LINEAR ALGEBRA AND ITS APPLICATIONS 7, 99–138 (1973)

A Numerically Stable Form of the Simplex Algorithm*

PHILIP E. GILL AND WALTER MURRAY
National Physical Laboratory
Teddington, Middlesex, England

Communicated by J. H. Wilkinson

ABSTRACT

Standard implementations of the Simplex method have been shown to be subject to computational instabilities, which in practice often result in failure to achieve a solution to a basically well-determined problem. A numerically stable form of the Simplex method is presented with storage requirements and computational efficiency comparable with those of the standard form. The method admits non-Simplex steps and this feature enables it to be readily generalized to quadratic and nonlinear programming. Although the principal concern in this paper is not with constraints having a large number of zero elements, all necessary modification formulae are given for the extension to these cases.

1. INTRODUCTION

This paper is concerned with the solution of the following linear programming problem:

$$\min\{z = c^T x\}, \tag{P1}$$

subject to the constraints

$$A^T x \geqslant b,$$

where A is an $n \times m$ matrix, with $m \geqslant n$.

* This paper was presented at the 7th Mathematical Programming Symposium 1970, The Hague, The Netherlands.

The problem is stated in this nonstandard form in order to emphasize the close relationship between linear and nonlinear programming problems. Initially no reference is made to the form of the constraints but later special consideration will be given to constraints of the form $x \geqslant b$, which are common in practical problems.

It is assumed that

(i) there exists at least one $x \in E^n$ for which

$$A^T x \geqslant b,$$

(ii) z is bounded in the feasible region, and
(iii) A and $[A \vdots c]$ satisfy the Haar condition for matrices.

The final condition ensures that degeneracy and cycling cannot occur during the Simplex algorithm.

It is unfortunate that the terminology associated with linear programming has served to isolate the subject from the mainstream of linear algebra and numerical analysis. It is not always appreciated by linear programming practitioners that any viable form of the Simplex algorithm ought to be numerically stable when applied to the problem of solving a set of linear algebraic equations. The form of the algorithm in the majority of linear programming implementations differs little from that given by Dantzig in 1947 [5]. Any changes that have been made have been concerned only with the manner in which the form of the algorithm is stored inside the computer, while aspects of the numerical errors involved have generally received little mention in the research literature.

The solution of a linear program consists of two stages:

(i) The identification of the set of constraints active at the solution.
(ii) The determination of the vertex defined by the set of active constraints.

Stage (ii) corresponds to the solution of a set of n equations in n unknowns and consequently any algorithm for the solution of a linear program must incorporate a method for the solution of a set of linear algebraic equations. The standard form of the Simplex algorithm embodies the Gauss-Jordan elimination process with the pivots chosen without regard to rounding error. In solving linear equations by Gaussian elimination it is essential for numerical stability that large pivots are chosen. Wilkinson [12] gives an example to illustrate this fact. Suppose the solution to the following set of equations is required

$$Ax = b,$$

where

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

When applied to this problem Gaussian elimination without any choice of pivots breaks down, despite the fact that A is an orthogonal matrix and consequently well conditioned. Avoiding zero pivots by requiring them to be larger than some threshold ε does not solve the problem, since A could just as easily be of the form

$$A = \begin{bmatrix} \varepsilon & 1 - \varepsilon \\ 1 + \varepsilon & -\varepsilon \end{bmatrix},$$

in which case $A = A^{-1}$. Although Gaussian elimination does not break down in this case, the small pivotal elements obtained lead to gross errors in the solution. Recently Bartels and Golub [1, 2] have drawn attention to the instabilities inherent in the standard Simplex algorithm and they have devised two alternative forms that exhibit numerical stability. The implementation of their methods into linear programming packages has been slow if not nonexistent. This could be attributed to certain disadvantages in terms of the storage and/or computational efficiencies which the methods have in comparison with the standard method. The basis of both procedures is a recursion from iteration to iteration of the triangular decomposition of the matrix of coefficients of the active constraints. Partial pivoting is used when updating the triangular factors, but unfortunately the simple form of the matrices carried from the previous iteration is then largely destroyed.

All the methods considered in this section so far have been based upon the factorization of a matrix into a product of upper and lower triangular matrices. The method proposed in this paper utilizes a factorization into the product of a lower triangular and orthogonal matrix. This factorization always exists and can be made without interchanging rows in the initial matrix.

2. THE SOLUTION OF LINEAR EQUATIONS USING ORTHOGONAL TRANSFORMATIONS

A vertex of the feasible space, and consequently the normalized directed distance between two vertices, can be defined by the solution of n linear equations in n unknowns. The normalized direction from a point other than a vertex to a point in the feasible region where the objective function is decreased can be defined by a solution of an underdetermined set of linear equations.

This point is of particular relevance in Sec. 10 where non-Simplex steps are considered.

Consider the system of equations

$$A^T y = b, \tag{1}$$

where A^T is now an $s \times t$ rectangular matrix with rank s, $s \leqslant t$.

The matrix A^T can be reduced to lower triangular form through successive postmultiplications by a sequence of elementary unitary matrices of either the Givens' or Householder type.

Then

$$A^T W_1 W_2 \cdots W_s = [L \vdots 0],$$

with $W_i^T W_i = I$, for $i = 1, \ldots, s$, and L a lower triangular matrix. Define

$$W_1 \cdots W_s = P,$$

then

$$A^T = [L \vdots 0] P^T. \tag{2}$$

A solution of Eq. (1) can be found from a forward substitution of the system

$$[L \vdots 0] x = b, \tag{3}$$

with $x_{s+1}, \ldots, x_t$ arbitrary and forming

$$y = Px. \tag{4}$$

Alternatively if $\bar{y}$ is found using the following equations,

$$LL^T \omega = b \qquad \text{and} \qquad \bar{y} = A\omega, \tag{5}$$

then $\bar{y}$ is a solution of (1). The two solutions (4) and (5) are identical if $s = t$ or $x_{s+1}, \ldots, x_t = 0$. The error analysis of the factorization and forward substitution has been given by Wilkinson [12]. Gaussian elimination is usually preferred when solving linear equations since it takes approximately half the computational effort of the Householder triangularization and a quarter that of Givens' method.

If A^T has any special form, elementary unitary transformations may be computationally more efficient since the effect of row permutations can radically alter the structure of A^T. In particular, Secs. 5 and 6 describe how Givens- and Householder-type reductions can be used to recur lower triangular matrices from one iteration to the next.

3. A BASIC ITERATION FOR TAKING SIMPLEX STEPS

At the beginning of the ith iteration the following matrices and vectors are available:

(i) $A^{(i)}$ an $n \times n$ matrix,
(ii) $\bar{A}^{(i)}$ an $n \times (m - n)$ matrix,
(iii) $b^{(i)}$ an $n \times 1$ column vector,
(iv) $\bar{b}^{(i)}$ an $(m - n) \times 1$ column vector,
(v) $x^{(i)}$ an $n \times 1$ column vector,

where the matrix of constraints is partitioned in the form

$$A^T = \begin{bmatrix} A^{(i)T} \\ \hdashline \bar{A}^{(i)T} \end{bmatrix}.$$

The rows of $A^{(i)T}$, $\bar{A}^{(i)T}$ are labelled $1(1)n$, $1(1)m - n$ respectively, giving

$$b = \begin{bmatrix} b^{(i)} \\ \bar{b}^{(i)} \end{bmatrix},$$

with

$$A^{(i)T}x^{(i)} = b^{(i)},$$

and

$$\bar{A}^{(i)T}x^{(i)} > \bar{b}^{(i)}.$$

(vi) Further, $L^{(i)}$ is a lower triangular matrix such that

$$L^{(i)}L^{(i)T} = A^{(i)T}A^{(i)};$$

(vii) and $$d^{(i)} = A^{(i)T}c$$

(viii) and $$\omega^{(i)} = \bar{A}^{(i)T}x^{(i)} - \bar{b}^{(i)}.$$

Step 1

Determine the Kuhn-Tucker multipliers, u, associated with the active constraints by solving the equations

$$L^{(i)}L^{(i)T}u = d^{(i)}.$$

(a) If $u_j > 0$, $j = 1, \ldots, n$ then $x^{(i)}$ is the optimal solution.
(b) If some $u_j < 0$ then choose an index q such that

$$u_q = \min\{u_j : j = 1, \ldots, n\}.$$

Step 2

Determine p^T, the qth row of $(A^{(i)})^{-1}$, by solving the equations

$$L^{(i)}L^{(i)T}y = e_q,$$

where e_q is the qth column of the identity matrix, then

$$p = A^{(i)}y.$$

Step 3

Determine the index k such that

$$\lambda_k = -\frac{\omega_k^{(i)}}{v_k} = \min_{v_j<0}\left\{-\frac{\omega_j^{(i)}}{v_j}\right\},$$

where

$$v = \bar{A}^{(i)T}p.$$

Step 4

Set

$$x^{(i+1)} = x^{(i)} + \lambda_k p,$$

$$d_j^{(i+1)} = d_j^{(i)}, \qquad j < q,$$

$$d_j^{(i+1)} = d_{j+1}^{(i)}, \qquad n > j \geqslant q,$$

$$d_n^{(i+1)} = \bar{a}_k^T c, \tag{6}$$

where $\bar{a}_k$ is the kth column of $\bar{A}^{(i)}$. Update the residuals of the inactive constraints

$$\omega_j^{(i+1)} = \omega_j^{(i)} + \lambda_k v_j, \qquad 1 \leqslant j \leqslant m - n, \quad j \neq k$$

$$\omega_k^{(i+1)} = \lambda_k.$$

Step 5

The qth column of $A^{(i)}$ is removed to become the kth column of $\bar{A}^{(i+1)}$. $A^{(i+1)}$ is formed by relabelling the remaining columns of $A^{(i)}$ from 1 to $n - 1$ and adding $\bar{a}_k$ in the nth position. Similarly, the qth element of $b^{(i)}$ is removed and placed in the kth position of $\bar{b}^{(i+1)}$. $b^{(i+1)}$ is formed by relabelling the remaining elements of $b^{(i)}$ from 1 to $n - 1$ and adding $\bar{b}_k^{(i)}$ in the nth position.

