SIAM J. NUMER. ANAL.
Vol. 13, No. 4, September 1976

CONSTRAINED MINIMIZATION USING POWELL'S CONJUGACY APPROACH*

A. G. BUCKLEY†

Abstract. We consider the problem of minimizing a function f of n real variables $x_1, \cdots, x_n$, subject to the provision that derivative information is not to be used in seeking the minimum. A number of methods exist for solving this problem and of these, several are based on the construction of conjugate or orthogonal search directions. The aim of this paper is to demonstrate how these methods may be adapted to work when linear constraints on the variables are present. We describe how orthogonal transformations may be applied to the search directions so that conjugacy or orthogonality relations are preserved whenever the search directions are modified in order to ensure that no constraints are violated.

1. Introduction. The problem we wish to consider is a standard one. We seek a point $\bar{x}$ which is a local minimum of a function $f(x) = f(x_1, \cdots, x_n)$, subject to the restriction that $\bar{x}$ satisfies a set of m linear constraints:

$$c_i^T x - d_i = 0, \qquad i = 1, \cdots, \bar{m}, \tag{1}$$

$$c_i^T x - d_i \geqq 0, \qquad i = \hat{m}, \cdots, m \quad (\hat{m} = \bar{m} + 1). \tag{2}$$

Our particular interest is in algorithms which iteratively seek a minimum of f by relying on function values alone, that is to say, without using derivative information.

When no constraints are present, there are several well-known algorithms available for approximating $\bar{x}$. We will be particularly interested in those given by Brent [1], Powell [9] and Zangwill [14]; these have the common property that they generate successive approximations to the minimum $\bar{x}$ by performing line searches in turn along each of a set of search directions, some of which are designed to be conjugate. (In fact, the algorithms of Brent and Zangwill are based on the one by Powell.) Our interest stems from the observation that none of these algorithms has provision for directly handling linear constraints, and it is our intention to demonstrate a technique which will allow these algorithms to efficiently handle the constraints (1) and (2). We note that the technique to be described observes the constraints without destroying any conjugacy relations which may have been constructed amongst the search directions by the algorithm. (The method can also preserve orthogonality amongst search directions and it may therefore be of interest in algorithms such as [13] which use orthogonal search directions.)

It should be noted that one can handle linear constraints by following Stewart [12] and using a quasi-Newton method with finite difference approximations to derivatives. This has been done in the constrained case by Gill and Murray [6]. However, in the paper [10] by Powell, there are reasons given (in the unconstrained case) why one should prefer to use an approach other than finite differences when analytic derivatives are not available. Since these reasons are equally valid in the constrained case, we feel there is just cause for investigating the application of search direction algorithms to the linearly constrained problem.

* Received by the editors November 19, 1974, and in revised form September 17, 1975.

† Department of Mathematics, Concordia University, Montreal, Quebec H3G 1M8. This work was supported by the National Research Council of Canada under Grant A8962.

We choose to illustrate our technique in a specific context. In particular we will show how Powell's method [9] may be adapted to work in the presence of linear constraints. We choose Powell's method for good reasons. First, it is still recognized as one of the best algorithms available. Second, both of the other algorithms are based on Powell's basic construction of conjugate directions. For example, Brent's algorithm is derived from Powell's algorithm by the addition of a routine which periodically orthogonalizes the search directions. This additional routine is only used at the end of certain iterations, and there is therefore no reason why, if one wished to implement Brent's algorithm, that one could not do the orthogonalization at the end of the iteration described in § 8. And finally, we choose to use Powell's algorithm as our model for the constrained problem because its basic simplicity allows us to exhibit our technique for handling constraints without an excessive amount of detail present to obscure our main ideas.

In §§ 2 to 7 we discuss our technique for handling linear constraints. In § 8 we summarize these sections by presenting an algorithm, based on Powell's algorithm [9], which iteratively seeks the constrained minimum $\bar{x}$ of f. Finally, in § 9, we exhibit numerical results which show that this algorithm effectively handles the constraints (1) and (2).

2. Basics. This section is devoted to introducing some basic definitions, some assumptions, and some of our notation.

The *theory* in the remainder of the paper will be developed for the special case when f is the quadratic function

$$f(x) = \tfrac{1}{2}x^TAx + b^Tx + c, \tag{3}$$

with A a symmetric positive definite $n \times n$ matrix. The algorithm will of course be designed to work on a general function and comments pertinent to the non-quadratic case will be made at the appropriate places. By examining the quadratic case we are making the standard assumption that a general smooth function f is approximately quadratic in the region of the minimum, and hence an algorithm which is well designed for a quadratic should work well in the general case. For a general function f we will let $g(x)$ denote the gradient vector at x, and we recall that when f is given by (3),

$$g(x) = Ax + b.$$

The basic concept on which Powell's algorithm is based is that of conjugacy. Given a symmetric positive definite matrix A, we say the directions $d_k, \cdots, d_p$ are mutually conjugate (with respect to A) if

$$d_i^TAd_j = 0 \quad \text{for} \quad k \leqq i < j \leqq p.$$

Clearly then conjugacy is well defined by the quadratic function f we are considering. Notice that, in this paper, when we introduce a set of directions $d_1, \cdots, d_p$, we will frequently assume there is a k, $1 \leqq k \leqq p$, such that $d_k, \cdots, d_p$ are conjugate.

In order to produce conjugate directions, and at the same time to seek the minimum of the function f of (3), we will generate a sequence of points $x_0, x_1, \cdots, x_p$. Given x_0, the point x_i, $i = 1, \cdots, p$, will be obtained by a line search along the direction d_i from the previous point x_{i-1}. That is, an appropriate value of s_i is determined and then we set

$$x_i = x_{i-1} + s_i d_i.$$

If the value of s_i used is such that x_i is a local minimum of f along the line through x_{i-1} and in the direction d_i, we write

$$x_i = x_{i-1} + \hat{s}_i d_i.$$

In common parlance, this means that we have an *exact* line search and we note that, in this case,

$$(d_i, g(x_i)) = 0.$$

At this point, only one remark will be made concerning the constraints (1) and (2). In practice, not all of the constraint vectors $c_1, \cdots, c_m$ need be independent, and so a practical algorithm must include checks for redundancy and must act accordingly. In the theoretical development we will disregard this possibility, as the modifications to handle redundancy do not contribute to our main ideas.

