On the Goldstein – Levitin – Polyak Gradient Projection Method

DIMITRI P. BERTSEKAS

Abstract—**This paper considers some aspects of a gradient projection method proposed by Goldstein [1], Levitin and Polyak [3], and more recently, in a less general context, by McCormick [10]. We propose and analyze some convergent step-size rules to be used in conjunction with the method. These rules are similar in spirit to the efficient Armijo rule for the method of steepest descent and under mild assumptions they have the desirable property that they identify the set of active inequality constraints in a finite number of iterations. As a result the method may be converted towards the end of the process to a conjugate direction, quasi-Newton or Newton's method, and achieve the attendant superlinear convergence rate. As an example we propose some quadratically convergent combinations of the method with Newton's method. Such combined methods appear to be very efficient for large-scale problems with many simple constraints such as those often appearing in optimal control.**

I. Introduction

In 1964 Goldstein [1], [2] proposed a projection method for minimizing a continuously differentiable function $f: H \to R$ over a closed convex subset Q of a Hilbert space H. The method consists of the iteration

$$x_{k+1} = P_Q\left[x_k - a_k \nabla f(x_k)\right], \qquad k=0,1,\cdots \tag{1}$$

where $P_Q(z)$ denotes the unique projection of a vector $z \in H$ on Q, $\nabla f(x_k)$ denotes the gradient of f at the point x_k, and $a_k \geqslant 0$ denotes the step size. The same method was independently proposed by Levitin and Polyak [3] one year later and further discussed in the books by Demyanov and Rubinov [4] and Daniel [12]. Goldstein, Levitin, and Polyak, under the Lipschitz assumption

$$|\nabla f(x) - \nabla f(y)| \leqslant L|x-y|, \qquad \forall x,y \in Q \tag{2}$$

where $|\cdot|$ denotes the norm on H, proved various convergence properties of their method for the case where the step size a_k satisfies

$$0 < \epsilon \leqslant a_k \leqslant \frac{2(1-\epsilon)}{L}, \qquad \forall k \tag{3}$$

with ϵ any scalar with $0 < \epsilon \leqslant 2/(2+L)$. These convergence results are extensions of known results [5], [6] for the method of steepest descent for unconstrained minimization ($Q = H$).

Manuscript received March 28, 1974; revised March 27, 1975 and December 2, 1975. Paper recommended by E. J. Davison, Chairman of the IEEE S-CS Computational Methods and Digital Systems Committee. This work was supported in part by the Joint Services Electronics Program under Contract DAAB-07-72-C-0259, the U.S. Air Force under Grant AFOSR-75-2570, and the National Science Foundation under Grant ENG 74-19332, with the Coordinated Science Laboratory, University of Illinois. A preliminary version of this paper was presented at the 1974 IEEE Decision and Control Conference, Phoenix, AZ.

The author is with the Coordinated Science Laboratory and the Department of Electrical Engineering, University of Illinois, Urbana, IL 61801.

It should be noted, of course, that in order for the method (1) to be effective, the set Q must be such that the projection operation $P_Q(\cdot)$ can be easily carried out. Levitin and Polyak point out several such cases, the most representative of which is when $H = R^n$ and Q implies upper and (or) lower bounds on all the variables of the problem:

$$Q = \left\{x = (x^1, x^2, \cdots, x^n) | \lambda_i \leqslant x^i \leqslant \mu_i,\ i = 1, \cdots, n\right\} \tag{4}$$

$$Q = \left\{x = (x^1, \cdots, x^n) | 0 \leqslant x^i,\ i = 1, \cdots, n\right\}. \tag{5}$$

When Q is given by (4) the iteration (1) takes the form

$$x_{k+1}^i = \begin{cases} \mu_i, & \text{if } x_k^i - a_k \dfrac{\partial f(x_k)}{\partial x^i} \geqslant \mu_i \\ x_k^i - a_k \dfrac{\partial f(x_k)}{\partial x^i}, & \text{if } \lambda_i \leqslant x_k^i - a_k \dfrac{\partial f(x_k)}{\partial x^i} \leqslant \mu_i, \\ & \qquad i = 1, \cdots, n \\ \lambda_i, & \text{if } x_k^i - a_k \dfrac{\partial f(x_k)}{\partial x^i} \leqslant \lambda_i \end{cases}$$

and is very easy to carry out. A similar easily implemented formula holds for the case of the constraint (5). However when Q is a general polyhedron the projection required in the iteration (1) requires the solution of a quadratic programming problem, and when Q is a general convex set things may become even more complicated. Thus the method is effectively limited to problems involving simple constraint sets such as (4), (5), spheres, Cartesian products of spheres, etc.

