [4], as in this case $\text{rank}(R_1) = m$.

Assume that we have obtained all the optimal basic solutions of the linear programming problem for system (1). Let $b_{B(1)}, b_{B(2)}, \dots$ be such solutions. Assume also that each of these solutions is degenerate. That is each has one or more zero components. We again deduce from Lemma 2.3 in [4] and from the definition of R_1 that R_1 consists of the union of the equations in the reference sets (4) associated with the nonzero elements of the corresponding $b_{B(i)}$.

To obtain all the optimal basic solutions and hence the characteristic set R_1 , we may follow the procedure suggested in Hadley [5, pp. 166–168]. However, this procedure is costly in both time and programming effort. That is because it requires the changing of the final simplex tableau.

A simpler procedure is here followed which requires the calculation of some of the optimal solutions without the need to change the simplex tableau. To start with, the equations in (4) corresponding to the nonzero elements of the b_B at hand belong

to R_1 . The nonbasic columns are then examined and those which have zero marginal costs are considered. Let i be one of such columns. Check in the usual manner [5, p. 167] if column i may enter the basis with positive level. If so, calculate the new optimal solution b'_B without changing the simplex tableau. Then the equations of the new RS (4) which correspond to the nonzero elements of b'_B belong to R_1 . This of course includes column i itself.

This procedure is repeated for every nonbasic column having zero marginal cost. The reduced system (6) is then calculated as described in Section 2.4 below. The equations in (6) which have zero coefficients also belong to R_1 and are to be deleted from system (6). We call this the simplified procedure.

This simplified procedure does not in general calculate all the optimal solutions, but it is successful in the majority of cases in finding the set R_1 . In Table 1 of Section 4 below, this procedure did work for all examples but one (Example 3a). For this example, two major iterations instead of one were needed to determine R_1 .

LEMMA 2. *Assume that we have used the simplified procedure to obtain the optimal solutions $b_{B(1)}, \dots, b_{B(r)}$. Assume that these solutions have q zero components in common. By a zero component in common, we mean that each $b_{B(i)}$ has its j th component say, $= 0$. Then the equations corresponding to the nonzero elements of these solutions have rank $(m - q)$.*

Proof. It is easy to show the following. For these $b_{B(i)}$ to have q zeros in common, the columns in the simplex tableau associated with the nonzero components of these $b_{B(i)}$, each has q zero elements in common with the q zeros. The proof of the lemma thus follows.

From Lemma 2, if $q = 0$, the C.S. is unique.

Example 1. Consider the C.S. of the equations

$$(8) \quad a_1 - 15a_2 = -5, \quad -.5a_1 + 7.5a_2 = 17.5, \quad 2a_2 = 12, \quad -4a_2 = 6.$$

Two of the optimal basic solutions obtained by the simplified procedure are $(0, 2/3, 1/3)^T$ and $(1/3, 0, 2/3)^T$. They correspond respectively to Eqs. (1, 3 and 4) and (1, 3 and 2) in (8). That is all four equations (8) form R_1 . Again, these two solutions have no zero component in common. Thus the C.S. is unique where $(\alpha^*)_{C.S.} = (0, 1)^T$ and $z^* = 10$.

Because we are using the simplified procedure, we assume that R_1 refers to the union of the equations associated with the nonzero components of the solutions $b_{B(i)}$ obtained by the simplified procedure and s_1 refers to the rank of these equations. Also assume that these $b_{B(i)}$ have q zero components in common.

2.2. Calculating Matrix $(D^T)^{-1}$. As mentioned before, this is done by successively modifying matrix B^{-1} .

In the product form, B^{-1} may be given by

$$(9) \quad B^{-1} = E_{m+1}E_m \cdots E_1,$$

where the E_i are $(m + 1)$ square matrices and are given for example in [5, p. 48]. It is known that the factorization (9) is not unique. In the present algorithm, we need to calculate the matrices $(E_{m+1}^{-1}B^{-1}), (E_m^{-1}E_{m+1}^{-1}B^{-1}), \dots, (E_{m+1-q}^{-1} \cdots E_{m+1}^{-1}B^{-1})$. If necessary we exchange the rows of B^{-1} before calculating E_{m+1}^{-1} and the columns

of $(E_{m+1}^{-1}B^{-1}), \dots$, before calculating E_m^{-1}, \dots . One of the reasons is to achieve maximum numerical stability in calculating the E_i^{-1} . See also Figure 1 below. The following is a useful lemma.

