

SOLVING RANK ONE REVISED LINEAR SYSTEMS BY THE SCALED ABS METHOD

KEYVAN AMINI¹

(Received 14 June, 2003; revised 4 December, 2003)

Abstract

In mathematical programming, an important tool is the use of active set strategies to update the current solution of a linear system after a rank one change in the constraint matrix. We show how to update the general solution of a linear system obtained by use of the scaled ABS method when the matrix coefficient is subjected to a rank one change.

1. Introduction

The ABS methods, introduced by Abaffy, Broyden and Spedicato [1], are a general class of algorithms for solving linear and nonlinear algebraic systems and have been extensively applied to several types of linear systems. The basic algorithm works on a system having the form

$$Ax = b \quad (A \in R^{m \times n}, x \in R^n, b \in R^m). \quad (1.1)$$

The scaled ABS method is one of a class of methods (for solving a linear system of equations) that are generalisations of the basic ABS algorithm, with an extra parameter vector available at each iteration. It can be seen that the scaled ABS method is a realisation of a very general class of algorithms ([2, 8]).

On the other hand, in a number of problems, after solving a linear system and computing solutions of it, we need to find the solutions of a new system where the coefficient matrix is modified. (Examples of modifications are changing, deleting or adding rows or columns to the matrix.)

In this paper, we show how to update the general solution of a linear system after a rank one change in the coefficient matrix of the system. We also show how to compute

¹Department of Mathematical Sciences, Sharif University of Technology, Tehran, Iran; e-mail: keyvanamini1353@yahoo.com or Amini@hotmail.com.

© Australian Mathematical Society 2004, Serial-fee code 1446-1811/04

the general solution of a rank one perturbed system

$$(A + uv^T)x = b \quad (A \in R^{m,n}, x \in R^n, b \in R^m, m \leq n), \quad (1.2)$$

where $u \in R^m$, $v \in R^n$, if the general solution of (1.1), obtained by the application of a scaled ABS algorithm, exists.

On the other hand, with an appropriate choice of parameters, almost all important algorithms for solving linear systems can belong to the scaled ABS method. (For example (see [2]), the QR method, LU, Huang, Hestenes-Stiefel and Lanczos methods [6, 7], a number of conjugate gradient-type algorithms and also the Voyevodin class of methods [9]). The new methods can be used for computing a solution of a system after a rank one perturbation of the coefficient matrix.

Special instances of system (1.2) may appear in the following forms:

- changing the i th column or the i th row of the coefficient matrix to the zero vector (by choosing $(u = -a_i, v = e_i)$ or $(u = e_i, v^T = -a'_i)$ respectively);
- changing the i th column or the i th row of the coefficient matrix (by choosing $v = e_i$ or $u = e_i$ respectively),

where e_i is the i th unit vector with appropriate dimension and a_i, a'_i are the i th column and the i th row of matrix A respectively.

We know that solving the above system is often necessary and very important in mathematical programming problems, such as methods using the active set strategy, the simplex method or the dual simplex.

In Section 2, we briefly describe the ABS method and the scaled ABS algorithms to solve linear systems. In Section 3, we show how to update the solution after a rank one change and present the resulting algorithm. In Section 4, we demonstrate an application of the new algorithm for linear programming and an application of the new algorithm for a family of the secant methods for nonlinear programming.

2. The scaled ABS algorithm for solving a linear system

The scaled ABS algorithm for solving a linear system of equations is a generalisation of the basic ABS algorithm (see [2, 8]). A basic ABS algorithm starts with an initial vector $x_1 \in R^n$ (arbitrary) and a nonsingular matrix $H_1 \in R^{n \times n}$ (Spedicato's parameter). Given that x_i is a solution of the first $i - 1$ equations, the ABS algorithm computes the solution of the given equations by performing the following steps (see [2] or [3]):

- (1) Determine z_i (Broyden's parameter) such that $z_i^T H_i a_i \neq 0$ and set $p_i = H_i^T z_i$.
- (2) Update the solution by $x_{i+1} = x_i + \alpha_i p_i$, where the step size α_i is given by

$$\alpha_i = (b_i - a_i^T x_i) / a_i^T p_i.$$

(3) Update the Abaffian matrix H_i by

$$H_{i+1} = H_i - \frac{H_i a_i w_i^T H_i}{w_i^T H_i a_i},$$

where $w_i \in R^n$ (Abaffy's parameter) is arbitrary provided that $w_i^T H_i a_i \neq 0$.