On the other hand the Goldstein–Levitin–Polyak algorithm has a unique characteristic which makes it extremely attractive for large-scale problems with many simple constraints. Contrary to other algorithms for constrained minimization which maintain feasibility, it proceeds along arcs on the constraint surface rather than along straight line segments. In algorithms such as Zoutendijk's feasible direction methods, Rosen's gradient projection method, the reduced gradient method, and the Frank–Wolfe or conditional gradient algorithm (see, e.g., [7], [8], [9]) the motion along the descent direction stops as soon as a new constraint is encountered. In large-scale problems with many constraints binding at the optimal solution, this fact may result in slow convergence and possibly jamming in a computational environment. Such problems arise typically in optimal control where the optimal control variables are often at the boundary of the constraint set for a large portion of the time interval and sometimes lie entirely on the boundary of the constraint set. As an example consider an optimal control problem of the form

$$\text{minimize } J(u_0,\cdots,u_{N-1}) = G(x_N) + \sum_{i=0}^{N-1} g_i(x_i,u_i)$$

subject to

$$x_{i+1} = f_i(x_i,u_i), \qquad u_i \in U_i,\ i=0,\cdots,N-1$$

where U_i is a constraint set. It is not unusual for such problems to have as many as a thousand control variables, particularly when they result from discretization of continuous time problems. If a large number of optimal control variables lie on the constraint boundary then Rosen's projection method or the reduced gradient method will inevitable take, except for degenerate cases, a very large number of iterations to converge. By contrast, the Goldstein–Levitin–Polyak algorithm does not suffer from this particular difficulty since it does not stop when a new constraint is encountered but rather proceeds along arcs on the boundary of the constraint set. It is to be noted that in optimal control problems the control constraint sets are often simple sets. For example U_i above may be an interval, or the positive orthant, or a closed sphere, etc. In such cases the projection required by the method can be very easily carried out. Iteration (1) takes the convenient form

$$u_i^{k+1} = P_{U_i}\left[u_i^k - \alpha_k \frac{\partial H_i}{\partial u_i}(x_i^k,u_i^k,p_{i+1}^k)\right], \qquad i=0,1,\cdots,N-1$$

where $(u_0^k,u_1^k,\cdots,u_{N-1}^k)$ is the kth control trajectory, $(x_0^k,x_1^k,\cdots,x_{N-1}^k)$ and $(p_1^k,p_2^k,\cdots,p_N^k)$ are the corresponding state and costate trajectories, and H_i denotes the Hamiltonian. The effectiveness of iterations such as the one above for optimal control problems was demonstrated recently in a paper by Quintana and Davison [15].

A weakness of the implementation proposed in [1], [3] is that, since the Lipschitz constant L in (2), (3) is usually unavailable, it is not known *a priori* how small one should take the step size a_k in order for the method to converge, and this in practice may impose serious difficulties. A possible modification of the step-size rule that comes to mind, and is often suggested in the literature, is to start with an initial guess for the step size and then, if the resulting function value is not decreased, reduce the step size successively by multiplication with a constant scalar until a decrease of the value of the function is observed. This step-size rule is quite unreliable since one may easily construct one-dimensional examples where the method converges to nonstationary points of the minimized function, even for the case where $Q=H$ and the problem is unconstrained.

An alternative method for selecting the step size a_k was proposed in 1969 by McCormick [10] for the case where the constraint set is given by (5). Three years later McCormick extended his results of [10] to the case of a general closed convex set in a joint paper with Tapia [19]. McCormick suggested determining a_k by means of the one-dimensional minimization

$$f[P_Q(x_k - a_k\nabla f(x_k))] = \min_{0\leqslant a} f[P_Q(x_k - a\nabla f(x_k))] \qquad (6)$$