Step 6

The lower triangular factors $L^{(i)}$ are modified in two stages. When the qth column of $A^{(i)}$ is removed an intermediate lower triangular matrix $\mathscr{L}$ is found by one of the methods given in Sec. 5 with $t = s = n$. This matrix is in turn modified by the method given in Sec. 6 when $\bar{a}_k$ is added in the nth position.

4. PROOF OF CONVERGENCE

Let $\{a_j: 1 \leqslant j \leqslant n\}$ be the n columns of $A^{(i)}$. It can be seen from the choice of p in Eq. (6) that

$$c^T x^{(i+1)} = c^T x^{(i)} + \lambda_k c^T p.$$

Substituting for p gives

$$c^T x^{(i+1)} = c^T x^{(i)} + \lambda_k c^T A^{(i)} y,$$

$$= c^T x^{(i)} + \lambda_k d^{(i)T} y,$$

$$= c^T x^{(i)} + \lambda_k u^T L^{(i)} L^{(i)T} y,$$

$$= c^T x^{(i)} + \lambda_k u^T e_q,$$

$$= c^T x^{(i)} + \lambda_k u_q.$$

Consequently, since $u_q < 0$ and $\lambda_k > 0$

$$c^T x^{(i+1)} < c^T x^{(i)}.$$

If all $u_i > 0$ no improvement can be made to the objective function. The new point is always feasible since

$$\text{for} \quad \begin{cases} 1 \leqslant j \leqslant n, \\ j \neq q, \end{cases} \qquad \begin{aligned} a_j^T x^{(i+1)} &= a_j^T x^{(i)} + \lambda_k a_j^T p; \\ &= a_j^T x^{(i)}, \end{aligned}$$

and we have

$$a_j^T x^{(i+1)} = b_j.$$

For $j = q$ we have

$$\begin{aligned} a_q^T x^{(i+1)} &= a_q^T x^{(i)} + \lambda_k a_q^T p, \\ &= b_q + \lambda_k, \\ &> b_q \quad \text{since} \quad \lambda_k > 0. \end{aligned}$$

5. MODIFICATION OF THE TRIANGULAR FACTORS OF A^TA WHEN A COLUMN IS REMOVED FROM A

The orthogonal triangularization of a rectangular matrix A^T is given by

$$A^T = [L \vdots 0] P^T, \qquad P^T P = I,$$

where L is an $s \times s$ lower triangular matrix, A^T an $s \times t$ matrix and P a $t \times t$ orthogonal matrix with $s \leqslant t$. Then

$$\begin{aligned} A^T A &= [L \vdots 0] P^T P \begin{bmatrix} L^T \\ \cdots \\ 0 \end{bmatrix}, \\ &= L L^T. \end{aligned}$$

If $a_1, \ldots, a_s$ are the s linearly independent columns of A, then the matrix

$\mathscr{A}$ obtained when a column a_k is removed from A is given by

$$\mathscr{A} = [a_1 \vdots a_2 \vdots \cdots \vdots a_{k-1} \vdots a_{k+1} \vdots \cdots \vdots a_s]$$

and

$$\mathscr{A}^T\mathscr{A} = \bar{L}\bar{L}^T,$$

where $\bar{L}$ is a $(s - 1) \times s$ lower Hessenberg matrix with zeros above the diagonal in the first $k - 1$ rows. Given the matrix $\bar{L}$ with elements $\bar{l}_{ij}$ three methods are given to obtain $\mathscr{L}$, the lower triangular factor of $\mathscr{A}^T\mathscr{A}$.

5.1. Method A: Elementary Hermitian Matrices

An orthogonal matrix W can be constructed such that

$$\mathscr{A}^T\mathscr{A} = \bar{L}W^TW\bar{L}^T,$$

$$= \mathscr{L}\mathscr{L}^T,$$

where $\mathscr{L}$ is the $(s - 1) \times (s - 1)$ lower triangular matrix associated with the symmetric triangular decomposition of $\mathscr{A}^T\mathscr{A}$.

For $r = k, k + 1, \ldots, s - 1$ the matrices W_r are defined as follows

$$\bar{L}^{(r+1)} = \bar{L}^{(r)}W_r,$$

where

$$\bar{L}^{(k)} = \bar{L},$$

and

$$W_r = I - \alpha_r\omega^{(r)}\omega^{(r)T}.$$

Let

$$\alpha_r = [S_r^2 \mp \bar{l}_{r,r}^{(r)}S_r]^{-1},$$

where

$$S_r^2 = \bar{l}_{r,r}^{(r)2} + \bar{l}_{r,r+1}^{(r)2}.$$

Since the columns of A are linearly independent, $\bar{l}_{r,r+1}^{(r)2}$ is nonzero and consequently S_r is nonzero. The column vector $\omega^{(r)}$ has two nonzero components,

$$\omega_r^{(r)} = \tilde{l}_{r,r}^{(r)} \mp S_r,$$

and

$$\omega_{r+1}^{(r)} = \tilde{l}_{r,r+1}^{(r)},$$

the sign of S_r being chosen to minimize rounding error. The W_r are symmetric matrices which transform the elements $\tilde{l}_{r,r+1}^{(r)}$ of $\tilde{L}^{(r)}$ into zero elements of $\tilde{L}^{(r+1)}$. During the reduction by a particular W_r only columns $r, r+1$ of $\tilde{L}^{(r)}$ are modified due to the sparsity of the $\omega^{(r)}$ vectors. The product of the W_r's is an orthogonal matrix (though not symmetric).

If the product is written as

$$W_k W_{k+1} \cdots W_{s-1} = W,$$

then

$$\tilde{L}W = \tilde{L}^{(s)},$$

where $\tilde{L}^{(s)}$ is a lower triangular matrix plus a null last column, i.e.

$$\tilde{L}^{(s)} = [\mathscr{L} \vdots 0].$$

5.2. Method B: Elementary Unitary Matrices

The reduction of $\tilde{L}$ to lower triangular form can also be obtained using elementary unitary matrices. Again, an orthogonal matrix Q is constructed such that

$$\begin{aligned} \mathscr{A}^T\mathscr{A} &= \tilde{L}\tilde{L}^T, \\ &= \tilde{L}Q^TQ\tilde{L}^T, \\ &= \mathscr{L}\mathscr{L}^T. \end{aligned}$$

The matrices Q_r, each of dimension $s \times s$, are defined as follows

$$\tilde{L}^{(r+1)} = \tilde{L}^{(r)}Q_r, \qquad r = k, k+1, \ldots, s-1,$$

with

$$\tilde{L}^{(k)} = \tilde{L},$$

where

$$Q_r = \begin{bmatrix} 1 & & & & & & & & & \\ & 1 & & & & & & & & \\ & & \ddots & & & & & & & \\ & & & 1 & & & & & & \\ & & & & 1 & & & & & \\ & & & & & c & -s & & & \\ & & & & & s & c & & & \\ & & & & & & & 1 & & \\ & & & & & & & & 1 & \\ & & & & & & & & & \ddots \\ & & & & & & & & & & 1 \\ & & & & & & & & & & & 1 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ r \\ r+1 \\ \\ \\ \\ \\ \\ \end{matrix}$$

$$\qquad\qquad r \quad r+1$$

and $c = \cos\theta$, $s = \sin\theta$. θ is chosen such that the upper diagonal element corresponding to $\tilde{l}^{(r)}_{r,r+1}$ in $\tilde{L}^{(r)}$ is zero in $\tilde{L}^{(r+1)}$, i.e.

$$-s\tilde{l}^{(r)}_{r,r} + c\tilde{l}^{(r)}_{r,r+1} = 0.$$

If

$$\sigma = \tan\theta,$$

then

$$\sigma = \tilde{l}^{(r)}_{r,r+1}/\tilde{l}^{(r)}_{r,r}.$$

The quantities $\cos\theta$ and $\sin\theta$ can be simply calculated using

$$\rho = (\tilde{l}^{(r)2}_{r,r} + \tilde{l}^{(r)2}_{r,r+1})^{1/2},$$

$$c = \frac{\tilde{l}^{(r)}_{r,r}}{\rho}, \qquad s = \frac{\tilde{l}^{(r)}_{r,r+1}}{\rho}.$$

The unitary matrix Q_r, which is a rotation in the $(r, r+1)$ plane, becomes a permutation matrix when $\tilde{l}^{(r)}_{r,r}$ is zero. Only two columns of $\tilde{L}^{(r)}$ are modified during a transformation by a particular Q_r. The final matrix obtained by recurrence is then of the form

$$\tilde{L}^{(s)} = [\mathscr{L} \,\vdots\, 0].$$

5.3. Method C: Elementary Nonunitary Matrices

The elementary nonunitary matrices M_r, $r = k, k+1, \ldots, s-1$, each with dimension $s \times s$ are defined as follows:

$$\tilde{L}^{(r+1)} = \tilde{L}^{(r)} M_r, \tag{7}$$

with

$$\tilde{L}^{(k)} = \tilde{L},$$

where

$$M_r = \left[\begin{array}{c:c:c:c} \begin{matrix} 1 & & & & \\ & 1 & & & \\ & & \cdots & & \\ & & & 1 & \\ & & & & 1 \end{matrix} & & & \\ \hdashline & 1 & -m_{r,r+1} & \\ \hdashline & 0 & 1 & \\ \hdashline & & & \begin{matrix} 1 & & & & \\ & 1 & & & \\ & & \cdots & & \\ & & & 1 & \\ & & & & 1 \end{matrix} \end{array}\right],$$

and $m_{r,r+1}$ is chosen such that the upper diagonal element $\tilde{l}^{(r)}_{r,r+1}$ of $\tilde{L}^{(r)}$ is transformed into the zero element of $\tilde{L}^{(r+1)}$.