LEMMA 3. *The last column in B^{-1} equals b_B .*

Assume now that we have obtained a degenerate optimal solution b_B . Assume that q of the zero elements of b_B are common among the other optimal solutions obtained by the simplified procedure. Assume also that these q zero elements in b_B are the elements $m, m - 1, \dots, (m - q + 1)$. If not, permute the elements of b_B and the corresponding rows B^{-1} .

FIGURE 1

b_B	B^{-1}	$E_6^{-1}B^{-1}$	$E_5^{-1}E_6^{-1}B^{-1}$	$E_4^{-1}E_5^{-1}E_6^{-1}B^{-1}$
x	x x x x x x	x x x x x 0	x x x x 0 0	x x x 0 0 0
x	x x x x x x	x x x x x 0	x x x x 0 0	x x x 0 0 0
x	x x x x x x	x x x x x 0	x x x x 0 0	x x x 0 0 0
0	x x x x x 0	x x x x x 0	x x x x 0 0	x x x 1 0 0
0	x x x x x 0	x x x x x 0	x x x x 1 0	x x x 0 1 0
x	x x x x x x	x x x x x 1	x x x x 0 1	x x x 0 0 1
(a)	(b)	(c)	(d)	(e)

Figure 1 shows vector b_B and matrix B^{-1} for $m = 5$ and $q = 2$. The matrices $E_6^{-1}B^{-1}, E_5^{-1}E_6^{-1}B^{-1}$ and $E_4^{-1}E_5^{-1}E_6^{-1}B^{-1}$ are also shown. An x denotes in general a nonzero element.

Let $[P]_{m \times m}$ denote the upper left m by m submatrix of a given matrix P . Then we mention here that $[E_{m+1}^{-1}B^{-1}]_{m \times m}$ is the inverse of the transpose of the coefficient matrix of the first m equations of (4'). Thus from (4'),

$$(10) \quad (a^*)_{C.S.}^T = f^T [E_{m+1}^{-1}B^{-1}]_{m \times m},$$

where the elements of f are the first m elements on the r.h.s. of (4'). See also Eq. (5) in [1].

Consider now the nonsingular system (7). In Figure (2a), we assume that system (7) consists of $s_1 (= 3)$ equations of R_1 plus $s_2 (= 2)$ equations of R_2 . The elements of $a^{(1)}$ are permuted such that $a^{(2)}$ is obtained by eliminating the first s_1 elements of $a^{(1)}$. It is not difficult to show that $(D^T)^{-1}$ would be given by Figure (2b), where again, an x denotes in general a nonzero element. Figure (2c) will be referred to later.

FIGURE 2

D	$(D^T)^{-1}$	$(D^T)^{-1}$
x x x x x	x x x 0 0	x x x 0 0
x x x x x	x x x 0 0	x x x 0 0
x x x x x	x x x 0 0	x x x 0 0
0 0 0 x x	x x x x x	0 x x x x
0 0 0 x x	x x x x x	0 x x x x
(a)	(b)	(c)

Again in the product form, matrix $(D^T)^{-1}$ may be given by

$$(11) \quad (D^T)^{-1} = (E_m \cdots) \cdots (E_{s_1+s_2} \cdots E_{s_1+1})(E_{s_1} \cdots E_1),$$

where the E_i are suitable m by m matrices.

LEMMA 4. Let matrix G_1 be defined by $G_1 = E_{m+1-q}^{-1} \cdots E_m^{-1} E_{m+1}^{-1} B^{-1}$ (G_1 is illustrated by Figure (1e)). Then matrix $(E_{s_1} \cdots E_1)$ given in (11) and matrix G_1 are related by

$$(12) \quad (E_{s_1} \cdots E_1) = [G_1]_{m \times m}.$$

Proof. Since $m - q = s_1$, from (9)

$$(13) \quad G_1 = E_{s_1} \cdots E_2 E_1.$$

Also since the first s_1 equations of (7) are themselves the first s_1 equations of (4'), (12) is true and the lemma is proved.