3. Powell's 1964 method. In order to facilitate the description of the new algorithm, we will now describe the basic iteration which Powell [9] introduced in 1964 to minimize f when no constraints are present.

POWELL'S ALGORITHM. We begin with an initial point x_0 and with a set of independent directions $d_1, \cdots, d_n$. Then for $i = 1, 2, \cdots, n$ we define x_i by

$$x_i = x_{i-1} + \hat{s}_i d_i. \tag{4}$$

Finally we set

$$d_{n+1} = x_n - x_0$$

and redefine

$$x_0 = x_n + \hat{s}_{n+1} d_{n+1}. \tag{5}$$

Then we renumber our search directions so that d_1 is discarded. That is,

$$d_i \leftarrow d_{i+1}, \qquad i = 1, \cdots, n,$$

where "←" is read "is replaced by". This completes one iteration of Powell's algorithm.

The main characteristics of the algorithm, which follow from the theorems stated in the next section, are the following. Initially the directions d_i are merely independent, with no conjugacy relations present. However, after completion of the rth iteration (which includes relabeling the search directions) the last r directions will be mutually conjugate; that is, with $k = n - r + 1$, $d_k, \cdots, d_n$ will be mutually conjugate. The fundamental step in developing these conjugacy relations is the final line search (5) by which x_0 is redefined. Finally then, after n iterations, $d_1, \cdots, d_n$ are mutually conjugate, and it is well known (see Powell [9] for example) that consecutive exact line searches along $d_1, \cdots, d_n$—that is, steps (4) and (5) of the nth iteration—will give precisely the minimum of the quadratic f, i.e., at x_0. For a general function f one simply repeats the iteration until a point sufficiently close to a minimum has been reached. Because f is approximately quadratic near the minimum, directions d_i will be developed which are (in some sense) approximately conjugate and hence the ultimate rate of convergence should be rapid.

At this point we consider one other property of the algorithm, primarily because we wish to refer to it as a precedent at a later point in our development. Powell pointed out in 1964 that the basic iteration described above must in practice be modified, for two reasons. First, $\hat{s}_1$ may be 0, in which case the new directions $d_1, \cdots, d_n$ will be linearly dependent. Second, it has been observed experimentally that the directions $d_1, \cdots, d_n$ show a tendency towards dependence after a number of iterations. Both of these facts can make it difficult to find a minimum. As a result, Powell proposed a modification to the basic algorithm which avoids both of these problems, but with the result that, on occasion, the new direction d_{n+1} may not be accepted, so that the next iteration uses the same set of search directions as its predecessor. It is accepted that this change is essential to make the algorithm truly viable.

4. Two basic theorems. We now state the two results which explain why Powell's algorithm generates conjugate directions, and which will be used to develop conjugacy when constraints are present.

THEOREM 1. *Let $z_k, \cdots, z_p$ be conjugate. Suppose x_{k-1} is given and suppose that for arbitrary s_i,*

$$x_i = x_{i-1} + s_i z_i, \qquad i = k, \cdots, p.$$

Then

$$(z_i, g(x_j)) = (z_i, g(x_t)), \qquad k \leqq i \leqq j \leqq t \leqq p.$$

Proof. For $k \leqq i \leqq j < t \leqq p$ we have

$$\begin{aligned}(z_i, g(x_t)) &= (z_i, Ax_t + b) = \left(z_i, A\left(x_j + \sum_{r=j+1}^{t} s_r z_r\right) + b\right) \\ &= (z_i, Ax_j + b) + \sum_{r=j+1}^{t} s_r(z_i, Az_r) \\ &= (z_i, g(x_j))\end{aligned}$$

since $z_{j+1}, \cdots, z_t$ are conjugate to z_i. □

This theorem simply says, given a point x_i and a direction z_i, that moving from x_i along directions conjugate to z_i does not change the directional derivative in direction z_i. The case of particular interest is when $s_i = \hat{s}_i$, in which case $(z_i, g(x_i)) = 0$, so

$$(z_i, g(x_j)) = 0 \quad \text{for} \quad k \leqq i \leqq j \leqq p.$$

THEOREM 2. *Let a set of directions $d_1, \cdots, d_p$ be given with $d_k, \cdots, d_p$ mutually conjugate and let x_0 be given such that $(d_k, g(x_0)) = \cdots = (d_p, g(x_0)) = 0$. Let x_{k-1} be arbitrary, and let*

$$x_i = x_{i-1} + s_i z_i \qquad \textit{for } i = k, \cdots, t,$$

where $k \leqq t \leqq p$ and where, for $i = k, \cdots, t$, z_i is any linear combination of $d_i, \cdots, d_p$

such that $z_k, \cdots, z_t$ are conjugate. Then for each i with $k \leqq i \leqq t$ and $s_i = \hat{s}_i$, $x_t - x_0$ is conjugate to z_i.

Proof. Choose i with $k \leqq i \leqq t$. Because $z_i, \cdots, z_t$ are conjugate, Theorem 1 applies with $p = t$, so

$$\begin{aligned}(z_i, A(x_t - x_0)) &= (z_i, Ax_t + b) - (z_i, Ax_0 + b)\\ &= (z_i, g(x_t)) - (z_i, g(x_0))\\ &= ((z_i, g(x_i)) - \sum_{j=i}^{p} b_{ij}(d_j, g(x_0)),\end{aligned}$$

where we have written $z_i = \sum_{j=i}^{p} b_{ij} d_j$. But by assumption each term $(d_j, g(x_0)) = 0$, and the choice $s_i = \hat{s}_i$ implies $(z_i, g(x_i)) = 0$, so we conclude that $x_t - x_0$ is conjugate to z_i. $\square$

Observe the following regarding Theorem 2. First, the conjugacy of $x_t - x_0$ to z_i depends only on s_i being $\hat{s}_i$ and on the conjugacy of $z_i, \cdots, z_t$. A choice of $s_j \neq \hat{s}_j$ for $j > i$ does not destroy the conjugacy. The second remark concerns the introduction of the directions z_i. In Powell's original algorithm this is not necessary (that is, $z_i = d_i$ for $i = k, \cdots, p = n$), but in the proposed algorithm the presence of constraints may cause the search directions to be altered in order that the constraints will not be violated. We shall see in the next section how this may be done, and in particular it will be done so that the conditions of Theorem 2 are satisfied.