We choose

$$\tilde{l}^{(r)}_{r,r+1} - \tilde{l}^{(r)}_{r,r} m_{r,r+1} = 0,$$

giving

$$m_{r,r+1} = \tilde{l}^{(r)}_{r,r+1} / \tilde{l}^{(r)}_{r,r}. \tag{8}$$

It is assumed for the moment that $\tilde{l}^{(r)}_{r,r}$ is nonzero. The M_r modify only one column of $\tilde{L}^{(r)}$. The product of the M_r can be written as

$$M = M_k M_{k+1} \cdots M_{s-2} M_{s-1}$$

and

$$\tilde{L}^{(s)} = \tilde{L} M. \tag{9}$$

The decomposition can be written as

$$\mathscr{A}^T \mathscr{A} = \tilde{L} \tilde{L}^T,$$

$$= \tilde{L}MM^{-1}(M^T)^{-1}M^T\tilde{L}^T,$$

$$= \tilde{L}^{(s)}M^{-1}(M^T)^{-1}\tilde{L}^{(s)T}. \tag{10}$$

Now

$$M^{-1} = \begin{bmatrix} I_{k-1} & 0 \\ 0 & N \end{bmatrix},$$

where I_{k-1} is the $(k-1) \times (k-1)$ identity matrix and

$$N = \begin{bmatrix} 1 & m_{k,k+1} & & & \\ & 1 & m_{k+1,k+2} & & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ & & & 1 & m_{s-1,s} \\ & & & & 1 \end{bmatrix}. \tag{11}$$

Then we have

$$M^{-1}(M^T)^{-1} = \begin{bmatrix} I_{k-1} & 0 \\ 0 & T \end{bmatrix},$$

where T is a symmetric positive definite tridiagonal matrix of the form

$$T = \begin{bmatrix} 1 + m_{k+1,k}^2 & m_{k+1,k} & & & \\ m_{k+1,k} & 1 + m_{k+2,k+1}^2 & m_{k+2,k+1} & & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ & m_{s-1,s-2} & 1 + m_{s,s-1}^2 & m_{s,s-1} \\ & & m_{s,s-1} & 1 \end{bmatrix}.$$

The symmetric decomposition of T can be formed giving

$$T = RR^T,$$

where R is a lower triangular matrix of the form

$$R = \begin{bmatrix} r_1 & & \\ t_1 & r_2 & \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ & t_{s-k} & r_{s-k+1} \end{bmatrix}.$$

The elements of R can be found using the recurrence relations

$$r_1^2 = 1 + m_{k+1,k}^2, \tag{12}$$

$$\left.\begin{aligned} r_i^2 + t_{i-1}^2 &= 1 + m_{k+i,k+i-1}^2 \\ r_i t_i &= m_{k+i,k+i-1} \end{aligned}\right\} i = 2, \ldots, s-k. \tag{13, 14}$$

The product of these decompositions gives a lower triangular decomposition of $\mathscr{A}^T\mathscr{A}$

$$\mathscr{L} = \bar{L}^{(s)} \left[\begin{array}{c|c} I_{k-1} & 0 \\ \hline 0 & R \end{array}\right].$$

The assumption that $\bar{l}_{r,r}^{(r)}$ is nonzero in Eq. (8) is in general invalid. The breakdown of attempts to modify the triangular factors of the identity matrix when a column is deleted indicates that zero divisors can occur even when the problem is relatively simple. In addition, the occurrence of very large elements in the elementary matrices can produce adverse effects upon the stability of the method. These problems can be solved by using column interchanges during the reduction process. A column interchange is made to obtain the larger of the two elements $\bar{l}_{r,r+1}^{(r)}, \bar{l}_{r,r}^{(r)}$ as divisor in Eq. (8).

The reduction (9) then becomes

$$\bar{L} I_k M_k I_{k+1} \cdots I_{s-1} M_{s-1} = \bar{L}^{(s)},$$

where I_j is a permutation matrix which interchanges columns $(j, j+1)$. Then

$$\bar{L} = \bar{L}^{(s)} \bar{M}_{s-1} \bar{M}_{s-2} \cdots \bar{M}_k \Pi,$$

where

$$\begin{aligned} \bar{M}_{s-1} &= M_{s-1}^{-1}, \\ \bar{M}_{s-2} &= I_{s-1} M_{s-2}^{-1} I_{s-1}, \\ &\vdots \\ \bar{M}_k &= I_{s-1} \cdots I_{k+1} M_k^{-1} I_{k+1} \cdots I_{s-1}, \end{aligned}$$

and

$$\Pi = I_{s-1} I_{s-2} \cdots I_{k+1} I_k.$$

Suppose no interchange of the columns is necessary for the calculation of M_i. Then

$$\bar{M}_j = I_{s-1} \cdots I_{i+1} I_{i-1} \cdots I_{j+1} M_j^{-1} I_{j+1} \cdots I_{i-1} I_{i+1} \cdots I_{s-1},$$

$$\text{for all} \quad j = k, \ldots, i-1,$$

$$= I_{i-1} \cdots I_{j+1} M_j I_{j+1} \cdots I_{i-1}.$$

The $\bar{M}_j$ are consequently unaffected by succeeding permutations of columns and instead of Eq. (11), the inverse product of the transformations is then

$$N = \begin{bmatrix} G_{k_1} & & & \\ & G_{k_2} & & \\ \cdots & \cdots & \cdots & \cdots \\ & & G_{k_\alpha} & \\ & & & 1 \end{bmatrix}, \tag{15}$$

where the G_{k_β} are $k_\beta \times (k_\beta + 1)$ submatrices with the structure

$$G_{k_\beta} = \begin{bmatrix} 1 & & & g_1 \\ & 1 & & g_2 \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ & & 1 & g_{k_\beta} \end{bmatrix}, \tag{16}$$

and $\sum k_i + 1 = s$. k_α indicates that there have been k_α permutations of a particular column, the last of which is the identity permutation. Since the matrix T is no longer tridiagonal the recurrence relations (12), (13) and (14) cannot be used.

Givens-type rotations can be used to reduce M^{-1} to lower triangular form. Equation (10) is then written

$$\tilde{L}\tilde{L}^T = \tilde{L}^{(s)} M^{-1} Q^T Q (M^{-1})^T \tilde{L}^{(s)T},$$

where $Q^T Q = I$ and $M^{-1} Q^T = R$ a lower triangular matrix whose columns are multiples of the nonzero columns of G_{k_β} in Eq. (16). The new lower triangular factor is

$$\mathscr{L} = \tilde{L}^{(s)} \left[\begin{array}{c|c} I_{k-1} & 0 \\ \hline 0 & R \end{array} \right].$$

$\mathscr{L}$ can be calculated in $O(s^2)$ operations using a technique similar to that described in Sec. 7.

6. MODIFICATION OF THE TRIANGULAR FACTORS OF $\mathscr{A}^T\mathscr{A}$ WHEN A COLUMN IS ADDED TO $\mathscr{A}$

If $\mathscr{A}^T$ is an $(s - 1) \times t$ matrix with

$$\mathscr{A}^T\mathscr{A} = \mathscr{L}\mathscr{L}^T$$

then there exists an orthogonal matrix $\mathscr{P}$ such that

$$\mathscr{A}^T = [\mathscr{L} \vdots 0]\mathscr{P}^T.$$

The new row a^T can be inserted anywhere within $\mathscr{A}^T$ with the effect of adding a new row l^T to the lower triangular factor $\mathscr{L}$. If the new row is added at the end of $\mathscr{A}^T$ giving

$$A^T = \begin{bmatrix} \mathscr{A}^T \\ \hdashline a^T \end{bmatrix},$$

then the amount of computational effort required for the modification of the lower triangular factors is minimized and

$$\begin{aligned} A^T &= \left[\begin{array}{c:c} \begin{array}{c:c} \mathscr{L} & 0 \\ \hdashline \multicolumn{2}{c}{l^T} \end{array} & 0 \end{array}\right] P^T, \\ &= [L \vdots 0]P^T, \end{aligned}$$

for some orthogonal matrix P. Now

$$A^TA = \left[\begin{array}{c:c} \mathscr{A}^T\mathscr{A} & 0 \\ \hdashline 0 & 0 \end{array}\right] + \left[\begin{array}{c:c} 0 & \mathscr{A}^Ta \\ \hdashline a^T\mathscr{A} & a^Ta \end{array}\right], \tag{17}$$

and also

$$LL^T = \left[\begin{array}{c:c} \mathscr{L}\mathscr{L}^T & 0 \\ \hdashline 0 & 0 \end{array}\right] + \left[\begin{array}{c:c} 0 & [\mathscr{L} \vdots 0]l \\ \hdashline l^T \begin{bmatrix} \mathscr{L}^T \\ \hdashline 0 \end{bmatrix} & l^Tl \end{array}\right]. \tag{18}$$

Comparison of Eq. (17) with Eq. (18) gives l as the solution of the equations

$$[\mathscr{L} \vdots 0]l = \mathscr{A}^Ta, \tag{19}$$

$$l^T l = a^T a. \tag{20}$$

Since $\mathscr{L}$ is lower triangular, all but one element of l can be found by forward substitution in Eq. (19) and the final element using Eq. (20).