The calculation of $(D^T)^{-1}$ is now obvious from (11) and (12). Matrix $[G_1]_{m \times m}$ is calculated from B^{-1} of system (1). Similarly, $[G_2]_{q \times q}$ is calculated from B^{-1} of system (6). We mention here that B^{-1} of system (6) would be calculated and stored into the right lower $(q + 1)$, i.e. $(m + 1 - s_1)$ unit submatrix resulting from calculating the matrix G_1 . See Figure (1e). The premultiplication of $[G_1]_{m \times m}$ by $[G_2]_{q \times q}$ is then performed. This is simply done by premultiplying the q rows $(s + 1), \dots, m$ of $[G_1]_{m \times m}$ by the submatrix $[G_2]_{q \times q}$. This procedure is continued until $(D^T)^{-1}$ is calculated.

2.3. When Does an Element of $(a^*)_{S.C.S.}$ Equal a Corresponding Element of $(a^*)_{C.S.}$.

LEMMA 5. Consider the first s_1 columns of matrix G_1 defined above. If in any one of these columns $j \leq s_1$, there exist q zero elements in the positions of the q zero elements in b_B , then $(a^*)_{S.C.S.} = (a^*)_{C.S.}$.

Proof. From (13) we write (11) in the form

$$(14) \quad (D^T)^{-1} = E_m \cdots E_{s_1+1} [G_1]_{m \times m}.$$

Consider a column $j \leq s_1$ of G_1 satisfying the assumption of the lemma. Because of the existence of the mentioned q zero elements in this column, and from the structure of matrix E_{s_1+1} , the last q elements of column j of $(E_{s_1+1} [G_1]_{m \times m})$ remain zeros. By inductive argument, the consecutive premultiplication by E_{s_1+2}, \dots, E_m , result in having the last q elements of column j in $(D^T)^{-1}$ zeros.

Then from (7') and by realizing that $[G_1]_{s_1 \times m} = [(D^T)^{-1}]_{s_1 \times m}$, where the $s_1 \times m$ denotes the upper left s_1 by m submatrix,

$$(15) \quad (a_j^*)_{S.C.S.} = \sum_{i=1}^m d_i (D^T)^{-1}_{ij} = \sum_{i=1}^{s_1} f_i [(G_1)_{ij}]_{m \times m},$$

where the f_i in (15) are s_1 elements on the r.h.s. of (5).

Once more, from (9), we write

$$(16) \quad E_{m+1}^{-1} B^{-1} = E_m \cdots E_{s_1+1} G_1 = P \quad (\text{say}).$$

By a similar argument, the premultiplication of G_1 by $(E_m \cdots E_{s_1+1})$, will leave the first m elements of column j in G_1 unaltered. Thus from (10) and (16),

$$(17) \quad (a_j^*)_{C.S.} = \sum_{i=1}^m f_i [P_{ij}]_{m \times m} = \sum_{i=1}^{s_i} f_i [(G_1)_{ij}]_{m \times m}.$$

Comparing (15) and (17) completes the proof of the lemma.

Let us assume that Figures 1 and 2 relate to the same problem. Then since in column 1 of Figure (2c), the position of the two zeros correspond to the position of the two zeros of b_B in Figure (1a), by Lemma 5, $(a_1^*)_{S.C.S.} = (a_1^*)_{C.S.}$.

Such zero elements may also occur in matrix G_i of system $C^{(i)}a^{(i)} = f^{(i)}$, $i = 2, 3, \dots$, and the same results are obtained. That is $(a_j^*)_{S.C.S.} = (a_j^*)_{C.S.}$ for some j , where the C.S. here refers to the C.S. of such systems.

Let us define an indicator set for vector a , as the index set $I(a) \subset [1, 2, \dots, m]$, with the property that $[(a_i^*)_{S.C.S.} = (a_i^*)_{C.S.} | i \in I(a)]$.