NOTE. It can be shown that

(1) $s_i = H_i a_i = 0$ if and only if a_i is linearly dependent on the vectors a_1, \dots, a_{i-1} (see [2]). Indeed if $s_i = 0$ and $t_i = b_i - a_i^T x_i = 0$ then the i th equation is redundant; if $s_i = 0$ and $t_i \neq 0$ then the system of equations is incompatible and hence lacks a solution.

(2) If the system is compatible, the ABS algorithm computes a solution x_{m+1} of system (1.1) and a matrix H_{m+1} so that $\text{Null}(H_{m+1}^T) = \text{Null}(A)$. Hence the general solution of the system has the form $\bar{x} = x_{m+1} + H_{m+1}^T s$, $s \in R^n$.

The scaled ABS family of algorithms is the essential generalisation of the ABS algorithms obtained by introducing a new parameter, called the scaling vector, at each iteration. The scaled ABS class of algorithms is formally and computationally different, but equivalent, in the sense of generating the same set of iterates x_i , to some general procedures for a linear system and shows that a fundamental characterisation of the scaled ABS algorithm is that it contains all possible algorithms of a very general iteration that find the solution, from an arbitrary starting point, in a number of steps no greater than the number of equations.

Now to introduce the scaled ABS class, let us consider, instead of the original system (1.1), the following scaled system:

$$K^T A x = K^T b, \quad (2.1)$$

where $K = (k_1, \dots, k_m) \in R^{m,m}$ is an arbitrary nonsingular $m \times m$ matrix. Then the above system is equivalent to system (1.1), any solution of one being a solution of the other. Note that, by letting $k_i \in R^m$ be the i th column of K , we can write system (2.1) componentwise in the form

$$k_i^T A x = k_i^T b, \quad i = 1, \dots, m,$$

or, also by introducing the unscaled residual vector $r = r(x) \in R^m$,

$$r(x) = A x - b$$

in the form

$$k_i^T r(x) = 0, \quad i = 1, \dots, m.$$

So if we apply the basic ABS algorithm to the scaled system, it is immediately seen that the relevant recursions are obtained by simply replacing a_i and b_i respectively by $A^T k_i$ and $b^T k_i$. So we get a solution of the original system (1.1).

Notice moreover that the full matrix K does not appear explicitly, since at the i th step only its i th column is used. Therefore K need not be defined initially, it being sufficient to define k_i at the i th step as an arbitrary vector linearly independent from k_1, \dots, k_{i-1} . The vectors k_i are called scaling vectors.

Note that if x_1 is a solution for system (2.1) and p is a vector in the null space of $K^T A$, then $\bar{x} = x_1 + \alpha p$, for any scalar α , will also be a solution to (2.1). Thus if H is a matrix, the columns of which span the null space of $K^T A$, then $x = x_1 + \alpha H z$, for arbitrary scalar α and vector z , solves the system (2.1).

Let $K^i = (k_1, \dots, k_i)$ and assume that y_i , the current approximation in iteration i is a solution for the first $i - 1$ equations in (2.1), that is,

$$K^{iT} A y_i = K^{iT} b. \tag{2.2}$$

We choose the matrix $H_i \in R^{n \times n}$ so that the columns of H_i span the null space of $K^{iT} A$, that is, equivalent with $H_i A^T k_j = 0, 1 \leq j \leq i - 1$, then we let

$$y_{i+1} = y_i - \alpha_i H_i^T z_i, \tag{2.3}$$

where $z_i \in R^n, \alpha_i \in R$.