and proved convergence of the resulting method without assuming the Lipschitz condition (2). More recently he extended his ideas and gave a reduction method [11] for general linear constraints which is different in spirit from the method of [1] and [3]. Step-size rules of the type proposed here should be useful in conjunction with methods of the type proposed in [11], however this subject has not been investigated. McCormick's proposal is quite impractical for problems of large dimension since the one-dimensional minimization indicated above involves a nondifferentiable function of a, having many points of discontinuity of the derivative. Furthermore this function need not be convex or unimodal even if f is a convex function. As a result it is very difficult to carry out the minimization even approximately. Furthermore this minimization is almost impossible to carry out for problems involving the constraint set (5) which arise in a primal-dual framework via decomposition (see, e.g., [13]). Daniel [12] has also proposed a convergent modification of the projection method (1) which does not require knowledge of the Lipschitz constant L in (2) and is easily implementable. However Daniel's modified method is not a special case of iteration (1) and relinquishes the feature of searching along the boundary of the constraint for an acceptable step size. This feature is very important in the author's opinion for optimal control problems.

The purpose of this paper is to propose and analyze some efficient and easily implementable variations of the gradient projection method (1). In the next section we propose a convergent modification of the well-known Armijo step-size rule for unconstrained minimization to be used in conjunction with iteration (1). We show convergence for the case of a general closed convex constraint set Q under the Lipschitz assumption (2). In Section III we concentrate attention on the special cases where Q has the form (4) or (5), we consider an additional step-size rule of the same type, and we sharpen our results by dispensing with the Lipschitz assumption (2). We furthermore show under mild assumptions that the resulting methods generate sequences which lie on the manifold of binding constraints at the solution after a finite number of iterations. This result reveals that the methods after a finite number of iterations become versions of the steepest descent method restricted to the binding constraint manifold and provides a sharper convergence rate result than the one available in the literature [3, th. 5.1]. Furthermore the result provides a sound basis for converting the method to a superlinearly convergent method such as a conjugate direction method or Newton's method once the set of binding constraints is identified. We discuss two such combined methods which switch automatically to Newton's method towards the end of the algorithmic process, and attain the corresponding superlinear convergence rate. While other Newton-type methods based on gradient projection have been proposed in the literature [12], [16], [20] it is felt that the methods proposed here are more appropriate for optimal control problems. The final section of the paper contains computational results.

We conduct our analysis for the case of a finite-dimensional Euclidean space ($H=R^n$). The norm on R^n is the usual one, $|x|=[\sum_{i=1}^n (x^i)^2]^{1/2}$. The results of Section II have straightforward generalizations to the case where H is an arbitrary Hilbert space.

II. A Generalized Armijo Step-Size Rule

Consider the problem

$$\min_{x\in Q} f(x) \tag{7}$$

where $f: R^n \to R$ is a continuously differentiable function and Q is a closed convex subset of R^n. We say that a point $\bar{x}\in Q$ is a *stationary point* of problem (7) if it satisfies the first-order condition for optimality

$$\nabla f(\bar{x})'(x-\bar{x}) \geqslant 0, \qquad \forall x \in Q \tag{8}$$

where $\nabla f(\bar{x})$ denotes the gradient of f at $\bar{x}$ and prime denotes transposition.

Given any point $x\in Q$, we denote by

$$x(a)=P_Q[x-a\nabla f(x)], \qquad a \geqslant 0 \tag{9}$$

the unique projection of the vector $[x-a\nabla f(x)]$ on Q where $a \geqslant 0$ is a nonnegative scalar parameter. We consider algorithms of the form

$$x_{k+1}=x_k(a_k)=P_Q[x_k-a_k\nabla f(x_k)] \tag{10}$$

where the step size a_k is chosen according to some rule.

Consider now the following rule for selecting the step size a_k in (10).

Generalized Armijo Step-Size Rule

Given a point x_k which is nonstationary, set

$$a_k=\beta^{m_k}s \tag{11}$$

where m_k is the first nonnegative integer m such that

$$f(x_k)-f[x_k(\beta^m s)] \geqslant \sigma\frac{|x_k-x_k(\beta^m s)|^2}{\beta^m s} \tag{12}$$

and σ,β,s are fixed scalars with $0<\sigma<1$, $0<\beta<1$, $0<s$. If x_k is a stationary point, set $a_k=0$.