2.4. *The System $C^{(2)}a^{(2)} = f^{(2)}$.* Lemma 5 may be applied to G_1 and the index set $I(a)$ is obtained. For every $j \in I(a)$, the term $c_{ij}a_j$ on the l.h.s. of each equation in the system is transferred to the r.h.s. If the number of elements in $I(a)$ equals s_1 , the set of equations (6) would be readily available. If the number of elements in $I(a) = l < s_1$, $(s_1 - l)$ Gauss elimination processes with partial pivoting would be performed to (1), and system (6) would be available. The same is applied in calculating any system $C^{(i)}a^{(i)} = f^{(i)}$, $i = 3, 4, \dots$.

Example 2. Consider the example of twenty-five equations in ten unknowns whose S.C.S. is calculated in [4, p. 698].

For this example, the linear programming problem has one optimal solution b_B which has one zero element. It is also found that eight columns of G_1 each has a zero element in the position of the zero element of b_B . This indicates that $(a_j^*)_{S.C.S.} = (a_j^*)_{C.S.}$, for j corresponding to these columns. Namely for $j = 1, 2, 4, 5, 6, 7, 9$ and 10.

The reduced system (6) is obtained as described in Section 2.4. It requires one Gauss elimination step only. About 160 multiplications are needed by our method. Yet it needs about 900 multiplications by the method [4] to obtain system (6) plus about another 300 multiplications to calculate the eight elements of $(a^*)_{S.C.S.}$ from Eq. (7).

3. Numerical Results. The algorithm described above may be illustrated by the following example.

Example 3 [4, p. 697].

$$a_1 + a_3 = 1, \quad a_2 = 1, \quad a_1 - a_2 + a_3 = 1, \quad a_3 = 3,$$

$$2a_3 = 0, \quad a_1 - a_2 - a_3 = -4, \quad 2a_1 - a_2 = 1.$$

After rearranging the elements of b_B and the corresponding rows of B^{-1} , as in Figure (1a, b), also after exchanging columns 1 and 3 of B^{-1} , b_B and B^{-1} are given by

b_B		B^{-1}		
.33	-.33	.33	.33	.33
0	0	-1	-1	0
0	0	0	-1	0
.67	.33	.67	1.67	.67

Here b_B is degenerate. Columns 5, 2, 6 and 4 form the basis. By examining the simplex tableau, we find that no nonbasic column with zero marginal cost can enter the basis with a positive level. Hence, R_1 has rank $s_1 = 1$ and the C.S. of (1) is not unique with $(a^*)_{C.S.} = (2, 3, 1)^T$ and $z_1^* = 2$. Also it is noticed that there exist two zeros in column 1 of B^{-1} which correspond to the two zeros of b_B . Column 1 of B^{-1} corresponds to a_3 . Hence according to Lemma 5, $(a_3^*)_{S.C.S.} = (a_3^*)_{C.S.} = 1$.

Since $s_1 = 1$ and one element of $(a^*)_{S.C.S.}$ is now known, system (6) is readily available as explained in Section 2.4. It is given by

$$a_1 = 0, \quad a_2 = 1, \quad a_1 - a_2 = 0, \quad a_1 - a_2 = -3, \quad 2a_1 - a_2 = 1.$$

The vector b_B of this system has one zero element with columns 7, 6 and 3 forming the basis and $z_2^* = 1.5$. However, the nonbasic column 2 has a zero marginal cost and can replace column 3 in the basis, with a positive level. The obtained optimal solution b'_B is not degenerate. Hence according to Lemma 1, the solution of this reduced system is unique. Matrix $(D^T)^{-1}$ is then calculated as described at the end of Section 2.2 and the S.C.S. is obtained from (7'). The final result is $(a^*)_{S.C.S.} = (1, 2.5, 1)^T$.

A computer program for the present algorithm is coded in Fortran IV and is used in calculating the S.C.S. of several test problems on the IBM 360/67 computer.

A tolerance δ is specified. A calculated parameter x is considered zero if $|x| < \delta$. For the IBM 360/67 computer, the round-off error level is about 10^{-6} for single precision and about 10^{-16} for double precision calculation. For this computer we take $\delta = 10^{-4}$ and 10^{-11} respectively.

The main purpose of the results in Table 1 below is to compare the present algorithm with an earlier version. In the earlier version no attempt was made to obtain the characteristic set R_1 by using the simplified procedure. As a result the sets R_i are determined in steps. We call each of these steps a major iteration. In Table 1, the data and the number of points are taken from a recent paper by Watson [9, Table 2], who used this data for a different purpose.