5. The rotation matrices. In order to adapt Powell's method to handle linear constraints we will follow the well-known active set strategy. This means that, if our current approximate minimum is x, we identify those constraints which are satisfied as equalities at x and we call (some of) them the set of *active* constraints. The minimization then proceeds by moving from x to a new point $\hat{x}$ which also satisfies the active constraints exactly and for which $f(\hat{x}) \leqq f(x)$. Then the set of active constraints may be modified by adding or deleting constraints. The question of how to add constraints is dealt with in this section, while removal of constraints is deferred to § 7.

To be specific, let $n_1, \cdots, n_q$ be the indices of the active constraints at x, so that

$$c_{n_i}^T x - d_{n_i} = 0, \qquad i = 1, \cdots, q.$$

(Clearly, assuming independence, we would define $n_j = j$ for $j = 1, 2, \cdots, \bar{m}$.) Thus $\hat{x}$ is obtained as a point which reduces the value of the function and which lies in the linear manifold M through x and orthogonal to $c_{n_1}, \cdots, c_{n_q}$.

In order to implement the active constraint approach in the present context, we must ensure that the search directions d_i lie in M. This means therefore, that should M change, as it will whenever the active set changes, we must have a means of modifying the directions d_i so that they lie in the new manifold. And of course, in order to develop conjugacy relations amongst the d_i, this change to the d_i should be done in a way which preserves any conjugacy relations already established amongst them. That we are able to accomplish these aims is a result of a theorem which we shall state after introducing some notation.

Let $d_1, \cdots, d_p$ be a given set of independent directions with $d_k, \cdots, d_p$ mutually conjugate. We define the $n \times p$ matrix

(6) $$D = (d_1, \cdots, d_p),$$

and we normalize these search directions by defining

(7) $$L = \operatorname{diag}(d_i^T A d_i), \qquad i = 1, \cdots, p,$$

and setting

(8) $$D' = (d_1', \cdots, d_p') = DL^{-1/2}.$$

It is worth observing at this point that, although the matrix A appears explicitly in (7), the normalization may be done without explicit knowledge of A. For, the number $d_i^T A d_i$ represents the second derivative of f in the direction d_i, and this derivative may be obtained from the second difference of 3 function values taken along d_i. Of course these function values will be available from the line search, and indeed, the normalization may be done in this way even for a non-quadratic function.

Now choose an orthogonal $p \times p$ matrix Q and define a new set of search directions $\bar{d}_1, \cdots, \bar{d}_p$ by setting

(9) $$\bar{D} = (\bar{d}_1, \cdots, \bar{d}_p) = D'Q.$$

Our fundamental result is then:

THEOREM 3. *Suppose independent vectors $c_{n_1}, \cdots, c_{n_q}$ are given. Set $p = n - q$ and suppose $d_1, \cdots, d_p$ are independent and orthogonal to $c_{n_1}, \cdots, c_{n_q}$, with $d_k, \cdots, d_p$ mutually conjugate. Let $c_{n_{q+1}}$ be given, independent of c_{n_i}, $i = 1, \cdots, q$. Then an orthogonal matrix Q can be constructed so that, using (6)–(9), we get new directions $\bar{d}_i$ for which:*

(a) *$\bar{d}_1, \cdots, \bar{d}_{p-1}$ are independent and orthogonal to $c_{n_1}, \cdots, c_{n_{q+1}}$;*

(b) *$\bar{d}_k, \cdots, \bar{d}_{p-1}$ are mutually conjugate.*

Proof. For convenience write c for $c_{n_{q+1}}$, and define $v = D'^T c$. Let $Q_1, \cdots, Q_{p-1}$ be the sequence of orthogonal Given's rotation matrices such that Q_i rotates in the plane $(p - i, p)$ and such that $Q_i^T \cdots Q_1^T v$ has a 0 in components $p - i, \cdots, p - 1$. Let $Q = Q_1 \cdots Q_{p-1}$.

Because $\bar{d}_i$ is then a linear combination of $d_i, \cdots, d_p$, each $\bar{d}_i$, $i = 1, \cdots, p$, is automatically orthogonal to c_{n_j}, $j = 1, \cdots, q$. Furthermore,

$$c^T \bar{D} = (0, 0, \cdots, 0, \|v\|_2)$$

and

$$c^T \bar{D} = (c^T \bar{d}_1, \cdots, c^T \bar{d}_p),$$

so that $\bar{d}_1, \cdots, \bar{d}_{p-1}$ are also orthogonal to c, and (a) is established.

To see (b), observe that if y and z are conjugate and normalized with respect to A (that is, $y^T A z = 0$ and $y^T A y = z^T A z = 1$), and if we define, for any θ,

$$\bar{y} = (\cos\theta) y - (\sin\theta) z,$$

$$\bar{z} = (\sin\theta) y + (\cos\theta) z,$$

then $\bar{y}$ and $\bar{z}$ are also conjugate and normalized. Also, if ω is conjugate to y and z, then it is conjugate to $\bar{y}$ and $\bar{z}$. From these comments and the method of construction of $\bar{D}$, (b) follows. □

Three important comments are in order. First, let

$$\hat{D} = D'Q_1 \cdots Q_{p-k}. \tag{10}$$

Clearly $\hat{d}_i = \bar{d}_i, i = k, \cdots, p - 1$. Furthermore, the argument of the theorem shows that $\hat{d}_p$ is conjugate to $\bar{d}_k, \cdots, \bar{d}_{p-1}$. Note the definition of $\hat{d}_p$ because this vector appears in Theorem 6.