A generalization of the above step-size rule, which is sometimes useful, is obtained when the constant initial step-size s is replaced by a variable initial step-size s_k. All the results of this and the next section may also be proved for a variable initial step-size s_k provided s_k is bounded above and below by fixed positive numbers.

The mode of operation of the step-size rule is depicted in Fig. 1. It is to be noted that the proposed rule is a direct generalization of the well-known Armijo rule [14], [7] for steepest descent in unconstrained minimization ($Q=R^n$). It is customary to write for the unconstrained case the inequality (12) in the form

$$f(x_k)-f[x_k(\beta^m s)] \geqslant \sigma\beta^m s|\nabla f(x_k)|^2 \tag{13}$$

or sometimes in the form

$$f(x_k)-f[x_k(\beta^m s)] \geqslant \sigma\nabla f(x_k)'[x_k-x_k(\beta^m s)]. \tag{14}$$

When the problem is unconstrained ($Q=R^n$) the inequalities (12), (13), (14) are equivalent. For the constrained case one may consider the natural possibility of replacing the gradient $\nabla f(x_k)$ in (13), (14) by the vector $\lim_{a\to 0^+}(1/a)[x_k-x_k(a)]$ which may be viewed as a projected gradient. When the set Q has special form, such as, for example, the form (4) or (5), the vector $\lim_{a\to 0^+}(1/a)[x_k-x_k(a)]$ is easy to calculate. However one may construct examples to show that when the inequality (13) (with $\lim_{a\to 0^+}(1/a)[x_k-x_k(a)]$ replacing $\nabla f(x_k)$) is used in conjunction with an Armijo step-size rule of the type considered above, then the resulting algorithm may converge to a nonstationary point. On the other hand we show in the next section that inequality (14) when used in conjunction with an Armijo step-size rule leads to a convergent algorithm when Q has the form (4) or (5).

The following proposition is the basic result of this section.

Proposition 1: Assume that, for some $L>0$, we have

$$|\nabla f(x)-\nabla f(y)| \leqslant L|x-y|, \qquad \forall x,y\in Q \tag{15}$$

and let $\{x_k\}$ be a sequence generated by iteration (1) where a_k is chosen according to the generalized Armijo rule of this section. Then every limit point of $\{x_k\}$ is a stationary point.

It is to be noted that the Lipschitz condition assumption (15) is satisfied if f is a twice continuously differentiable function and Q is a bounded set. It is convenient to prove Proposition 1 by proving first two lemmas which will also be useful later on.

Lemma 1: For every $x,y\in Q$ and $a \geqslant 0$ we have

$$a\nabla f(x)'[y-x(a)] \geqslant [x-x(a)]'[y-x(a)]. \tag{16}$$

Proof: By the definition (9) of $x(a)$ as the projection on Q of $x-a\nabla f(x)$, we have

$$[x-a\nabla f(x)-x(a)]'[y-x(a)] \leqslant 0, \qquad \forall x,y\in Q, a\geqslant 0$$

which is equivalent to (16). Q.E.D.

Lemma 2: Assume (15). Then for any point $x\in Q$, and any scalar σ with $0<\sigma<1$, the inequality

$$a\{f(x)-f[x(a)]\} \geqslant \sigma|x-x(a)|^2$$

is satisfied for all a with

$$0 \leqslant a \leqslant \frac{2(1-\sigma)}{L}.$$

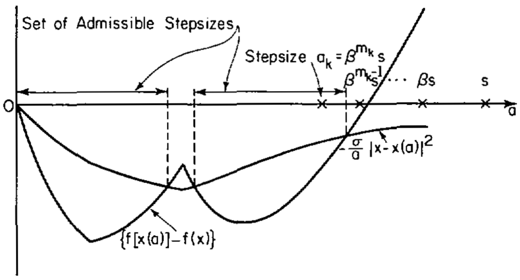

Fig. 1. Generalized Armijo step-size rule.

Proof: We have, for any $a \geq 0$ and $x \in Q$,

$$f[x(a)] = f(x) + \nabla f(x)'[x(a) - x] - \int_0^1 \{\nabla f[x - t(x - x(a))] - \nabla f(x)\}'[x - x(a)]\, dt$$

or, equivalently,

$$a\{f(x) - f[x(a)]\} = a\nabla f(x)'[x - x(a)] - a\int_0^1 \{\nabla f[x - t(x - x(a))] - \nabla f(x)\}'[x(a) - x]\, dt$$

and from (15), (16),

$$a\{f(x) - f[x(a)]\} \geq |x - x(a)|^2 - \frac{aL}{2}|x - x(a)|^2$$

from which the result follows. Q.E.D.