7. MODIFICATION OF THE TRIANGULAR FACTORS OF A^TA WHEN A ROW IS REMOVED FROM A

7.1. *Method A: Elementary Nonunitary Matrices*

Let A^T be an $s \times (t+1)$ matrix and $\bar{A}^T$ the resulting matrix when the column a is removed from A^T. Then

$$\bar{A}^T\bar{A} = A^TA - aa^T.$$

$\bar{A}^T\bar{A}$ is obtained by applying a rank one modification to A^TA.

In general consider the matrix $\bar{B}$ where

$$\bar{B} = B + kgg^T,$$

g is an $s \times 1$ vector, k a scalar, $\bar{B}$ and B are $s \times s$ positive definite symmetric matrices. The triangular factors of $\bar{B}$ are required when the triangular factors of B are given. Bennett [4] has given an algorithm for the modification of the triangular factors of a general matrix under a rank r ($r \geqslant 1$) modification but since the general case is relatively complex, a similar method for symmetric matrices under a rank-one modification is described below. It is emphasized that the calculation of the modified triangular factors for the case considered is numerically stable whereas the general algorithm given by Bennett is not.

Define

$$B^{(1)} = B \quad \text{with} \quad B^{(1)} = [b_{ij}^{(1)}].$$

The matrix $B^{(1)}$ has the symmetric triangular decomposition

$$B^{(1)} = LDL^T,$$

where L is a lower triangular matrix with unit diagonal and D is a diagonal matrix. The triangular decomposition can be found in the following manner. Let $l^{(j)}$ be the $(s-j) \times 1$ vector of elements $(j+1)$ to s of the jth column of L. Let M_j be the identity matrix with elements $(j+1)$ to s of its jth column replaced by $l^{(j)}$.

Then

$$D = M_s^{-1}M_{s-1}^{-1} \cdots M_1^{-1}B^{(1)}(M_1^{-1})^T \cdots (M_s^{-1})^T.$$

Also if $B^{(1)}$ is partitioned in the form

$$B^{(1)} = \begin{bmatrix} b_{11}^{(1)} & v^T \\ v & H \end{bmatrix},$$

the following relation holds

$$M_1^{-1}B^{(1)}(M_1^{-1})^T = \begin{bmatrix} b_{11}^{(1)} & 0 \\ 0 & H - (1/b_{11}^{(1)})vv^T \end{bmatrix},$$

where

$$l^{(1)} = (1/b_{11}^{(1)}) \begin{bmatrix} b_{21}^{(1)} \\ \vdots \\ b_{s1}^{(1)} \end{bmatrix}. \qquad (21)$$

If the $(s-1) \times (s-1)$ matrix $B^{(2)}$ is defined as

$$\begin{aligned} B^{(2)} &= [b_{ij}^{(2)}], \\ &= H - (1/b_{11}^{(1)})vv^T, \end{aligned}$$

then

$$M_1^{-1}B^{(1)}(M_1^{-1})^T = \begin{bmatrix} b_{11}^{(1)} & 0 \\ 0 & B^{(2)} \end{bmatrix}.$$

Corresponding to Eq. (21) $l^{(2)}$ is a multiple of the first column of $B^{(2)}$, i.e.

$$l^{(2)} = (1/b_{11}^{(2)}) \begin{bmatrix} b_{21}^{(2)} \\ \vdots \\ b_{s-1,1}^{(2)} \end{bmatrix}.$$

Generally $l^{(j)}$ is expressed as

$$l^{(j)} = (1/b_{11}^{(j)}) \begin{bmatrix} b_{21}^{(j)} \\ \vdots \\ b_{s-j+1,1}^{(j)} \end{bmatrix}.$$

By considering each new submatrix $B^{(j)}$ and forming $l^{(j)}$ as a multiple of its first column, the complete triangular decomposition can be found.

The quantities $l^{(j)}$ and $B^{(j)}$ can be used to modify the lower triangular factors of B to give those of $\bar{B}$. Define $g = g^{(1)}$, $k = k^{(1)}$ and

$$\bar{B}^{(1)} = [\bar{b}_{ij}^{(1)}],$$
$$= B^{(1)} + k^{(1)} g^{(1)} g^{(1)T}.$$

If $\bar{B}^{(1)}$ and $g^{(1)}$ are partitioned as

$$g^{(1)} = \left[\begin{array}{c} g_1 \\ \hline \omega \end{array}\right]$$

and

$$\bar{B}^{(1)} = \left[\begin{array}{c|c} b_{11} + g_1{}^2 k^{(1)} & v^T + \omega^T g_1 k^{(1)} \\ \hline v + g_1 k^{(1)} \omega & H + k^{(1)} \omega\omega^T \end{array}\right],$$

then

$$\bar{M}_1{}^{-1}(B^{(1)} + k^{(1)} g^{(1)} g^{(1)T})(\bar{M}_1{}^{-1})^T =$$

$$\left[\begin{array}{c|c} b_{11} + k^{(1)} g_1{}^2 & 0 \\ \hline 0 & H + k^{(1)}\omega\omega^T - \dfrac{1}{(b_{11} + k^{(1)} g_1{}^2)}[v + k^{(1)} g_1 \omega][v + k^{(1)} g_1 \omega]^T \end{array}\right],$$

where $\bar{M}_1$ is the multiplier corresponding to the modified triangular factors. Consider the $(s-1) \times (s-1)$ submatrix

$$H + k^{(1)}\omega\omega^T - \frac{1}{(b_{11} + k^{(1)} g_1{}^2)}[v + k^{(1)} g_1 \omega][v + k^{(1)} g_1 \omega]^T$$

$$= H - \left(\frac{1}{b_{11}}\right) vv^T + \left(\frac{1}{b_{11}}\right) vv^T - \frac{1}{(b_{11} + k^{(1)} g_1{}^2)}$$
$$\cdot [v + k^{(1)} g_1 \omega][v + k^{(1)} g_1 \omega]^T + k^{(1)}\omega\omega^T$$

which, after some manipulation gives

$$= H - \left(\frac{1}{b_{11}}\right) vv^T + \frac{k^{(1)}}{b_{11}(b_{11} + k^{(1)} g_1{}^2)}$$
$$\cdot [g_1{}^2 vv^T + b_{11}^2 \omega\omega^T - b_{11} g_1 \omega v^T - b_{11} g_1 v \omega^T],$$

$$= B^{(2)} + \frac{k^{(1)}}{b_{11}(b_{11} + k^{(1)} g_1{}^2)}(b_{11}\omega - g_1 v)(b_{11}\omega - g_1 v)^T.$$

If $g^{(2)}$ is defined as

$$g^{(2)} = (b_{11}\omega - g_1 v) = b_{11}(\omega - l^{(1)} g_1),$$

and

$$k^{(2)} = \frac{k^{(1)}}{b_{11}(b_{11} + k^{(1)}g_1^2)}, \tag{22}$$

then

$$\bar{M}_1^{-1}\bar{B}^{(1)}(\bar{M}_1^{-1})^T = \left[\begin{array}{c:c} b_{11} + k^{(1)}g_1^2 & 0 \\ \hdashline 0 & B^{(2)} + k^{(2)}g^{(2)}g^{(2)T} \end{array}\right].$$

Since the first column of $B^{(2)}$ is known from the old lower triangular factors L, it is possible to calculate the new triangular factors and the quantities $k^{(2)}$, $g^{(2)}$. The elements 2 to s of the first column of the modified triangular factor are given by

$$\left(\frac{1}{\bar{b}_{11}^{(1)}}\right)\begin{bmatrix} \bar{b}_{21}^{(1)} \\ \vdots \\ \bar{b}_{s,1}^{(1)} \end{bmatrix} = \left(\frac{1}{\bar{b}_{11}^{(1)}}\right)[b_{11}l^{(1)} + g_1k^{(1)}\omega].$$

Since $\bar{B}$ is positive definite it can have no zero diagonal elements and the denominator in (22) is nonzero. The algorithm is repeated on the $(s-1) \times (s-1)$ submatrix $\bar{B}^{(2)}$ and the process continued, the jth column of the modified triangular factor being obtained from the first column of $\bar{B}^{(j)}$.

7.2. Method B: Elementary Hermitian Matrices

Consider the matrix

$$A^T = [L \vdots 0]P^T,$$

where A^T is an $s \times (t+1)$ matrix, $t \geqslant s$, L is an $s \times s$ lower triangular matrix, and P is a $(t+1) \times (t+1)$ orthogonal matrix. Let $\bar{A}^T$ denote the matrix A^T with the kth column a deleted and $\bar{P}^T$ the first s rows of the matrix P^T with the kth column p deleted.