It is clear that y_{i+1} satisfies $K^{iT} A y_{i+1} = K^{iT} b$. We determine the scalar $\alpha_i \in R$ so that y_{i+1} satisfies the i th equation as well, that is,

$$K^{i+1T} A y_{i+1} = K^{i+1T} b. \tag{2.4}$$

Defining $r_j = A y_j - b$ and considering (2.3), (2.4) is equivalent to

$$k_j^T r_i = \alpha_i k_j^T A p_i, \quad j \leq i, \tag{2.5}$$

where $p_i = H_i^T z_i$ is the search direction defined at the i th iteration. In (2.4), for $j = i$ we have

$$k_i^T r_i = \alpha_i k_i^T A p_i. \tag{2.6}$$

(Note that if $k_i^T A p_i \neq 0$ then $\alpha_i = k_i^T r_i / k_i^T A p_i$.)

For $j \leq i - 1$, using the induction hypothesis (2.2), (2.5) becomes

$$\alpha_i k_j^T A p_i = 0, \quad j \leq i - 1,$$

or

$$(H_i A^T k_j)^T (\alpha_i z_i) = 0, \quad j \leq i - 1.$$

So the following property is needed:

$$H_i A^T k_j = 0, \quad j \leq i - 1. \quad (2.7)$$

Assume H_i to have this property. Let H_{i+1} be defined as $H_{i+1} = H_i + g_i f_i^T$, where $f_i, g_i \in R^n$ are defined so that they also satisfy (2.7). We must then have

$$(H_i + g_i f_i^T) A^T k_j = 1, \quad j \leq i - 1.$$

To achieve the above, it will suffice to define $g_i = -H_i A^T k_i$ and $f_i = H_i^T w_i$, where $w_i \in R^n$ satisfies

$$w_i^T H_i A^T k_i = 1. \quad (2.8)$$

Therefore we obtain

$$H_{i+1} = H_i - H_i A^T k_i w_i^T H_i. \quad (2.9)$$

The relations (2.3), (2.6), (2.8) and (2.9) define the scaled ABS algorithm. Below, we give an outline of this algorithm for solving a linear system.

THE SCALED ABS ALGORITHM.

(1) Given $x_1 \in R^n$, k_1 arbitrary nonzero vectors, and $H_1 \in R^{n \times n}$ an arbitrary nonsingular matrix, set $i = 1$.

(2) Compute the vector $s_i = H_i A^T k_i$. If $s_i \neq 0$ go to (3). If $s_i = 0$, compute $\tau_i = r_i^T k_i$. If $\tau_i = 0$, set $x_{i+1} = x_i$, $H_{i+1} = H_i$ and go to (6), the i th equation is redundant, otherwise stop, the system has no solution.

(3) Determine $z_i \in R^n$ such that $z_i^T H_i A^T k_i \neq 0$ and set $p_i = H_i^T z_i$.

(4) Update the estimate of the solution by $x_{i+1} = x_i - \alpha_i p_i$, where

$$\alpha_i = \frac{r_i^T k_i}{p_i^T A^T k_i}.$$

(5) Update the Abaffian matrix H_i by

$$H_{i+1} = H_i - H_i A^T k_i w_i^T H_i, \quad (2.10)$$

where $w_i \in R^n$ is arbitrary and satisfies $w_i^T H_i A^T k_i = 1$.

(6) If $r_i = 0$, stop, x_{i+1} solves the system. Otherwise give a vector $k_{i+1} \in R^n$ linearly independent from k_1, \dots, k_i . Increment i by one and go to (2).

We now list some properties of the scaled ABS algorithm (see [2, 8]).

- Let $H_1 \in R^{n \times n}$ be an arbitrary nonsingular matrix, a_1, \dots, a_m be linearly independent vectors in R^n ($m \leq n$) and H_i be a sequence of matrices generated by the update formula (2.10) in the scaled ABS algorithm. Then

- (a) For $i \leq j \leq m$, the vectors $H_i A^T k_j$ are nonzero and linearly independent.
- (b) For $i = 1, \dots, m$, the vectors $A^T k_1, \dots, A^T k_i$ are nonzero and linearly independent, and form a basis for $\text{Null}(H_{i+1})$.