The second remark concerns construction of the matrix Q and is included so as to explain to the reader why we do not use (as is often the case) a Householder reflection matrix to reduce v. Clearly it is straightforward here to construct a single Householder matrix which will reduce v to $\|v\|_2(0, \cdots, 0, 1)^T$, whence (a) follows. But then the directions $\bar{d}_k, \cdots, \bar{d}_{p-1}$ defined by (9) do not retain their conjugacy properties, for each $\bar{d}_i$ will contain components of every one of the original directions, including the nonconjugate directions $d_1, \cdots, d_{k-1}$. This difficulty can be avoided by using two Householder transformations, the first to reduce v to $(v_1, \cdots, v_{k-1}, 0, \cdots, 0, v_p')^T$, and the second to reduce this to $\|v\|_2(0, \cdots, 0, 1)^T$. Then the theorem follows as before. However, we still prefer the Given's approach, the reason being that the sequence of rotations preserves conjugacy with no explicit reference to k, which is clearly not the case for the pair of reflections. The importance of not explicitly needing to know k will be discussed in the next section.

Finally we wish to point out an added advantage to using orthogonal transformations. We begin by stating the following theorem given in Powell [10, p. 7]:

THEOREM 4. *Let $p = n$ and let n directions $d_1, \cdots, d_n$ be given, with D' defined by* (6) *and* (8). *Then the maximum value of* $|\det D'|$ *is attained if and only if $d_1, \cdots, d_n$ are mutually conjugate.*

Observe as well that if the directions $d_1, \cdots, d_n$ are dependent, then they are certainly not conjugate and $|\det D'|$ attains its minimum of 0. Thus we can accept $|\det D'|$ in some sense as a measure of the "degree of conjugacy" of the directions d_i. It is then natural to wish to compare the degree of conjugacy of the given directions d_i with that of the directions $\bar{d}_1, \cdots \bar{d}_n$ defined by (9). This relation is also given by Powell [10, p. 7]:

THEOREM 5. *Given D' as before, let Q be any orthogonal matrix. Define new directions $\bar{D} = D'Q$, as in* (9), *and normalize these by letting $d_i'' = \bar{d}_i/(\bar{d}_i, A\bar{d}_i)^{1/2}$. Then* $|\det D''| \geqq |\det D'|$.

The significance of this result is clear, It assures us that, aside from preserving specific conjugacy relations, the procedure of Theorem 3 never gives us a new set of directions whose overall degree or measure of conjugacy is worse.

6. Conjugacy with constraints. We now wish to combine the results of §§ 4 and 5 so that conjugacy may be developed by consecutive line searches, as suggested by Powell, but without violating any constraints. Throughout this section it is assumed as usual that, unless stated otherwise, f is quadratic. The results of this section will be summarized in the algorithm outlined in § 8.

Suppose a point x_0 is given. Let $n_1, \cdots, n_q$ denote the indices of the active constraints at x_0. The dimension of the manifold M through x_0 and orthogonal to $c_{n_1}, \cdots, c_{n_q}$ is then $n - q$, and so, in order to seek the minimum of f in M it is clear that we require a set of $p = n - q$ independent search directions, each of which is orthogonal to c_{n_i}, $i = 1, \cdots, q$. Let $d_1, \cdots, d_p$ be such a set and assume $d_k, \cdots, d_p$ are conjugate. Furthermore, in accordance with Powell's algorithm (see § 3 and Theorem 2), assume $(d_k, g(x_0)) = \cdots = (d_p, g(x_0)) = 0$.

We now apply Powell's iteration of line searches, thus generating points $x_1, \cdots, x_p$ which seek the minimum of f in M. This will proceed exactly as in the iteration (4), (5) described by Powell, unless for some i, x_i would violate some constraint. If the iteration is completed with no new constraints being encountered, then the situation is just as in Powell's original algorithm, and we obtain the search directions for the next iteration in the same way. The choice of these search directions will be subject to the same precautions as suggested by Powell [9] and mentioned earlier in § 3.

But now consider the situation where new constraints are encountered, and let r be the first index which leads to a nonfeasible point. In this case we redefine x_r to be on the boundary of the feasible region. That is, we find s_r such that

$$\hat{x}_r = x_{r-1} + s_r d_r \tag{11}$$

is feasible and lies on the boundary of some new (i.e., previously inactive) constraint halfspace, and we set $x_r = \hat{x}_r$. We assume that $f(x_r) \leqq f(x_{r-1})$ (otherwise we could find a local minimum along d_r which is inside the feasible region). Let $c_{n_{q+1}}$, or c for short, denote the new constraint which has become active. We must now add c to our set of active constraints and change the manifold M to a new manifold $\bar{M}$. We therefore apply Theorem 3, which gives us a new set of directions $\bar{d}_1, \cdots, \bar{d}_{p-1}$ lying in $\bar{M}$ and with $\bar{d}_k, \cdots, \bar{d}_{p-1}$ still conjugate.

There are two important questions to be answered:

1. After constructing $\bar{d}_1, \cdots, \bar{d}_{p-1}$, and starting from x_r,
   (A) do we continue our line searches along the directions $\bar{d}_{r+1}, \cdots, \bar{d}_{p-1}$ (or perhaps along $\bar{d}_r, \cdots, \bar{d}_{p-1}$); or
   (B) do we essentially restart the iteration by searching in turn along $\bar{d}_1, \cdots, \bar{d}_{p-1}$?
2. How do we construct a new conjugate direction?

For now we assume that method (B) is followed and we answer question 2. It will then be appropriate to explain why (A) is not acceptable.