Then

$$\bar{A}^T = L\bar{P}^T.$$

Let $p^{(1)}$ be an $s \times 1$ vector such that

$$p = \begin{bmatrix} p^{(1)} \\ \hdashline \hat{p}^{(1)} \end{bmatrix}.$$

Define $p^{(j)}$ as the vector consisting of all the elements of $p^{(j-1)}$ except the first. Then

$$Lp^{(1)} = a,$$

and

$$\bar{P}^T\bar{P} = I - pp^T,$$

hence

$$\bar{A}^T\bar{A} = L(I - p^{(1)}p^{(1)T})L^T.$$

This can be written in the form

$$\bar{A}^T\bar{A} = LB^{(1)}B^{(1)T}L^T,$$

where

$$B^{(j)} = I - \sigma^{(j)}p^{(j)}p^{(j)T}, \qquad j = 1, \ldots, s,$$

$$\sigma^{(1)} = 1/[1 \mp (\hat{p}^{(1)T}\hat{p}^{(1)})^{1/2}].$$

In general $\sigma^{(1)}$ is chosen as

$$\sigma^{(1)} = 1/(1 + |\hat{p}^{(1)T}\hat{p}^{(1)}|^{1/2}).$$

$\sigma^{(j)}$, $j = 2, \ldots, s$ are given later in this section. Note that if $\hat{p}^{(1)T}\hat{p}^{(1)} = 1$, $p^{(1)} = 0$ hence $a = 0$ and no reduction of $B^{(1)}$ is necessary.

The matrix $B^{(1)}$ can be reduced to a lower triangular matrix $\tilde{L}$ using a sequence of Householder orthogonal matrices W_r, where the W_r are of the form

$$W_r = I - \alpha^{(r)}\omega^{(r)}\omega^{(r)T},$$

$$\omega^{(r)T}\omega^{(r)} = 2/\alpha^{(r)},$$

hence

$$\tilde{L} = B^{(1)}W_1 \cdots W_s.$$

If

$$\gamma^{(1)} = 1 - \sigma^{(1)}p_1{}^{(1)2},$$

then

$$\alpha^{(1)} = [d^{(1)2} \mp \gamma^{(1)}d^{(1)}]^{-1},$$

where

$$d^{(1)2} = \sigma^{(1)2} p_1^{(1)2} \sum_{j=2}^{s} p_j^{(1)2} + \gamma^{(1)2}.$$

Hence the elements of $\omega^{(1)}$ are given by

$$\omega_1^{(1)} = \gamma^{(1)} \mp d^{(1)},$$

and

$$\omega_j^{(1)} = -\sigma^{(1)} p_1^{(1)} p_j^{(1)}, \quad \text{for} \quad j = 2, \ldots, s.$$

Hence $B^{(1)} W_1$ is of the form

$$\left[\begin{array}{c|c} d^{(1)} & 0 \\ \hline \beta^{(1)} p^{(2)} & B^{(2)} \end{array}\right],$$

where

$$\sigma^{(2)} = \sigma^{(1)}(1 + p_1^{(1)} \alpha^{(1)} \sigma^{(1)} (p_1^{(1)} + p^{(1)T} \omega^{(1)})),$$

and

$$\beta^{(1)} = -\sigma^{(1)} p_1^{(1)} + \alpha^{(1)} \sigma^{(1)} p_1^{(1)} \omega_1^{(1)} + \omega_1^{(1)} \alpha^{(1)} \sigma^{(1)} p^{(1)T} \omega^{(1)}.$$

Postmultiplication by W_j, $j = 2, \ldots, s$ leaves the first row and column of $B^{(1)} W_1$ unaltered. Extending therefore the definition of $\beta^{(1)}$ and $d^{(1)}$ to $\beta^{(j)}$ and $d^{(j)}$, respectively, the jth column of $\tilde{L}$ can be written

$$\tilde{l}_j = \left[\begin{array}{c} 0 \\ \hline d^{(j)} \\ \hline \beta^{(j)} p^{(j+1)} \end{array}\right].$$

If $\mathscr{L}$ is defined to be the lower triangular matrix such that

$$\bar{A}^T = [\mathscr{L} \,\vdots\, 0] \mathscr{P}^T,$$

where $\mathscr{P}$ is an orthogonal matrix, then

$$\mathscr{L} = L\tilde{L}.$$

In general L and $\tilde{L}$ are dense lower triangular matrices and straightforward multiplication would take $s^3/6 + O(s^2)$ operations; however $\mathscr{L}$ can be determined in only $O(s^2)$ operations in the following fashion.

Let l_j denote the jth column of $\mathscr{L}$ and $l^{(j)}$ the $(s - j) \times 1$ vector of the last $(s - j)$ elements of l_j. Then

$$l_j = L\bar{l}_j.$$

Partitioning L in the form

$$L = \begin{bmatrix} l_{1,1} & 0 & & & \\ & l_{2,2} & & 0 & 0 \\ & & & l_{j,j} & \\ l^{(1)} & l^{(2)} & \cdots & l^{(j)} & \\ & & & & L^{(j+1)} \end{bmatrix}$$

gives

$$\begin{aligned} l_{j,j} &= d^{(j)}l_{j,j}, \\ l^{(j)} &= d^{(j)}l^{(j)} + \beta^{(j)}L^{(j+1)}p^{(j+1)}. \end{aligned} \tag{23}$$

Now recall that

$$a = Lp^{(1)}.$$

If $a^{(j)}$ denotes the $(s - j + 1) \times 1$ vector of the last $(s - j + 1)$ elements of a,

$$a^{(2)} = p_1{}^{(1)}l^{(1)} + L^{(2)}p^{(2)},$$

and substituting in (23) with $j = 1$ gives

$$l^{(1)} = d^{(1)}l^{(1)} + \beta^{(1)}(a^{(2)} - p_1{}^{(1)}l^{(1)}),$$

or

$$l^{(1)} = \beta^{(1)}a^{(2)} + (d^{(1)} - \beta^{(1)}p_1{}^{(1)})l^{(1)}.$$

$L^{(j+1)}p^{(j+1)}$ can be determined from the relationship

$$L^{(j)}p^{(j)} = \begin{bmatrix} l_{j,j}p_j{}^{(j)} \\ l^{(j)}p_j{}^{(j)} + L^{(j+1)}p^{(j+1)} \end{bmatrix}.$$

$L^{(2)}p^{(2)}$ and $L^{(s)}p^{(s)}$ are known and since all the $\beta^{(j)}$ and $d^{(j)}$ are available the l_j can be determined recurring backward or forward.

8. MODIFICATION OF THE TRIANGULAR FACTORS OF A^TA WHEN A ROW IS ADDED TO A

8.1. *Method A: Elementary Nonunitary Matrices*

In this case the following relation holds

$$\bar{A}^T\bar{A} = A^TA + aa^T.$$

Exactly the same procedure as method A in Sec. 7 can be used but with $k^{(1)} = 1$ instead of -1.

8.2. *Method B: Elementary Hermitian Matrices*

Let

$$\bar{A}^T\bar{A} = A^TA + aa^T \quad \text{where } A \text{ is a } t \times s \text{ matrix}, \quad t \geqslant s,$$

with

$$\bar{A}^T = [a \,\vdots\, A^T] = [\bar{L} \,\vdots\, 0]\bar{P}^T,$$

where

$$\bar{L} = [a \,\vdots\, L],$$

and

$$\bar{P}^T = \left[\begin{array}{c|c} 1 & 0 \\ \hline 0 & P^T \end{array}\right].$$

Note that $\bar{P}$ is still an orthogonal matrix. The matrix $\bar{L}$ is a lower Hessenberg matrix and can be reduced to lower triangular form by the methods given in Sec. 5.

An alternative procedure is possible if $\bar{A}$ is defined as

$$\bar{A}^T = [A^T \,\vdots\, a].$$

Then

$$\bar{A}^T = [L \,\vdots\, 0]\bar{P}^T,$$

where

$$\bar{P}^T = \left[\begin{array}{c|c} P^T & p \\ \hline 0 & p_{t+1} \end{array}\right].$$

Let $p^{(1)}$ be the vector consisting of the first s elements of p. Then

$$Lp^{(1)} = a.$$

Now

$$\bar{p}^T\bar{p} = \left[\begin{array}{c|c} I + pp^T & p_{t+1}p \\ \hline p_{t+1}p^T & p^Tp + p_{t+1}^2 \end{array}\right],$$

so that

$$\begin{aligned} \bar{A}^T\bar{A} &= L(I + p^{(1)}p^{(1)T})L^T, \\ &= L(I + \sigma^{(1)}p^{(1)}p^{(1)T})(I + \sigma^{(1)}p^{(1)}p^{(1)T})L^T, \end{aligned}$$

where

$$\sigma^{(1)} = \frac{(1 + p^{(1)T}p^{(1)})^{1/2} - 1}{p^{(1)T}p^{(1)}}.$$

The matrix $(I + \sigma^{(1)}p^{(1)T}p^{(1)})$ can be reduced to lower triangular form in exactly the same fashion as in Sec. 7.2.

9. MODIFICATIONS TO THE BASIC ITERATION IN THE CASE OF SIMPLE CONSTRAINTS

In many problems solved by linear programming techniques simple constraints of the form $\pm x_j \geqslant b_j$ are imposed upon the variables. The particular structure of the coefficient matrix can be utilized within the basic iteration to obtain a saving in arithmetic operations and storage requirements.