• The residual vector in x_{i+1} is orthogonal to the first i columns of K , that is, $K^{iT} r_{i+1} = 0$ where $K^i = (k_1, \dots, k_i)$ and any vector y such that the residual $r(y)$ is orthogonal to the first i columns of K has the form

$$y = x_{i+1} + H_{i+1}^T s \quad (s \in R^n). \tag{2.11}$$

Moreover any such vector y can be obtained at the i th step of the scaled ABS as x_{i+1} by suitable choice of z_i .

So in particular, x_{i+1} solves the first i equations of system (1.1) and every solution of the first i equations of (1.1) can be written in the form (2.11). In particular, the general solution of (1.1) is given by $\bar{x} = x_{m+1} + H_{m+1}^T s, s \in R^n$.

• If $\text{rank}(A) = q < m$ and the system has a solution then the scaled ABS algorithm computes a solution in q steps (in exact arithmetic). In addition, all solutions of (1.2) have the form $\bar{x} = x_{q+1} + H_{q+1}^T s, s \in R^n$, where H_{q+1} is the Abaffian obtained by an application of the scaled ABS algorithm to the independent rows of matrix A .

3. Solving rank one perturbed linear systems

We now consider the following perturbed system:

$$(A + uv^T)x = b, \quad x \in R^n.$$

This system can be transformed into the following form:

$$\begin{bmatrix} I_m & u \end{bmatrix} \begin{bmatrix} A \\ v^T \end{bmatrix} x = b, \quad x \in R^n, \tag{3.1}$$

where $I_m \in R^{m \times m}$ is the identity matrix. Assume that the system is a determined or an undetermined system ($m \leq n$). The system (3.1) is equivalent to the following systems of $m + n + 1$ equations:

$$\begin{bmatrix} I_m & u \end{bmatrix} y = b, \quad y \in R^{m+1}, \tag{3.2a}$$

$$\begin{bmatrix} A \\ v^T \end{bmatrix} x = y, \quad x \in R^n. \tag{3.2b}$$

The general solution of (3.2a) is characterised by the following theorem.

THEOREM 3.1. *The general solution of (3.2a) has the form*

$$y = \begin{bmatrix} b \\ 0 \end{bmatrix} + \begin{bmatrix} -u \\ 1 \end{bmatrix} t, \quad t \in R. \tag{3.3}$$

PROOF. First observe that (3.3) satisfies

$$\begin{bmatrix} I_m & u \\ & t \end{bmatrix} y = \begin{bmatrix} b - ut \\ t \end{bmatrix} = b - ut + ut = b.$$

Conversely, let y be a solution of the system (3.2a) and let

$$y = \begin{bmatrix} \bar{y} \\ t \end{bmatrix}, \quad t \in R, \quad \bar{y} \in R^m.$$

Then we have $\bar{y} = b - ut$ so

$$y = \begin{bmatrix} b - ut \\ t \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} + \begin{bmatrix} -u \\ 1 \end{bmatrix} t.$$

Now, by substituting (3.3) into (3.2b) we obtain

$$\begin{bmatrix} A \\ v^T \end{bmatrix} x = \begin{bmatrix} b - ut \\ t \end{bmatrix}$$

or

$$Ax = b - ut, \tag{3.4a}$$

$$v^T x = t. \tag{3.4b}$$

Using the search directions of the original system (1.1) (obtained by the application of a scaled ABS algorithm), the solution of the system (3.4a) can be written as

$$x_{m+1}(t) = x_1(t) + \sum_{i=1}^m \alpha_i(t) p_i,$$

where the step size α_i is a linear function of t , that is,

$$\alpha_i(t) = \frac{k_i^T r_i(t)}{k_i^T A p_i} = k_i^T \left(\frac{b - ut - A x_i(t)}{k_i^T A p_i} \right)$$