Proof of Proposition 1: If the algorithm stops at a stationary point we are done. So assume otherwise. First notice that by the construction of the step-size a_k, and Lemma 2

$$a_k \geq \min\left\{\frac{2\beta(1-\sigma)}{L}, s\right\} > 0, \qquad \forall k. \tag{17}$$

Hence the algorithm is well defined in the sense that a_k is obtained by a finite number of function evaluations.

Now let $\{x_k\}$ be a sequence generated by the algorithm and let $\{x_k\}_{k \in K}$ be a subsequence converging to a point $\bar{x}$. Since $\{f(x_k)\}$ is a monotonically decreasing sequence we have $\{f(x_k)\} \to f(\bar{x})$ and $\{f(x_k) - f(x_{k+1})\} \to 0$. Hence by (12) we have

$$|x_k - x_{k+1}| \to 0. \tag{18}$$

Now we have, by using Lemma 1, for any vector $y \in Q$ and every $x_k, k \in K$

$$\begin{aligned} \nabla f(x_k)'(x_k - y) &= \nabla f(x_k)'(x_{k+1} - y) + \nabla f(x_k)'(x_k - x_{k+1}) \\ &\leq \frac{1}{a_k}(x_k - x_{k+1})'(x_{k+1} - y) + \nabla f(x_k)'(x_k - x_{k+1}) \\ &\leq |x_k - x_{k+1}|\left|\frac{x_{k+1} - y}{a_k} + \nabla f(x_k)\right|. \end{aligned}$$

Taking limits as $k \to \infty$ and using (17), (18), and the fact that $\{x_{k+1}\}_{k \in K} \to \bar{x}$, we have

$$\nabla f(\bar{x})'(\bar{x} - y) \leq 0, \qquad \forall y \in Q,$$

i.e., the point $\bar{x}$ is stationary. Q.E.D.

We close this section by stating some corollaries, the first of which may be proved by a trivial modification of the proof of Proposition 1.

Corollary 1.1: Under the assumption of Proposition 1 let $\{x_k\}$ be a sequence generated by iteration (1) where a_k is chosen in a way that

$$f(x_k) - f(x_{k+1}) \geq f(x_k) - f[x_k(\bar{a}_k)], \qquad \forall k$$

where $\bar{a}_k$ is the step size corresponding to the generalized Armijo step-size rule of this section. Then every limit point of $\{x_k\}$ is a stationary point.

Corollary 1.2: Under the assumption of Proposition 1 let $\{x_k\}$ be a sequence generated by iteration (1) where a_k is chosen according to the minimization rule

$$f[x_k(a_k)] = \min_{0 \leq a} f[x_k(a)], \qquad \forall k$$

or the limited minimization rule

$$f[x_k(a_k)] = \min_{0 \leq a \leq \bar{a}} f[x_k(a)], \qquad \forall k$$

where $\bar{a} > 0$ is a fixed scalar. Then every limit point of $\{x_k\}$ is a stationary point.

III. The Case of a Simple Constraint Set—Combinations with Newton's Method

We now turn our attention to the special case where the constraint set Q is given by

$$Q = \{x = (x^1, \cdots, x^n) | x^i \geq 0, i = 1, \cdots, n\}. \tag{5}$$

For this case the coordinates of the vector $x(a)$ of (9) are given simply by

$$x^i(a) = \max\left[0, x^i - a\frac{\partial f(x)}{\partial x^i}\right], \qquad i = 1, \cdots, n. \tag{19}$$

All the results of this section can be similarly proved in appropriate form for the case of the constraint set (4) or other similar constraint sets where only some of the variables have upper and/or lower bounds.

Consider now the following rule for selecting the step size a_k in the iteration

$$x_{k+1} = x_k(a_k) \tag{20}$$

where $x_k(a_k)$ is given by (19).