In the following discussion the original problem is considered to possess r general constraints and n simple constraints of the form $x_j \geqslant b_j$. If r' general constraints are active at the beginning of the ith iteration the variables can be ordered such that the matrix of active constraints has the form

$$A^{(i)T} = \left[\begin{array}{c|c} A_1^{(i)T} & A_2^{(i)T} \\ \hline I_{n-r'} & 0 \end{array}\right],$$

where $A_1^{(i)T}$ is an $r' \times (n - r')$ matrix, $A_2^{(i)}$ is an $r' \times r'$ matrix and $I_{n-r'}$ the $(n - r') \times (n - r')$ identity matrix. Similarly $\bar{A}^{(i)T}$ has the form

$$\bar{A}^{(i)T} = \begin{bmatrix} \bar{A}_1^{(i)T} & \bar{A}_2^{(i)T} \\ 0 & I_{r'} \end{bmatrix},$$

where $\bar{A}_1^{(i)T}$, $\bar{A}_2^{(i)T}$ have shapes: $(r - r') \times (n - r')$, $(r - r') \times r'$ respectively. The modifications to each step of the basic iteration are as follows.

Step 1

The Lagrange multipliers are solutions to the equations

$$A^{(i)}u = c.$$

If the column vectors u and c are partitioned accordingly, the equations become

$$\begin{bmatrix} A_1^{(i)} & I_{n-r'} \\ A_2^{(i)} & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix},$$

giving

$$A_2^{(i)}u_1 = c_2, \tag{24}$$

$$u_2 = c_1 - A_1^{(i)}u_1. \tag{25}$$

Using the strategy outlined in the basic iteration, Eq. (24) is replaced by

$$A_2^{(i)T}A_2^{(i)}u_1 = A_2^{(i)T}c_2;$$

u_1 is found, and substitution in Eq. (25) gives u_2. Only the triangular factors of $A_2^{(i)T}A_2^{(i)}$ need be stored during any one iteration, resulting in a considerable saving in storage and computational effort.

Step 2

The method used to determine the qth row of $A^{(i)-1}$ is dependent upon the nature of the constraint about to be deleted. The equations for p are given by

$$\begin{bmatrix} A_1^{(i)T} & A_2^{(i)T} \\ I_{n-r'} & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = e_q.$$

(*a*) *General Constraint Deleted at Step 1*. In this case $q \leqslant r'$, $p_1 = 0$ and $A_2^{(i)T}p_2 = e_q$ where e_q is now the qth column of $I_{r'}$. Then p_2 is found from

$$L_2^{(i)}L_2^{(i)T}y = e_q,$$

$$p_2 = A_2^{(i)}y.$$

(*b*) *Simple Constraint Deleted at Step 1.* Now $q > r'$ and the equations for p reduce to

$$A_1^{(i)T}e_{q-r'} + A_2^{(i)T}p_2 = 0,$$

i.e.,

$$A_2^{(i)T}p_2 = -a_{q-r'},$$

where $a_{q-r'}$ is the $(q - r')$th row of $A_1^{(i)}$, $e_{q-r'}$ is the $(q - r')$th column of $I_{n-r'}$. p_2 can be calculated from

$$L_2^{(i)}L_2^{(i)T}y = -a_{q-r'},$$

$$p_2 = A_2^{(i)}y.$$

Step 3

The calculation of v can be significantly simplified. In general v is given by

$$v = \begin{bmatrix} \bar{A}_1^{(i)T} & \bar{A}_2^{(i)T} \\ 0 & I_{r'} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \end{bmatrix}.$$

(*a*) *Simple Constraint Deleted at Step 1*

$$v = \begin{bmatrix} \bar{A}_1^{(i)T} & \bar{A}_2^{(i)T} \\ 0 & I_{r'} \end{bmatrix} \begin{bmatrix} e_{q-r'} \\ p_2 \end{bmatrix},$$

giving

$$v = \begin{bmatrix} \bar{a}_{q-r'} + \bar{A}_2^{(i)T}p_2 \\ p_2 \end{bmatrix},$$

where $\bar{a}_{q-r'}$ is the $(q - r')$th row of $\bar{A}_1^{(i)}$.

(*b*) *General Constraint Deleted at Step 1.* In this case v becomes

$$v = \begin{bmatrix} \bar{A}_2^{(i)T}p_2 \\ p_2 \end{bmatrix}.$$

Step 4

Some economy can be achieved in the calculation of $x^{(i+1)}$ (and sometimes $d_n^{(i+1)}$) due to the sparsity of the vector p.

Step 5

The lower triangular factors of the recurred matrices must be modified. As in the basic iteration the triangular factors are modified twice, after the first constraint has been deleted and after the new constraint has been added. There are several possibilities regarding the nature of the incoming/outgoing constraints.

(*a*) *Deletion of a Simple Constraint.* When a simple constraint disappears from the basis a row of $I_{n-r'}$ is deleted and the variables reordered such that the corresponding row of $A_1^{(i)}$ is placed amongst those of $A_2^{(i)}$. The algorithm described in Sec. 8 is used to modify the triangular factors of $A_2^{(i)}$.

(*b*) *Deletion of a General Constraint.* A column is deleted from $A_1^{(i)}$ and $A_2^{(i)}$ and the corresponding triangular factors of $A_2^{(i)T}A_2^{(i)}$ are updated as in Sec. 5.

(*c*) *Addition of a Simple Constraint.* If the simple constraint corresponds to the variable x_j then the variables must be reordered such that the coefficients of x_j which make up a column of $A_2^{(i)T}$ are added to $A_1^{(i)T}$. Since $A_2^{(i)T}$ has been column-deleted, the lower triangular factors are modified as in Sec. 7. The new $A_1^{(i)T}$ is formed so that the identity matrix is maintained in the bottom left hand corner of the coefficient matrix, in which case formulas (24) and (25) hold for the next iteration.

(*d*) *Addition of a General Constraint.* In this case $A_1^{(i)T}$, $A_2^{(i)T}$ are both row-augmented. The triangular factors of $A_2^{(i)T}A_2^{(i)}$ are modified using the procedure outlined in Sec. 6. Step 5 completes the modified basic iteration.

It is not the intention of this paper to consider further problems posed by $A^{(i)}$ being sparse or possessing special structure. Certainly the algorithm can be adapted to consider the type of coefficient matrix that arises in practical problems. All necessary formulas for the modification of the lower triangular factors which could arise from these considerations are given in Secs. 5–8.

10. NONSIMPLEX STEPS

The algorithm described in Sec. 3, does not depend upon n', the number of active constraints, being equal to n, the number of variables. This implies that the initial approximation to the solution, x_0, need not be a feasible vertex but any feasible point. At each subsequent iteration a new constraint enters the basis while the value of the objective function is still being decreased. During any period with $n' < n$ a considerable saving in computational effort and storage requirements is obtained and since progress can be made across the interior of the simplex, the number of iterations necessary to find the solution will be reduced.

Having reached a point with r (say) constraints active it is also possible to move off more than one constraint simultaneously. In this case the computational effort increases with the number of constraints being deleted until approximately $r/3$ constraints are discarded simultaneously when it is more advantageous to drop all the constraints and build up the active basis afresh. The work then decreases until it is possible to move off all the constraints with no work at all. This strategy is recommended when a large number of constraints are likely to be redundant. In particular, if $-c$ lies interior to the simplex, a step in this direction can be made with no constraints in the basis. At each subsequent iteration a new constraint becomes active but a currently active constraint for which the Lagrange multiplier is negative can be deleted, giving an expected small number of iterations with a full basis.

When $n' < n$ the direction of search, p, is no longer unique, but need only satisfy the relations

$$\mathscr{A}^{(i)T}p = 0, \tag{26}$$

$$c^T p < 0, \tag{27}$$

and

$$a_q{}^T p > 0, \tag{28}$$

where $A^{(i)T}$ is the matrix of active constraints at the ith iteration, and $\mathscr{A}^{(i)T}$ the matrix $A^{(i)T}$ with the qth row $a_q{}^T$ deleted. If a constraint is not deleted from the basis at the ith iteration, only conditions (26) and (27) need be satisfied, where $\mathscr{A}^{(i)}$ is now equal to $A^{(i)}$.

The conditions (26), (27), and (28) ensure that the new direction of search remains feasible and decreases the objective function. Such a p

will be called a feasible descent direction. In step 2 of the basic iteration when $n' = n$, p was chosen as the qth column of $(A^{(i)T})^{-1}$ and shown to satisfy Eqs. (27) and (28). For $n' < n$, providing the multiplier β can be chosen to satisfy Eqs. (27) and (28), any p of the form

$$p = \beta\{\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}t - t\} \tag{29}$$

will suffice, where t is an arbitrary vector linearly independent of the columns of $\mathscr{A}^{(i)}$. In the case where $n' = n$ the vector p is unique apart from the arbitrary multiplier β.

Judicious choice of the vector t in Eq. (29) can significantly reduce the amount of computation required. Three possible choices are c, a_q, or $\mathscr{p}$, the latter being the p of the previous iteration. Let the respective p's from these choices be p_c, p_a and p. The remainder of this section is devoted to examining under what circumstances these p's are feasible descent directions and giving details concerning their computation. The following lemma will prove useful.

LEMMA 1. *Let γ be the (q, q)th element of $(A^{(i)T}A^{(i)})^{-1}$. Then*

$$\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}a_q - a_q = -\frac{1}{\gamma}v,$$

where v is the qth row of the matrix $(A^{(i)T}A^{(i)})^{-1}A^{(i)T}$.