and

$$x_i(t) = x_1(t) + \sum_{j=1}^{i-1} \alpha_j(t) p_j.$$

So, if we let x_1 be a linear function of t ($t \in R$), that is, $x_1(t) = x_1^2 + x_1^1 t$, $x_1^1, x_1^2 \in R$, then it is obvious that $\alpha_i(t)$ and $x_i(t)$ are linear functions of t , that is, for every $i = 1, \dots, m$, $\alpha_i(t) = \alpha_i^2 + \alpha_i^1 t$ and $x_i(t) = x_i^2 + x_i^1 t$. Accordingly, we have

$$\begin{aligned} x_{m+1}(t) &= x_1(t) + \sum_{i=1}^m \alpha_i(t) p_i = x_1^2 + x_1^1 t + \sum_{i=1}^m (\alpha_i^2 + \alpha_i^1 t) p_i \\ &= \left(x_1^2 + \sum_{i=1}^m \alpha_i^2 p_i \right) + \left(x_1^1 + \sum_{i=1}^m \alpha_i^1 p_i \right) t. \end{aligned}$$

Thus $x_{m+1}(t) = r_2 + r_1t$, where

$$\begin{cases} r_1 = x_1^1 + \sum_{i=1}^m \alpha_i^1 p_i = x_1^1 - \sum_{i=1}^m \frac{k_i^T(u + Ax_i^1)}{k_i^T A p_i} p_i, \\ r_2 = x_1^2 + \sum_{i=1}^m \alpha_i^2 p_i = x_1^2 + \sum_{i=1}^m \frac{k_i^T(b - Ax_i^2)}{k_i^T A p_i} p_i. \end{cases} \tag{3.5}$$

Therefore the general solution of (3.4a) has the following form:

$$\begin{aligned} \hat{x} &= x_{m+1}(t) + H_{m+1}^T z, \quad z \in R^n, t \in R, \\ \Rightarrow \hat{x} &= r_2 + r_1t + H_{m+1}^T z, \quad z \in R^n, t \in R. \end{aligned}$$

We thus have the following result.

THEOREM 3.2. *The general solution of the system $Ax = b - ut$ is given by*

$$\hat{x} = r_2 + r_1t + H_{m+1}^T z, \quad (z \in R^n, t \in R), \tag{3.6}$$

where r_1, r_2 are given by (3.5).

Now, by substituting (3.6) into (3.4b) we can obtain the general solution of the perturbed system (1.2). After substituting we obtain

$$v^T H_{m+1}^T z + v^T r_1t + v^T r_2 = t. \tag{3.7}$$

We consider the following two cases.

Case 1. $H_{m+1}v = 0$ implying that v^T is a linear combination of the rows of the matrix A .

In this case, (3.7) is written as

$$(1 - v^T r_1)t = v^T r_2. \tag{3.8}$$

There are three possibilities for (3.8) as we outline below.

(a) $(v^T r_1 \neq 1)$. In this situation, the general solution of (1.2) is given by

$$\hat{x} = r_2 + \left(\frac{v^T r_2}{1 - v^T r_1} \right) r_1 + H_{m+1}^T z, \quad z \in R^n.$$

(b) $(v^T r_1 = 1 \text{ and } v^T r_2 = 0)$. Then (3.7) becomes $v^T r_2 = (1 - v^T r_1)t = 0$. Since $v^T r_2 = 0$, then (3.7) is satisfied for any arbitrary value of t . Therefore in this case the general solution is given by

$$\hat{x} = r_2 + r_1t + H_{m+1}^T z, \quad z \in R^n, t \in R$$

or

$$\hat{x} = (H_{m+1}^T r_1) \begin{pmatrix} z \\ t \end{pmatrix} + r_2 = \hat{H}^T g + r_2, \quad g \in R^{n+1}.$$

(c) ($v^T r_1 = 1$ and $v^T r_2 \neq 0$). Then the system (1.2) has no solution.

Case 2. $H_{m+1}v \neq 0$.