Given are the number of major iterations, the total number of iterations which is the total number of times the simplex tableau is changed and the execution (CPU) time in seconds. This calculation is done in double precision.

The present method is seen to be superior over the earlier version in ten out of fourteen examples.

In Duris and Temple's algorithm [4] however, the exchange rule of Duris [3] is used in obtaining the C.S. As pointed out by the referee, this exchange rule is much superior to the linear programming exchange rule [1] used here. This has been confirmed by our calculation. Using the routine [4], the number of major iterations is

TABLE 1

Example	$C(n \times m)$	Earlier Version			Present Method		
		Major Iter	Total Iter	Time (Sec.)	Major Iter	Total Iter	Time (sec.)
1a	16×9	3	32	.19	3	32	.19
b	25×16	6	89	.88	6	89	.89
2a	16×9	5	44	.29	3	24	.15
3a	16×9	3	40	.26	3	40	.26
4a	16×9	4	34	.19	4	34	.20
5a	16×6	8	40	.18	2	12	.07
b	25×10	13	87	.61	2	20	.21
c	36×15	19	185	2.47	3	38	.75
6a	16×6	8	42	.20	3	20	.10
b	25×10	10	85	.61	3	33	.30
c	36×15	15	163	2.28	4	56	1.03
7a	16×6	5	35	.16	2	17	.09
b	25×10	10	97	.71	3	35	.33
c	36×15	15	185	2.50	4	58	1.07

the same as those of the present method for all but two of the above examples, where it is slightly higher. The number of total iterations are significantly smaller than ours. However, the CPU times of the routine [4], for the above examples, are between 1.5 and 3.3 times those of the present method. It would be worthwhile attempting to use Duris' exchange rule in the present routine. The present method might be even faster.

4. Concluding Remark. For the rank deficient cases, the columns of matrix C which are linearly dependent on other columns are detected while obtaining the C.S. by the algorithm [1] and are deleted. The parameters a_i associated with these columns are set equal to zero. In these cases, the calculated S.C.S. would be for the overdetermined system whose coefficient matrix C consists of the linearly independent columns of the given coefficient matrix.

5. Acknowledgements. I am grateful to Dr. C. S. Duris for sending me a deck of cards for his and Dr. Temple's routine. I am also indebted to the referee for his remarks, as a result of which this paper is significantly improved. Finally I wish to acknowledge the programming effort of Ms. Olga L. Lapczak, in calculating the L_p solutions used in checking the small size problems.

Division of Electrical Engineering
National Research Council
Ottawa, Ontario, Canada

2. J. DESCLOUX, "Approximations in L^p and Chebyshev approximations," *J. Soc. Indust. Appl. Math.*, v. 11, 1963, pp. 1017–1026. MR 28 #2389.
3. C. S. DURIS, "An exchange method for solving Haar and non-Haar overdetermined linear equations in the sense of Chebyshev," *Proc. ACM Nat. Conf.*, 1968, pp. 61–66.
4. C. S. DURIS & M. G. TEMPLE, "A finite step algorithm for determining the "strict" Chebyshev solution to $Ax = b$," *SIAM J. Numer. Anal.*, v. 10, 1973, pp. 690–699. MR 48 #7574.
5. G. HADLEY, *Linear Programming*, Addison-Wesley, Reading, Mass., 1962. MR 24 #B1669
6. M. R. OSBORNE & G. A. WATSON, "On the best linear Chebyshev approximation," *Comput. J.*, v. 10, 1967, pp. 172–177. MR 36 #1892.
7. J. R. RICE, "Tchebycheff approximation in a compact metric space," *Bull. Amer. Math. Soc.*, v. 68, 1962, pp. 405–410. MR 25 #3313.
8. J. R. RICE, *The Approximation of Functions*. Vol. 2, Addison-Wesley, Reading Mass., 1969. MR 39 #5989.
9. G. A. WATSON, "A multiple exchange algorithm for multivariate Chebyshev approximation," *SIAM J. Numer. Anal.*, v. 12, 1975, pp. 46–52. MR 51 #9430.