Consider the situation where on $s \geqq 1$ occasions during the iteration, a line search leads to an infeasible point, with the result that a new constraint is added to the active set. When the jth constraint is added, apply Theorem 3 to obtain a new set of directions $d_1^j, \cdots, d_{p-j}^j$ lying in a manifold M_j and choose as next search direction d_1^j. Thus, after adding the sth constraint, we finish the iteration with $t = p - s$ exact line searches along $d_1^s, \cdots, d_t^s$ lying in M_s. (Observe that this procedure must terminate since, as we will discover in the next section, no constraints are ever removed from the active set during an iteration.) We now recall that the object of Powell's unconstrained conjugacy algorithm is to create a new direction d which is conjugate to $d_k, \cdots, d_n$. In the present context, assuming

$t \geqq k$, we seek therefore a new direction d which lies in M_s and which is conjugate to $d_k^s, \cdots, d_t^s$. (The case $t < k$ is of no interest since all conjugacy has been lost.)

Before showing how to find d, a comment should be made. It is that we will not adopt the procedure to be described, for reasons to be given later. We nonetheless wish to describe the method for constructing d since readers who wish to implement Brent's orthogonalization may wish to use this procedure and the direction d it yields, as will also be explained later.

To construct d, we first recall that Powell chooses $d = x_n - x_0$. Now, when $t \geqq k$, we know by Theorem 3 that $d_k^s, \cdots, d_t^s$ are mutually conjugate. Furthermore, these new directions have been constructed so that for each i, d_i^s is a linear combination of $d_i, \cdots, d_p$. Thus, because Theorem 2 allows complete freedom in the choice of x_{k-1}, we can apply it to conclude that $x_t - x_0$ is conjugate to $d_k^s, \cdots, d_t^s$. However $x_t - x_0$ need not lie in M_s. To resolve this difficulty, we use (10). To be more specific, the ith time we add a constraint, we find from Theorem 3 and from (10) that an extra conjugate direction (corresponding to $\hat{d}_p$) is produced. We call these directions $\hat{d}_p, \cdots, \hat{d}_{p-s+1}$, where $\hat{d}_{p-i}$ is conjugate to $d_k^{i+1}, \cdots, d_{p-i-1}^{i+1}$ for $i = 0, \cdots, s-1$. We can then state

THEOREM 6. *Constants $a_0, \cdots, a_s$ may be determined so that*

$$d = a_0(x_t - x_0) + \sum_{i=1}^{s} a_i \hat{d}_{p-i+1}$$

lies in the manifold M_s and is conjugate to $d_k^s, \cdots, d_t^s$, when $t \geqq k$.

Proof. Let $z_0 = x_t - x_0$ and let $z_i = (\sin\theta_i)z_{i-1} + (\cos\theta_i)\hat{d}_{p-i+1}$ for $i = 1, \cdots, s$. For $i \geqq 1$ assume z_{i-1} is orthogonal to $c_{n_1}, \cdots, c_{n_{q+i-1}}$ and conjugate to $d_k^s, \cdots, d_t^s$. This is true for $i = 1$. Inductively then consider $i > 1$. Recall that $\hat{d}_{p-i+1}$ is orthogonal to $c_{n_1}, \cdots, c_{n_{q+i-1}}$. Therefore, for suitable choice of θ_i (which can be determined much as in Theorem 3), we can make z_i orthogonal to $c_{n_1}, \cdots, c_{n_{q+i}}$. Furthermore, $\hat{d}_{p-i+1}$ is conjugate to $d_k^i, \cdots, d_{p-i}^i$, and hence, for any $j \geqq i$, $\hat{d}_{p-i+1}$ is conjugate to $d_k^j, \cdots, d_t^j$. Thus both z_{i-1} and $\hat{d}_{p-i+1}$ are conjugate to $d_k^s, \cdots, d_t^s$, so z_i is also. The theorem thus follows with $d = z_s$. $\square$

From Theorem 6 it is clear that we can construct an acceptable new conjugate direction d, provided we have saved the directions $\hat{d}_p, \cdots, \hat{d}_{p-s+1}$. If we did in fact compute and accept d, we would then complete this iteration by searching along d from x_t to obtain a new point x_0, as in [9]. Provided that no new constraints were active at x_0, we would then prepare to start the next iteration by setting $p = t$ and by choosing directions $d_1, \cdots, d_p$ to be $d_2^s, \cdots, d_t^s$ and d—subject to Powell's precautions; see § 8, Step 6. Notice that, because we chose approach (B), this iteration would then have finished with searches along $d_2^s, \cdots, d_t^s$ and d so that, by Theorem 1, we would start the next iteration with $(d_k, g(x_0)) = \cdots = (d_p, g(x_0)) = 0$, as required by Theorem 2. (If a new constraint were active at x_0, we would just use $d_1^s, \cdots, d_t^s$ for the next iteration since these inner products would not be 0.)

Unfortunately we feel that this construction of d is not acceptable in practice. To explain why, we observe that the construction used the directions $\hat{d}_p, \cdots, \hat{d}_{p-s+1}$. Now, it follows from (10) that each $\hat{d}_i$ can be determined only if a value is known for k. But we have already indicated in the previous section that we prefer an algorithm not requiring an explicit knowledge of k and we promised to explain why. The reason is that, although the theoretical basis for the algorithm lies in properties observed when f is quadratic, its true usefulness is in the minimization of nonquadratic functions. In this general context, Powell's algorithm does not in fact generate conjugate directions and therefore there is no value defined for k. There is of course the artificial device available of imposing on the algorithm a restarting feature where initially k is set to $p + 1$, then decremented by 1 each iteration and then reset to $p + 1$ when it reaches 0. Although this sort of technique has proved worthwhile in some algorithms (notably the Fletcher–Reeves conjugate gradient method [4]), we do not feel that there is any justification for this approach in this context, simply because in the general case the current value of k would then bear no relation to the number of directions which were (in some approximate sense) conjugate.

In the context of Brent's algorithm however, one has already imposed a restarting feature on the algorithm by periodically orthogonalizing the search directions. Therefore one may feel that it is reasonable to artificially use k as described, in which case Theorem 6 can be used to determine a new conjugate direction, even when constraints are added during the iteration.