Proof. Let

$$A^{+} = (A^{(i)T}A^{(i)})^{-1}A^{(i)T}$$

and reorder the rows of A^{+} such that v^T appears as the n'th row. Then by definition

$$\begin{bmatrix} \mathscr{A}^{(i)T}\mathscr{A}^{(i)} & \mathscr{A}^{(i)T}a_q \\ a_q{}^T\mathscr{A}^{(i)} & a_q{}^T a_q \end{bmatrix} \begin{bmatrix} H \\ v^T \end{bmatrix} = \begin{bmatrix} \mathscr{A}^{(i)T} \\ a_q{}^T \end{bmatrix},$$

where H is the matrix of remaining $n' - 1$ rows of A^{+}. If H is eliminated from this system we obtain

$$\{a_q{}^T\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}a_q - a_q{}^Ta_q\}v = \mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}a_q - a_q.$$

We must now obtain an expression for the scalar multiple of v. Consider the set of equations

$$\begin{bmatrix} \mathscr{A}^{(i)T}\mathscr{A}^{(i)} & \mathscr{A}^{(i)T}a_q \\ a_q{}^T\mathscr{A}^{(i)} & a_q{}^Ta_q \end{bmatrix} \begin{bmatrix} z^{(1)} \\ \gamma \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}.$$

The vector $[z^{(1)T} \vdots \gamma]$ corresponds to the qth row of $(A^{(i)T}A^{(i)})^{-1}$, and if $z^{(1)}$ is eliminated from this set of equations we obtain

$$\gamma = \{a_q{}^Ta_q - a_q{}^T\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}a_q\}^{-1}.$$

Clearly, since $\gamma > 0$ is the (q, q)th element of $(A^{(i)T}A^{(i)})^{-1}$, we have

$$-\frac{v}{\gamma} = \mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}a_q - a_q,$$

and the Lemma is proved. ■

THEOREM 1. *The direction of search*

$$p_c = \beta_c\{\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}c - c\}$$

is a feasible descent direction for all $\beta_c > 0$.

Proof. We have by definition

$$c^Tp_c = \beta_c\{c^T\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}c - c^Tc\}.$$

We first prove that Eq. (27) is valid. Define the matrix

$$\tilde{A}^T = \begin{bmatrix} \mathscr{A}^{(i)T} \\ c^T \end{bmatrix};$$

then $-\beta_c(c^Tp_c)^{-1}$ is the (n', n') element of $(\tilde{A}^T\tilde{A})^{-1}$. Consequently $c^Tp_c < 0$ if $\beta_c > 0$ and relation (27) is satisfied.

To prove Eq. (28) we form

$$a_q{}^Tp_c = \beta_c\{a_q{}^T\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}c - a_q{}^Tc\}.$$

If the Lagrange multipliers calculated during the ith iteration are reordered in the form

$$\begin{bmatrix} \hat{u} \\ u_q \end{bmatrix},$$

where $\hat{u}$ are the multipliers with u_q excluded, and their partitioned equations written explicitly, then

$$\mathscr{A}^{(i)T}\mathscr{A}^{(i)}\hat{u} + \mathscr{A}^{(i)T}a_q u_q = \mathscr{A}^{(i)T}c, \tag{30}$$

$$a_q{}^T\mathscr{A}^{(i)}\hat{u} + a_q{}^T a_q u_q = a_q{}^T c. \tag{31}$$

Eliminating $\hat{u}$ from Eqs. (30) and (31) we have

$$\begin{aligned} &a_q{}^T\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}c - a_q{}^T c \\ &= u_q\{a_q{}^T\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}a_q - a_q{}^T a_q\}, \end{aligned}$$

giving

$$a_q{}^T p_c = \beta_c u_q\{a_q{}^T\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}a_q - a_q{}^T a_q\}.$$

Using Lemma 1 this equation can be written as

$$a_q{}^T p_c = -\frac{u_q\beta_c}{\gamma},$$

where γ is the (q, q)th element of $(A^{(i)T}A^{(i)})^{-1}$. If $u_q < 0$ and $\beta > 0$ then $a_q{}^T p_c > 0$ and p_c is a feasible descent direction. ■

THEOREM 2. *The direction of search*

$$p_a = \beta_a\{\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}a_q - a_q\}$$

is a feasible descent direction for all $\beta_a < 0$.

Proof. We first prove relation (27) by calculating

$$c^T p_a = \beta_a\{c^T\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}a_q - c^T a_q\},$$

and we obtain

$$c^T p_a = -\frac{\beta_a u_q}{\gamma},$$

giving $c^T p_a < 0$ if $u_q < 0$ and $\beta_a < 0$.

In a similar fashion

$$\begin{aligned} a_q{}^T p_a &= \beta_a\{a_q{}^T\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}a_q - a_q{}^T a_q\}, \\ &= -\beta_a/\gamma, \end{aligned}$$

with $a_q^T p_a > 0$ for $\beta_a < 0$. Clearly p_a is a feasible descent direction. ■

When less than n constraints are in the basis and the third choice of $t = \mathscr{p}$ is made in Eq. (29) it has not been possible to show that a β exists which will satisfy both Eq. (27) and Eq. (28). However, when $n' = n$, p must be a scalar multiple of p_c. Also if $n' \neq n$ and we do not delete a constraint then β can be chosen so that p is a descent direction.

Evaluation of p_c

We have

$$p_c = \beta_c\{\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}c - c\},$$

which, after using Eq. (30), becomes

$$p_c = \beta_c\{\mathscr{A}^{(i)}\hat{u} + u_q\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)}a_q^T - c\}.$$

Now

$$A^{(i)}u - c = \mathscr{A}^{(i)}\hat{u} + u_q a_q - c,$$

and the equation for p_c then reduces to

$$p_c = \beta_c\{A^{(i)}u - c + u_q(\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}a_q - a_q)\}.$$

We now use Lemma 1 to obtain

$$p_c = \beta_c\left\{A^{(i)}u - c - \frac{u_q}{\gamma}A^{(i)}y\right\},$$

where

$$L^{(i)}L^{(i)T}y = e_q, \tag{32}$$

that is, y is that vector defined in step 2 of the basic iteration. If β_c is chosen equal to γ then

$$p_c = \gamma(A^{(i)}u - c) - u_q A^{(i)}y.$$

If $n' = n$, the residual $A^{(i)}u - c$ is zero and

$$p_c = -u_q A^{(i)}y.$$

Evaluation of p_a

Since p_a is defined as

$$p_a = \beta_a\{\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}a_q - a_q\},$$

we see that Lemma 1 gives

$$p_a = -\frac{\beta_a}{\gamma}v,$$

$$= -\frac{\beta_a}{\gamma}A^{(i)}y.$$

If we put

$$\beta_a = \gamma u_q \qquad (\beta_a < 0),$$

then

$$p_a = - u_q A^{(i)}y,$$

and

$$p_c = p_a + \gamma(A^{(i)}u - c).$$

A natural question that arises is whether one of these directions is always better than the other. The matrix $I - A^{(i)}(A^{(i)T}A^{(i)})^{-1}A^{(i)T}$ is positive semidefinite; therefore

$$c^T(I - A^{(i)}(A^{(i)T}A^{(i)})^{-1}A^{(i)T})c \geqslant 0,$$

hence

$$c^T(c - A^{(i)}u) \geqslant 0.$$

Now

$$c^Tp_c = c^Tp_a + \gamma c^T(A^{(i)}u - c),$$

and since γ is positive

$$c^Tp_c \leqslant c^Tp_a < 0.$$

The best direction, say p_b, satisfies the inequality

$$\frac{c^Tp_b}{\|p_b\|} \leqslant \frac{c^Tz}{\|z\|}, \qquad \forall z \in E_n,$$

where

$$||z|| = (z^T z)^{1/2}.$$

It can be shown from the relation between p_a and p_c that

$$||p_c||^2 = ||p_a||^2 + \gamma^2 ||A^{(i)}u - c||$$

and consequently no a priori comment can be made as to which of the two directions is best.

Evaluation of p

We have defined

$$p = \beta\{\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}\mathscr{A}^{(i)T}\mathscr{P} - \mathscr{P}\}.$$

However, $\mathscr{P}$ must be orthogonal to all but one of the constraints that make up $\mathscr{A}^{(i)T}$ (the constraint $a_{n'}^T$ added during the last iteration). Consequently we have

$$p = \beta\{\alpha\mathscr{A}^{(i)}(\mathscr{A}^{(i)T}\mathscr{A}^{(i)})^{-1}e_{n'-1} - \mathscr{P}\},$$

where

$$\alpha = a_{n'}^T\mathscr{P}.$$

If we recur the factorization

$$\mathscr{A}^{(i)T}\mathscr{A}^{(i)} = \mathscr{L}^{(i)}\mathscr{L}^{(i)T},$$

then

$$p = \beta\left\{\frac{\alpha}{\delta}\mathscr{A}^{(i)}z - \mathscr{P}\right\},$$

where δ is the $(n'-1)$, $(n'-1)$ element of $\mathscr{L}^{(i)}$ and z is obtained from the back substitution

$$\mathscr{L}^{(i)T}z = e_{n'-1}. \tag{33}$$

The reason for considering this choice of t at all is that Eq. (32) used in the evaluation of p_a and p_c simplifies to become Eq. (33) in this case.

11. STORAGE AND COMPUTATIONAL REQUIREMENTS

When the matrix of active constraints has an insignificant number of zero elements the form of the algorithm presented here requires $n^2/2 + O(n)$ storage locations in addition to the original data. The number of arithmetic operations per iteration varies according to

(i) the position of the constraint leaving the basis of constraints, and

(ii) the particular modification method used.