Generalized Armijo Step-Size Rule

Given a point x_k which is nonstationary, set

$$a_k = \beta^{m_k} s \tag{21}$$

where m_k is the first nonnegative integer m such that

$$f(x_k) - f[x_k(\beta^m s)] \geq \sigma \nabla f(x_k)'[x_k - x_k(\beta^m s)] \tag{22}$$

where σ, β, s are fixed scalars with $0 < \sigma < 1$, $0 < \beta < 1$, $0 < s$. If x_k is a stationary point, set $a_k = 0$.

The following proposition shows convergence of the above step-size rule as well as of the rule of the previous section under assumptions which are weaker than those of Proposition 1. The proof of the proposition is quite lengthy and has been relegated to the Appendix. The author communicated the result of the proposition below to Professor A. A. Goldstein, who was able to show its validity for a more general class of convex constraint sets than the one considered in this section (including all polyhedral convex sets) and for a step-size rule similar to the one above [16]. It is as yet unclear whether the result holds for an arbitrary closed convex constraint set Q.

Proposition 2: Let $\{x_k\}$ be a sequence generated by iteration (20), (19) where a_k is chosen according to the generalized Armijo step-size rule of this section or the one of the previous section. Then every limit point of the sequence $\{x_k\}$ is a stationary point for problem (7) with the constraint set Q given by (5).

We note that results analogous to Corollaries 1.1 and 1.2 follow immediately from the above proposition. The analog of Corollary 1.2 for the case of the minimization rule (6) has already been proved by McCormick [10].

Now let $\bar{x}$ be a stationary point for problem (7) with Q having the form (5), which satisfies the following second-order sufficiency conditions for an isolated local minimum.

Assumption 1: $(\partial f(\bar{x})/\partial x^i)>0$ if $\bar{x}^i=0$ and $(\partial f(\bar{x})/\partial x^i)=0$ if $\bar{x}^i>0$, $i=1,\cdots,n$.

Assumption 2: f is twice continuously differentiable within a set of the form $Q\cap\{x||x-\bar{x}|\leqslant\epsilon\}$ ($\epsilon>0$ is some scalar) and

$$m|y-\bar{x}|^2\leqslant(y-\bar{x})'\nabla^2 f(\bar{x})(y-\bar{x})\leqslant M|y-\bar{x}|^2$$

for all $y\in\{z|z^i=0,\ i\in A(\bar{x})\}$, where $A(\bar{x})=\{i|\bar{x}^i=0\}$, $\nabla^2 f(\bar{x})$ denotes the Hessian matrix of f evaluated at $\bar{x}$, and m, M are some positive scalars.

Notice that Assumption 2 implies the Lipschitz condition

$$|\nabla f(x)-\nabla f(y)|\leqslant L|x-y|,\qquad \forall x,y\in Q\cap\{z||z-\bar{x}|\leqslant\epsilon\} \tag{23}$$

where L is a positive scalar depending on the Hessian matrix of f.

We have the following proposition showing that the algorithms considered are attracted by local minima satisfying Assumptions 1 and 2, and furthermore that the generated sequence after a certain index lies on the manifold of binding constraints at $\bar{x}$.

Proposition 3: Let $\bar{x}$ be an isolated local minimum satisfying Assumptions 1 and 2 and let $\{x_k\}$ be any sequence generated by iteration (20), (19) with a_k chosen according to the step-size rule of this section or the one of the previous section. Then there exists a $\delta>0$ such that if for some index l we have $|x_l-\bar{x}|\leqslant\delta$ then the sequence $\{x_k\}$ converges to $\bar{x}$. Furthermore we have for every $k\geqslant l+1$

$$x_k^i=0,\qquad \text{for all } i \text{ such that } \bar{x}^i=0.$$

Again the proof of this proposition is lengthy and has been relegated to the Appendix. The proposition shows that once the algorithm gets close enough to a point $\bar{x}$ satisfying Assumptions 1 and 2, it eventually becomes equivalent to the method of steepest descent on the subspace $\{z|z^i=0,\ i\in A(\bar{x})\}$ where $A(\bar{x})$ is the set of indices i with $\bar{x}^i=0$. It follows that the rate of convergence of the method is governed by the eigenvalue structure of the Hessian $\nabla^2 f(\bar{x})$ over the subspace $\{z|z^i=0,\ i\in A(\bar{x})\}$ rather than over the whole space as suggested by an existing result [3, th. 5.1]. Thus a sharper rate of convergence estimate is obtained than the one existing. Furthermore Proposition 3 suggests the possibility of converting the method to Newton's method (or some other superlinearly convergent method) once the subspace of binding constraints is reached by the algorithm. Some possibilities along these lines are presented below.