In this case, we use the basic ABS method and its properties (respectively with the choice $k_{m+1} = e_{m+1}$ in the scaled ABS method). Set

$$\begin{aligned}
 p_{m+1} &= H_{m+1}^T z_{m+1}, & \text{where } z_{m+1}^T H_{m+1} v &\neq 0, \\
 x_{m+2}(t) &= x_{m+1}(t) + \alpha_{m+1} p_{m+1}, & \text{where } \alpha_{m+1} &= \frac{t - v^T x_{m+1}(t)}{v^T p_{m+1}}, \\
 H_{m+2} &= H_{m+1} + H_{m+1} v w_{m+1}^T H_{m+1}, & \text{where } w_{m+1}^T H_{m+1} v &\neq 0
 \end{aligned}$$

so the general solution of the total system (1.2) is

$$\begin{aligned}
 \hat{x} &= x_{m+2}(t) + H_{m+2}^T q \\
 &= r_1 t + r_2 - \frac{(v^T r_1 - 1)t + v^T r_2}{v^T p_{m+1}} p_{m+1} + H_{m+2}^T q \\
 &= \left(r_2 - \frac{v^T r_2}{v^T p_{m+1}} p_{m+1} \right) + \left[H_{m+2}^T r_1 + \frac{1 - v^T r_1}{v^T p_{m+1}} p_{m+1} \right] \begin{bmatrix} q \\ t \end{bmatrix}, \quad q \in R^n.
 \end{aligned}$$

Now, considering the above results, we propose the following algorithm for solving the perturbed system (1.2).

ALGORITHM 1.

(1) Let H_{m+1} be the Abaffian matrix and p_1, \dots, p_m be the search vectors obtained by applying the scaled ABS algorithm for (1.1). Let $x_1 = x_1^1 t + x_1^2$ be a linear function of t ($t \in R$).

(2) **For** $i := 1$ **until** m **do**

$$x_{i+1}^1 := x_i^1 - \frac{k_i^T (u + Ax_i^1)}{k_i^T A p_i} p_i, \quad x_{i+1}^2 := x_i^2 + \frac{k_i^T (b - Ax_i^2)}{k_i^T A p_i} p_i.$$

(3) **Compute** $s = H_{m+1}v$.

(4) **If** ($s \neq 0$) **then** compute $p = H_{m+1}^T z_{m+1}$, where $z_{m+1}^T H_{m+1} v \neq 0$ and then

$$\hat{x} = r_2 - \frac{v^T r_2}{v^T p_{m+1}} p_{m+1}, \quad \hat{H} = \left[H_{m+2}^T r_1 + \frac{1 - v^T r_1}{v^T p_{m+1}} p_{m+1} \right]^T.$$

Else ($s = 0$).

If ($v^T r_1 \neq 1$) **then** compute

$$\hat{x} = \left(\frac{v^T r_2}{1 - v^T r_1} \right) r_1 + r_2, \quad \hat{H} = H_{m+1}.$$

If $(v^T r_1 = 1$ and $v^T r_2 \neq 0)$ then system (1.2) has no solution and stop.

If $(v^T r_1 = 1$ and $v^T r_2 = 0)$ then set

$$\hat{x} = r_2, \quad \hat{H} = \begin{pmatrix} H_{m+1} \\ r_1^T \end{pmatrix}.$$

(5) Stop.

(If the perturbed system has a solution, then the general solution of system (1.2) is given by $\bar{x} = \hat{x} + \hat{H}^T g$, ($g \in R^{n+1}$ OR $g \in R^n$).

REMARK 1. It can be seen that the algorithm, after substituting \bar{b} for b , without changing the other steps, can solve the system $(A + uv^T)x = \bar{b}$, $x \in R^n$, using the solution of system (1.1) (where $\bar{b} \in R^m$ is an arbitrary vector).

REMARK 2. In the case when the coefficient matrix A and vector b are integer and we need to find the integer solution of a system, a similar problem was solved by Amini and Mahdavi-Amiri [3].