However if Powell's original algorithm is followed, we suggest that Theorem 6 can not be used. Therefore we do not attempt to determine a new search direction d. Instead we simply set $x_0 = x_t$, reset $p = t$, rename $d_1^s, \cdots d_t^s$ as $d_1, \cdots, d_p$ and continue with the next iteration. Notice that, because searches were performed along $d_k, \cdots, d_p$, we again have by Theorem 1 that the next iteration begins with $(d_k, g(x_0)) = \cdots = (d_p, g(x_0)) = 0$. By Theorem 2 this is necessary in order to be able to introduce a new conjugate direction on the *next* iteration (providing no new constraints are encountered there).

We have now described what to do when new constraints are encountered. There are two points about which we wish to reassure the reader. First, is it bad that we perform an iteration of the algorithm without getting a new conjugate direction? Well, in § 3 we observed that Powell's algorithm contained a modification which had the effect of sometimes using the same set of search directions in two or more successive iterations. Powell also stated that, not only was this not harmful, it was essential to be able to minimize a function of more than 10 variables. Thus we have a precedent for leaving the search directions unchanged, and we feel therefore that it is not a serious weakness. Second, observe that we restart the line searches with the first search direction each time a constraint is added. It appears therefore that we are introducing a bias towards searching in the subspace spanned by the first few search directions. We do not however think that this is the case, for when we compute the modified directions $d_1^j, \cdots, d_{p-j}^j$ from $d_1^{j-1}, \cdots, d_{p-j+1}^{j-1}$, each direction d_i^j contains components of every direction $d_i^{j-1}, \cdots, d_{p-j+1}^{j-1}$. This should eliminate any bias.

We conclude this section by showing that the answer to question 1(A) is no. Suppose for convenience that $s = 1$. Recalling the definition of x_r from (11), and assuming that $r \geqq k$, we see that, in case (A), such an iteration would consist of searches along $d_1, \cdots, d_k, \cdots d_r, d_{r+1}^1, \cdots, d_{p-1}^1$. It follows then as in Theorem 2 that $x_{p-1} - x_0$ is conjugate to $d_k, \cdots, d_r$ and to $d_{r+1}^1, \cdots d_{p-1}^1$. Unfortunately it is not true that $x_{p-1} - x_0$ is conjugate to $d_k^1, \cdots, d_r^1$, and that of course is what is needed to be able to augment our set of conjugate directions. Furthermore, $x_{p-1} - x_0$ does not lie in the new manifold M_1, and to transform $x_{p-1} - x_0$

into d lying in M_1 would again require knowledge of k. These difficulties could be avoided just as in method (B) by proceeding to the next iteration without changing the set of search directions. But another difficulty arises. We have seen that one of the properties necessary in order to compute a new conjugate direction at the end of an iteration is that the iteration was begun with $(d_k, g(x_0)) = \cdots = (d_p, g(x_0)) = 0$ (see Theorem 2). This property is obtained by performing a line search along each of $d_k, \cdots, d_p$ during the previous iteration. In the present situation, using (A), a line search would *not* be done along $d_k^1, \cdots, d_r^1$, so $(d_k^1, g(x_{p-1})), \cdots, (d_r^1, g(x_{p-1}))$ need not be zero. Therefore even the *next* iteration could not be used to generate a conjugate direction. In comparison to method (B) we do not feel that this is acceptable.

7. Removing constraints. In any algorithm which seeks a minimum of f subject to linear constraints (1) and (2) and which follows an active set strategy, some criterion must be available for deciding when to allow a constraint to be removed from the active set. This then allows successive points to move off of that constraint boundary and into the interior of the corresponding halfspace. In those algorithms which utilize computed first derivatives, such a criterion is readily available, for example from the Lagrange multipliers (see Goldfarb [7]) or from gradients in a transformed space (see Buckley [2]). Such techniques are clearly not available to us. However the following approach seems reasonable.

Let the $n \times q$ matrix N be defined by $N = (c_{n_1}, \cdots, c_{n_q})$, where as before $c_{n_1}, \cdots, c_{n_q}$ are the active constraints at the current point x. We recall that the Moore–Penrose generalized inverse of N is given by

$$N^\dagger = (N^T N)^{-1} N^T,$$

we let $r_1^T, \cdots, r_q^T$ denote the q rows of $N^\dagger$, and we assume $(N^T N)^{-1}$ has been computed. We call r_i a downhill direction from x if there exists an $\hat{a} > 0$ with $f(x + \hat{a} r_i) < f(x)$, and with

$$f(x + a r_i) \leqq f(x)$$

for all $0 \leqq a < \hat{a}$. We then have (see Fletcher [5], for example), for a general smooth function f,

THEOREM 7. *Let x be the current point, with active constraints given by N. Suppose x is a constrained local minimum of f. Then none of the directions $r_{\hat{m}}, \cdots, r_q$ is downhill from x.*

Proof. The theorem is just a restatement of well-known results concerning Lagrange multipliers. □

We apply this theorem to the present situation in the following way. Assume we are at the beginning of an iteration, with the current point given as x_0. Before we begin the conjugacy cycle, we will in turn perform a line search along each of the directions $r_{\hat{m}}, \cdots r_q$. If for any of these we obtain a feasible point

$$\hat{x} = x_0 + s_i r_i, \qquad \hat{m} \leqq i \leqq q,$$

with $f(\hat{x}) < f(x_0)$ and with $s_i > 0$, then we will accept r_i as downhill and we remove the appropriate constraint vector from the active set. $(N^T N)^{-1}$ is recomputed. The manifold M now has dimension $p + 1$ so we renumber the search directions

$d_1, \cdots, d_p$ as $d_2, \cdots, d_{p+1}$ and append r_i as the new first direction, i.e. as d_1. Thus the search along d_1 has already been done, so we set $x_1 = \hat{x}$ and continue the conjugacy cycle with searches in directions $d_2, \cdots, d_{p+1}$.

Of course we can remove more than one constraint without affecting the conjugacy building properties of the algorithm. At x_1 we simply repeat this removal procedure, inserting an accepted downhill direction as d_2 and renumbering the current directions $d_2, \cdots, d_{p+1}$. This can be repeated as often as necessary in the obvious way. (Note that, each time we can consider all available rows of $N^\dagger$, or else just those not already considered. The latter alternative is followed in § 8.)