The number of multiplications for each method used to modify the lower triangular form when a row is deleted from A^T is shown in Table 1, where it is assumed that the deletion of a particular constraint between 1 and n is equally likely.

TABLE 1

AVERAGE NUMBER OF MULTIPLICATIONS REQUIRED FOR THE MODIFICATION OF THE TRIANGULAR FACTORS WHEN A ROW IS CHANGED IN A^T

	Constraint Leaving the Basis			Constraint Entering the Basis
Method	A	B	C	
Number of multiplications	$\frac{5}{6}n^2 + O(n)$	$\frac{2}{3}n^2 + O(n)$	min: $\frac{1}{2}n^2 + O(n)$ max: $\frac{2}{3}n^2 + O(n)$	$\frac{3}{2}n^2 + O(n)$

If method B is used when a constraint leaves the basis and p is obtained using the method requiring the least number of multiplications, the average total amount of work is given by

$$4\tfrac{1}{2}n^2 + n(m - n) + O(n)$$

(see Table 2).

The inclusion of non-Simplex steps reduces the amount of work and the additional storage requirements. If n' constraints are active at the ith iteration with $n' < n$, then $(n'/2)^2 + O(n')$ additional storage locations are required, and the work reduces to

$$1\tfrac{5}{6}(n')^2 + 2nn' + n(m - n') + O(n - n') \quad \text{multiplications.}$$

When average figures of $m = 3n$, $n' = n/2$ are assumed the average work becomes

$$3\tfrac{23}{24}n^2 + O(n) \quad \text{with} \quad n^2/8 + O(n) \quad \text{locations.}$$

TABLE 2

NUMBER OF MULTIPLICATIONS PER CYCLE FOR THE EVALUATION OF THE BASIC ITERATION WITH GENERAL CONSTRAINTS

Source	Number of Multiplications
Calculation of the Lagrange multipliers	$\frac{5}{6}n^2 + O(n)$
Calculation of p	$\frac{3}{2}n^2 + O(n)$
Calculation of v	$n(m - n)$
Modification of the lower triangular factors	$\frac{13}{6}n^2 + O(n)$

A direct comparison can be made with the explicit inverse version of the Simplex method when A^T in Eq. (P1) is given by

$$A^T = \begin{bmatrix} B \\ I \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} b \\ 0 \end{bmatrix},$$

B is an $r \times n$ matrix with $r < n$, and $m = n + r$. The constraints are equivalent to

$$Bx \geqslant b, \qquad x \geqslant 0.$$

In the standard Simplex method the variables x are augmented by r slack variables y such that

$$x' = \begin{bmatrix} x \\ y \end{bmatrix},$$

and the constraints become

$$B'x' = b',$$

$$x' \geqslant 0,$$

where

$$B' = [B \vdots D],$$

D being a diagonal matrix with elements ± 1. During any current iteration of the standard Simplex method r linearly independent columns are chosen from B' to form the "column basis" and the explicit inverse of the matrix formed by these columns is stored. As columns are interchanged in the column basis the explicit inverse is modified. The size of the matrix recurred during the process is constant, the amount of work required per iteration being (Hadley [7])

$$(n + r)r + r^2 + O(r) \quad \text{multiplications.}$$

If an average figure of $n = 3r$ is assumed the amount of work becomes

$$5r^2 + O(r) \quad \text{multiplications,} \tag{34}$$

with the amount of additional storage needed fixed at

$$r^2 + O(r) \quad \text{locations.} \tag{35}$$

The linear programming problem with matrix of constraints given by A^T can be dealt with as described in Sec. 9. In this case r' corresponds to the number of elements of D not in the column basis of the explicit inverse. During any iteration it may be necessary to change a column of the $r' \times r'$ submatrix of A^T corresponding to $A_2^{(i)}$ in Sec. 9. The computational requirements for each of the methods described in Secs. 7 and 8 are given in Table 3.

TABLE 3

NUMBER OF MULTIPLICATIONS REQUIRED FOR THE MODIFICATION OF THE TRIANGULAR FACTORS OF A^TA WHEN A COLUMN OF A^T IS ALTERED

Method	Modification due to Incoming Column		Modification due to Outgoing Column	
	A	B	A	B
Number of multiplications	$\frac{3}{2}(r')^2 + O(r')$	$2(r')^2 + O(r')$	$\frac{3}{2}(r')^2 + O(r')$	$2(r')^2 + O(r')$

If an expected figure of $n = 3r$ is taken and it is assumed that approximately half the simple constraints are active on average, then the total amount of work for any iteration is given by

$$2\tfrac{5}{6}r^2 + O(r). \tag{36}$$

The corresponding average storage per iteration is given by

$$r^2/8 + O(r). \tag{37}$$

A comparison of Eq. (34) with Eq. (36) and Eq. (35) with Eq. (37) demonstrates the saving over the explicit inverse form of the Simplex method.

If non-Simplex steps are taken further economy can be achieved, for example, the expected amount of storage reduces to

$$r^2/32 + O(r).$$

TABLE 4

EXPECTED NUMBER OF MULTIPLICATIONS FOR THE EXECUTION OF THE MODIFIED BASIC ITERATION

Source		Number of Multiplications
Calculation of Lagrange multipliers		$\frac{5}{6}(r')^2 + r'(n - r') + O(r')$
Calculation of p	simple constraint outgoing	$2(r')^2 + O(r')$
	general constraint outgoing	$\frac{3}{2}(r')^2 + O(r')$
Calculation of v	simple constraint outgoing	$r'(r - r') + O(r')$
	general constraint outgoing	$r'(r - r') + O(r')$
Modification of $A_2^{(i)T}A_2^{(i)}$:		
Simple constraint incoming + Simple constraint outgoing		$3(r')^2 + O(r')$
Simple constraint incoming + General constraint outgoing		$\frac{13}{6}(r')^2 + O(r')$
General constraint incoming + General constraint outgoing		$\frac{13}{6}(r')^2 + O(r')$
General constraint incoming + Simple constraint outgoing		$3(r')^2 + O(r')$

11. COMMENTS AND CONCLUSIONS

The importance of numerical stability in methods used for the solution of linear programs is not always appreciated. Although the schemes of Bartels and Golub [1, 2] have been well publicized, practitioners have seemingly preferred to sacrifice numerical stability for apparent advantages in storage and computational effort. However, the number of iterations needed is likely to decrease for a numerically stable algorithm, since it is possible for a numerically unstable method not to converge at all. An algorithm has been presented which is competitive in storage requirements and computational effort with the standard Simplex method.

This consideration has principally been with linear programming problems which possess no special form of constraints. Alternative versions of the algorithm more suitable to sparse and structured systems will be the subject of future publications.

Murray [9] has extended the method to indefinite quadratic programming. Although still regarded as an extension of the Simplex method it differs radically from the two most popular methods [**3**, **13**] which transform the problem into an artificial linear program. The formulation of the problem and its method of solution illustrates the natural link between linear and quadratic programming. Further investigations have been made extending the numerical techniques to the minimization of a nonlinear function without constraints [6] and with linear constraints [7].

Since the first appearance of this paper, further developmental work has taken place. An account of this work can be found in the publications of Dr. M. A. Saunders [10, 11].

The authors wish to thank Mr. E. L. Albasiny, Dr. M. A. Saunders, Dr. J. H. Wilkinson, Dr. D. W. Martin, and Miss Susan M. Picken for their careful reading of the manuscript and a number of helpful suggestions.

REFERENCES

1 R. H. Bartels, *A Numerical Investigation of the Simplex Method.* Technical Report, No. CS 104, Computer Science Department, Stanford University, Stanford, California (July 1968).

2 R. H. Bartels and G. H. Golub, The Simplex Method of linear programming using LU decomposition, *Comm. ACM* **12**(May 1969), 266–268.

3 E. M. L. Beale, On quadratic programming, *Naval Res. Log. Quart.* **6**(1959), 227–243.

4 J. M. Bennett, Triangular factors of modified matrices, *Numerische Math.* **7**(1965), 217–221.

5 G. B. Dantzig, *Linear Programming and Extensions*, Princeton University Press (1965).

6 P. E. Gill and W. Murray, Quasi-Newton methods for unconstrained optimization, *J. Inst. Math. Appl.* **9**(1972), 91–108.

7 P. E. Gill and W. Murray, *Two Methods for the Solution of Linearly Constrained and Unconstrained Optimization Problems*, NPL DNAC Report No. 25 (1972).

8 G. Hadley, *Linear Programming*, Addison-Wesley, Reading, Mass. (1962).

9 W. Murray, *An Algorithm for Finding a Local Minimum of an Indefinite Quadratic Program*, NPL DNAC Report No. 1 (1971).

10 M. A. Saunders, *Large-Scale Linear Programming Using the Cholesky Factorization*, Technical Report No. CS 252, Computer Science Department, Stanford University, Stanford, California (1972).

11 M. A. Saunders, *Product Form of the Cholesky Factorization for Large-Scale Linear Programming*, Technical Report No. CS 301, Computer Science Department, Stanford University, Stanford, California (1972).

12 J. H. Wilkinson, *The Algebraic Eigenvalue Problem*, Oxford University Press (1965).

13 P. Wolfe, The Simplex Method for Quadratic Programming, *Econometrica* **27**(1959), 382–398.

Received November, 1970