Combined Gradient Projection and Newton's Method

Step 1: Select $x_0\in Q$, $\beta\in(0,1)$, $\sigma\in(0,\frac{1}{2})$, $s>0$, $c_1\in(0,1)$, $c_2>0$. (Note: We take c_1,c_2 very small so as to make the corresponding test in Step 4 below as "easy" as possible.)

Step 2: Given x_k if x_k is a stationary point, set $x_{k+1}=x_k$. Else determine $A(x_k)=\{i|x_k^i=0\}$ and $x_k(s)$ as given by (19). If $A[x_k(s)]=A(x_k)$ go to Step 4. Else go to Step 3.

Step 3: Set $x_{k+1}=x_k(a_k)$ with $a_k=\beta^{m_k}s$ where m_k is the smallest nonnegative integer m satisfying

$$f(x_k)-f[x_k(\beta^m s)]\geqslant\sigma\nabla f(x_k)'[x_k-x_k(\beta^m s)]$$

and return to Step 2.

Step 4: Find z_k which renders the function $\frac{1}{2}z'\nabla^2 f(x_k)z+\nabla f(x_k)'z$ stationary subject to the constraint $z^i=0\ \forall i\in A(x_k)$. If there does not exist a unique such point return to Step 3. If

$$(x_k^i+z_k^i)<0,\qquad \text{for some } i\notin A(x_k),$$

or

$$-\sum_{i\notin A(x_k)} z_k^i\frac{\partial f(x_k)}{\partial x^i} < c_1\left\{\sum_{i\notin A(x_k)}(z_k^i)^2\right\}^{\frac{1}{2}}\left\{\sum_{i\notin A(x_k)}\left[\frac{\partial f(x_k)}{\partial x^i}\right]^2\right\}^{\frac{1}{2}},$$

or

$$\sum_{i\notin A(x_k)}(z_k^i)^2<c_2\sum_{i\notin A(x_k)}\left[\frac{\partial f(x_k)}{\partial x^i}\right]^2,$$

determine x_{k+1} as in Step 3 and return to Step 2. Otherwise set $x_{k+1}=x_k+\beta^{m_k}z_k$ where m_k is the first nonnegative integer m satisfying

$$f(x_k)-f(x_k+\beta^m z_k)\geqslant-\sigma\beta^m\nabla f(x_k)'z_k$$

and return to Step 2.

Unfortunately the statement of the algorithm is quite complicated. For this reason some explanations may be helpful. Step 2 determines whether a gradient projection step leads to violation or relaxation of any constraints. If not we switch to Newton's method (Step 4). Step 3 is the basic iteration of the gradient projection method. Step 4 is a Newton step on the subspace of binding constraints combined with the Armijo step-size rule. However the step is foregone (return to Step 3) if either it leads to constraint violation or the tests involving the scalars c_1, c_2 fail. These tests are designed to ensure that the Newton direction z_k is a direction of descent and also that it is of the same order or magnitude as the gradient of the function on the constraint manifold. The Newton direction z_k may be obtained computationally as follows. Consider the $n\times n$ matrix $G(x_k)=[G_{ij}(x_k)]$ and the vector $g(x_k)\in R^n$ with its coordinates denoted by $g_i(x_k)$ where

$$G_{ij}(x_k)=\begin{cases}\dfrac{\partial^2 f(x_k)}{\partial x^i \partial x^j}, & \text{if } i\notin A(x_k), j\notin A(x_k)\\ 1, & \text{if } i\in A(x_k) \text{ and } i=j\\ 0, & \text{otherwise}\end{cases} \tag{24}$$

$$g_i(x_k)=\begin{cases}\dfrac{\partial f(x_k)}{\partial x^i}, & \text{if } i\notin A(x_k)\\ 0, & \text{otherwise.}\end{cases} \tag{25}$$

Then z_k solves the system of equations

$$G(x_k)z_k=g(x_k). \tag{26}$$

These equations can be solved by some form of the Gaussian elimination method based for example on triangular factorization. In a different version of the algorithm the elimination method may be efficiently combined with a stabilization procedure for Newton's method which appropriately modifies the Newton direction (see, e.g., [21]). Such a stabilization procedure can be used in place of the tests involving the scalars c_1 and c_2 in Step 4. In addition in such a version of the algorithm it is not necessary to return to Step 3 if the tests involving c_1, c_2 in Step 4 fail.