There are some practical considerations. First, if we are minimizing a nonquadratic function, then, because of the fact that in practice each line search uses only a few function values, it may happen that each of the line searches along r_i, $i = \hat{m}, \cdots, q$, will yield an $s_i \leqq 0$, whereas in fact some r_j, $\hat{m} \leqq j \leqq q$, is downhill from x_0. This is not important unless at x_0 we have already established that little or no progress can be made in reducing f within M, which is to say that to the algorithm, x_0 appears to be a minimum within M. If this is not the case, obtaining $s_j \leqq 0$ when a more careful search could have obtained $s_j > 0$ can be taken as an indication that better reduction of f can be expected by remaining in M rather than by expanding M. However, if x_0 appears to be very nearly a local minimum within M, and if we have also obtained $s_{\hat{m}}, \cdots, s_q \leqq 0$, then really all we can do is to reperform the line searches along $r_{\hat{m}}, \cdots, r_q$, being especially careful to examine values of each s_i which are small and positive. This should not however be considered a defect in the algorithm but rather just another guise of the problem encountered by any minimization algorithm which does not have access to derivative information that it may be quite difficult to establish with confidence that the point obtained is a minimum. In summary then, we accept x_0 as a minimum if no progress can be made within M and when we are satisfied that no r_i is downhill.

Second, we must ask how often one should test for constraint removal. To begin, observe that a substantial amount of work must be expended in searching along $r_{\hat{m}}, \cdots, r_q$, especially if $q - \hat{m}$ is large. Thus it is obvious that such testing should be limited in its frequency; in particular it should certainly not be done at every point generated by the algorithm. An obvious alternative is to test only at the beginning of each iteration. (Notice that our notation above has already implied this procedure.) In this way the relations used in § 6 to establish conjugacy are not interfered with by introduction of the extra line searches needed to test for removal, and furthermore, if some r_i is accepted as being downhill it fits in naturally, as we have seen, as the first step in the new iteration.

A final point concerns computation of $N^\dagger$. First, notice that we require the rows r_i^T one at a time (starting with $r_{\hat{m}}^T$) and that if some r_i is accepted as downhill, there is no need to compute $r_{i+1}, \cdots, r_q$. Thus we choose to compute $(N^TN)^{-1}$ and determine r_i^T by multiplying the ith row of $(N^TN)^{-1}$ by N^T. The technique for computing $(N^TN)^{-1}$ is well known. At the very beginning of the algorithm, $(N^TN)^{-1}$ is computed. During the algorithm, changes to the active set, by removal or addition of constraints, change N by removal or addition of one column at a time, and rank 1 formulas for computing the revised matrix $(N^TN)^{-1}$, and which require only $O(n^2)$ operations, can be found in Goldfarb [7], Rosen [11] or Buckley [2]. We should mention that in some cases the computation outlined here is not the most efficient. Specifically, if many of the rows $r_{\hat{m}}^T, \cdots, r_q^T$ were to be computed each time, and if $\hat{m}$ was not large, then it would be better to directly store and update the matrix $N^{\dagger}$. Alternately one may follow the more numerically stable approach of storing and updating an orthogonal decomposition of N and then solving for the rows r_i^T.

8. The constrained algorithm. We now summarize the preceding sections by outlining one iteration of Powell's algorithm with the constraint handling modifications included. The three key points to note are the provisions for removal of constraints in Step 2 and for addition of constraints in Step 8, and the fact that in certain circumstances—Steps 5–7—we proceed to the next iteration with no change in the set of search directions. Some details which are not concerned with the handling of the constraints and which are available elsewhere have been omitted from this description. Unnecessary subscripting has also been eliminated.

We assume that we are at the beginning of an iteration at a point x_0, with active constraints given by $N = (c_{n_1}, \cdots, c_{n_q})$, with $(N^TN)^{-1}$ stored and with search directions $d_1, \cdots, d_{p=n-q}$ given which satisfy $N^Td_i = 0$ for $i = 1, \cdots, p$.

POWELL'S ALGORITHM WITH LINEAR CONSTRAINTS.

Step 1. Initialize. Set $i = 0$ and $x = x_0$.

Step 2. Test for removal of constraints. If $q = \overline{m}$, skip to Step 3. Otherwise, search in turn along $r_{\hat{m}}, \cdots, r_q$ until a downhill direction is found, say at r_j. If no r_j is found to be downhill, proceed to Step 3. Otherwise, if the search along r_j finds a point $\hat{x}$ with $f(\hat{x}) < f(x)$, then $i \leftarrow i + 1$, $d_{s+1} \leftarrow d_s$ for $s = p, \cdots, i$ and $d_i \leftarrow r_j$. The constraint corresponding to r_j is dropped from the active set and $(N^TN)^{-1}$ is recomputed by a formula given in [2], [7] or [11]. Also, $p \leftarrow p + 1$ and $x \leftarrow \hat{x}$. If no new constraint is encountered at $\hat{x}$, repeat this step, starting with the new direction r_j; otherwise add the new constraint as in Step 8 and repeat this step starting with r_1.

Step 3. Do Powell's line searches. First, $i \leftarrow i + 1$. If $i > p$ go to Step 5. Otherwise, perform a line search along d_i from x to obtain a new point $\hat{x} = x + s_id_i$.

Step 4. Test for constraint violation. If a new constraint c is active at $\hat{x}$, skip to Step 8. If not, $x \leftarrow \hat{x}$ and we return to Step 3.

Step 5. End of the line searches. If a constraint was added at any time during this iteration, simply go to Step 9. If not, proceed to Step 6.

Step 6. Apply Powell's precautions (see [9]). Compute the new search direction $d_{p+1} = x - x_0$. Using the points x_0, x and $\frac{1}{2}(x_0 + x)$, apply the precautions described in [9, p. 157] to determine whether d_{p+1} should be accepted or rejected. If it is not acceptable, skip to Step 9.

Step 7. Perform a line search along d_{p+1} from x to a point $\hat{x}$. If the line search encounters a new constraint, skip to Step 9. If not, drop the appropriate one of the original directions and renumber the remaining ones as $d_1, \cdots, d_p$. Then $x \leftarrow \hat{x}$ and we finish with Step 9.