4. Applications of the algorithm

4.1. Application of the algorithm to linear programming Consider the following linear programming problem:

$$\min c^T x \quad \text{s.t.} \quad \begin{cases} Ax = b, \\ x \geq 0, \end{cases}$$

where $x \in R^n$, $A \in R^{m,n}$, $n \geq m$.

At any iteration of the simplex algorithm (or dual-simplex), assume that B and N are the basic and nonbasic matrices respectively and c_B, c_N are the vectors of elements corresponding to B, N . We know that in simplex or dual-simplex methods, at any stage we need the information below ([4]):

$$x = B^{-1}b, \quad \bar{c} = c_B^T - c_B^T B^{-1}N.$$

So we need solutions of the systems $Bx = b$ and $B^T y = c_B$. So, in any iteration, in the time at which a new variable is added to the basic elements (x_i) and one variable exits the basic elements (x_j), the basic matrix B is revised by a one rank change $\hat{B} = B + ve_i^T$, where $v = a_j - a_i$.

So we need to find solutions of the following systems:

$$(B + ve_i^T)x = b \quad \text{and} \quad (B^T + e_i v^T)y = c_B.$$

The algorithm in Section 3 can be used to solve these systems.

4.2. Application of the algorithm to nonlinear programming An important class of known optimisation methods is the quasi-Newton methods, and one of the important and effective subclasses of this family is the secant method. This type of method is like Newton's method with line search, except that G_k is replaced by a symmetric positive definite matrix B_k , which is corrected or updated from iteration to iteration. Thus the k th iteration has the following basic structure:

- (a) Set s_k to be one solution of system $B_k s_k = -g_k$.
- (b) Line search along s_k giving $x_{k+1} = x_k + \alpha_k s_k$.
- (c) Update B_k giving B_{k+1} .

The initial matrix can be any positive definite matrix ([5]).

Generally, the matrix B_{k+1} can be computed from B_k by a rank one update or a rank two update.

The most important rank one update is given by

$$B_{k+1} = B_k + \frac{(\gamma - B_k \delta)(\gamma - B_k \delta)^T}{(\gamma - B_k \delta)^T \delta},$$

where $\delta = x_{k+1} - x_k$ and $\gamma = g_{k+1} - g_k$. Here g_k is the gradient vector in x_k .

Now for computing s_k , B_k at any stage, we can use Algorithm 1 for generating the solutions of system $B_k s_k = -g_k$ using the solutions of the previous system.

NOTE. It is necessary that we use the scaled ABS method in the first step for solving the first system.

REMARK 3. If B_{k+1} is computed from B_k by a two rank update (for example a DFP or BFGS update [5]), we can use Algorithm 1 twice in any step.

References

- [1] J. Abaffy, C. G. Broyden and E. Spedicato, "A class of direct methods for linear equations", *Numer. Math.* **45** (1984) 361–376.
- [2] J. Abaffy and E. Spedicato, *ABS projection algorithms: mathematical techniques for linear and nonlinear equations* (Ellis Harwood, Chichester, 1989).
- [3] K. Amini and N. Mahdavi-Amiri, "Solving rank one perturbed linear Diophantine systems by the ABS methods", submitted.
- [4] M. S. Bazaraa, J. Jarvis and H. D. Sherali, *Linear programming and network flows* (John Wiley and Sons, New York, 1990).
- [5] R. Fletcher, *Practical methods of optimization* (John Wiley and Sons, Chichester, 1991).
- [6] M. R. Hestense and E. Stiefel, "Methods of conjugate gradient for solving linear system", *J. Research Nat. Bur. Standards* **49** (1952) 409–436.

- [7] C. Lanczos, "Solution of systems of linear equations by minimized iterations", *J. Research Nat. Bur. Standards* **49** (1952) 33–53.
- [8] E. Spedicato, "Numerical methods for linear and nonlinear equations and nonlinear programming", Report DMSIA 98/4, University of Bergamo.
- [9] V. V. Voevodin, *Linear algebra* (Mir, Moscow, 1983).