Step 8. At least one new constraint is active, and possibly several need to be added to the active set. For each new active constraint c, set $\hat{N} = (N : c)$, compute $(\hat{N}^T\hat{N})^{-1}$ (see [2], [7] or [11]) and replace N by $\hat{N}$. For each new constraint c, we must also revise the search directions according to § 5. So, for each constraint

TABLE 1

Test problems

Problem	Beale	Shell Primal	Gauthier
Number of Variables	3	5	16
Number of Constraints	4	15	40 (8 equalities)
Starting Point	0, 0, 0	0, 0, 0, 0, .9	0, .9026518, 0, .3779330, 1.364408, .7436653, 2.476527, 0, 1.641335, 0, 0, 0, .1586814, 0, 0, .2690046,
Function Value	9.0	14.95800	312.7313
Number of Active Constraints	3	4	16
Solution Found	1.333333, .7777778 .4444444	.3000000, .3334058, .4000000 .4281597, .2241099	.03982150, .7918948, .2026242, .8439505, 1.269789, .9347277, 1.682262, .1556755, 1.567973, 0, 0,.0, .6600665, 0, .6741013, 0
Function Value	.1111111	−32.34868	244.8987
Number of Active Constraints	1	4	13
Number of Line Searches			
Total	19	38	66
Used to test for removal	7	31	48
Number of removals rejected	4	22	43
Number of Function Evaluations	48	75	127
Number of Constraint Removals	3	9	5
Number of Constraint Additions	1	9	2

being added, compute a new set of directions as in Theorem 3, replace p by $p - 1$ and rename the revised search directions as $d_1, \cdots, d_p$. Finally, set $i = 0$, set $x = \hat{x}$ and go back to Step 3.

Step 9. This iteration is complete. Set $x_0 = x$ and return to Step 1.

Termination occurs when Step 2 indicates that no direction $r_{\hat{m}}, \cdots, r_q$ is downhill from the current point x_0 and when the progress made during the last iteration is beneath some given tolerance so that x_0 appears to the algorithm to be a minimum within the subspace defined by the active constraints.

A FORTRAN program based on the above algorithm has been written and is available from the author. It should be noted that it is currently in test form and might not be suitable for inclusion in a general subroutine library without some modifications. It is hoped however that such a subroutine will be available soon.

9. Numerical results. In this section we present a short summary of results obtained by applying the algorithm of § 8 to some standard test problems. However, since the algorithm's basic approach to minimization is not new, we are not going to present extensive results comparing it to other well-known minimization algorithms. Rather, the results presented here are intended to illustrate that the technique described in the previous sections does indeed enable a good nonderivative conjugate direction algorithm, such as Powell's, to successfully solve linearly constrained problems.

First the algorithm was used to minimize a quadratic function of 3 variables due to Beale (see [8]). The behavior observed was as expected from the theory. From an initial vertex the algorithm moved into the interior of the feasible region where, in a sequence of line searches, it produced 2 conjugate directions. It then encountered another constraint, which was the one and only constraint active at the minimum. It then did one iteration of Powell's algorithm in the manifold defined by the constraint and the number of conjugate directions was restored to two. Since the manifold has dimension two, we were therefore then lead exactly to the constrained minimum of f.

Of course the algorithm was then used to find the minima of some nonquadratic problems. The results of tests on some of the standard problems found in [3] are given in Table 1. The data given there emphasizes the handling of the constraints and shows that the algorithm can also treat linear constraints efficiently when the function is not quadratic.

Acknowledgments. The author would like to thank Dr. Chris Paige for several helpful discussions regarding this work and specifically for his suggestion which constitutes the proof of Theorem 3. Some suggestions of M. J. D. Powell have also been included in the paper, and the author would also like to express his appreciation for these.

REFERENCES

[1] R. P. BRENT (1973), *Algorithms for Minimization Without Derivatives*, Prentice-Hall, Englewood Cliffs, N.J.

[2] A. G. BUCKLEY (1975), *A revised implementation of Goldfarb's minimization algorithm*, Math. Prog., 8, pp. 207–231.

[3] A. R. COLVILLE (1968), *A comparative study of nonlinear programming codes*, Rep. 320-2949, IBM New York Science Center, New York.

[4] R. Fletcher and C. M. Reeves (1964), *Function minimization by conjugate gradients*, Comput. J., 7, pp. 149–154.

[5] R. Fletcher (1971), *Minimizing general functions subject to linear constraints*, U.K.A.E.R.E. Report T.P. 453, Harwell, England.

[6] P. Gill and W. Murray (1971), *Math. Rep.* 97, National Physical Laboratory, Teddington, England.

[7] D. Goldfarb (1969), *Extension of Davidon's variable metric method to maximization under linear inequality and equality constraints*, SIAM J. Appl. Math., 17, pp. 739–764.

[8] J. Kowalik and M. R. Osborne (1968), *Methods for Unconstrained Optimization Problems*, American Elsevier, New York.

[9] M. J. D. Powell (1964), *An efficient method for finding the minimum of a function of several variables without calculating derivatives*, Comput. J., 7, pp. 155–162.

[10] ——— (1972), *Unconstrained minimization algorithms without computation of derivatives*, U.K.A.E.R.E. Rep. T.P. 483, Harwell, England. Also Bolletina U.M.I., 9, pp. 60–69.

[11] J. B. Rosen (1960), *The gradient projection method for nonlinear programming. Part I: Linear constraints*, J. Soc. Indust. Appl. Math., 8, pp. 181–217.

[12] G. W. Stewart (1967), *A modification of Davidon's minimization method to accept difference approximations of derivatives*, J. Assoc. Comput. Mach., 14, pp. 72–83.

[13] W. H. Swann (1964), *Report on the development of a new direct search method of optimization*, ICI Ltd., Central Instrument Lab. Research Note 64/3.

[14] W. I. Zangwill (1967), *Minimizing a function without calculating derivatives*, Comput. J., 10, pp. 263–269.