The following proposition gives the convergence rate of the above algorithm. Its proof involves similar arguments as those of the proof of Proposition 3 together with standard arguments of quadratic convergence of Newton's method and is left to the reader.

Proposition 4: Let $\{x_k\}$ be any sequence generated by the combined algorithm above. Then if $\bar{x}$ is a local minimum satisfying Assumptions 1 and 2, there exists a $\delta>0$ such that if for some l we have $|x_l-\bar{x}|\leqslant\delta$ the sequence $\{x_k\}$ converges to $\bar{x}$. Furthermore if the scalars c_1 and c_2 are sufficiently small and $\nabla^2 f$ is Lipschitz continuous within $Q\cap\{x\,||x-\bar{x}|\leqslant\epsilon\}$, then the sequence $\{x_k\}$ converges at least quadratically to $\bar{x}$, i.e., there exists some index $\bar{k}$ and a constant $q>0$ such that

$$|x_{k+1}-\bar{x}|\leqslant q|x_k-\bar{x}|^2, \qquad \forall k\geqslant\bar{k}.$$

It is to be noted that the combined gradient projection and Newton's method given above should be viewed as only one out of several possibilities for improving the convergence rate of the method. Combinations with other stabilized versions of Newton's method or with conjugate direction methods are also possible along the lines of the algorithm provided.

An important modification of the combined algorithm given above is obtained by modifying Step 4 so that if $(x_k^i+z_k^i)<0$ for some $i\notin A(x_k)$ (i.e., the Newton step leads to constraint violation), while the other tests of Step 4 are passed, we do not return to Step 3 but rather we set

$$x_{k+1}=P_Q(x_k+\beta^{m_k}z_k) \tag{27}$$

where m_k is the first nonnegative integer m such that

$$f(x_k)-f\left[P_Q(x_k+\beta^k z_k)\right] \geqslant \max\left\{0,\sigma\nabla f(x_k)'\left[x_k-P_Q(x_k+\beta^m z_k)\right]\right\}. \tag{28}$$

One may prove that Proposition 4 still holds when the algorithm is modified as described above. In order to increase reliability and guarantee that every limit point of the sequence generated by the algorithm is a stationary point is may be necessary to introduce some antizigzagging device. There are several possibilities along these lines. A simple scheme is to perform a gradient projection step (Step 3), rather than possibly a Newton step, if x_k satisfies $0<x_k^i<\epsilon$ for some index i, where ϵ is some (small) prespecified scalar. Proposition 4 will still hold if this additional modification is introduced.

When faced with an optimal control problem a combination of the gradient projection method with Newton's method must take into account the standard computational procedure for Newton's method which involves the Riccati equation [17], [18]. One possibility is to test the condition of Step 2 sequentially and backwards in time as the gradient of the Hamiltonian and the gains for the feedback correction of the current control sequence, say $u^k=(u_0^k,u_1^k,\cdots,u_{N-1}^k)$, are computed. Once the condition $A[u^k(s)]=A(u^k)$ of Step 2 is violated, say at time index $\bar{i}$, then one may perform a partial Newton step involving those of the control variables $(u_{i+1}^k,\cdots,u_{N-1}^k)$ which do not lie on the constraint boundary while the remaining variables together with $u_0^k,\cdots,u_i^k$ remain unchanged. Subsequently a gradient projection step involving the whole control trajectory is performed. One can expect that this procedure will eventually identify the control variables which lie on the boundary of the constraint set and under the assumptions of Proposition 4 it will attain the quadratic convergence rate of Newton's method. Notice that since the control variables which lie on the boundary are not taken into account in the computation of the feedback correction, the computational burden associated with the solution of the Riccati equation may be significantly reduced.

IV. Scaling—Computational Aspects and Results

The step-size rules proposed in this paper have been tried on a few moderate-size problems. The steepest descent version of the algorithm was tested and performed efficiently for reasonably well-conditioned problems. Generally speaking the method identified the active constraints in relatively few iterations and subsequently performed in a manner typical of steepest descent in unconstrained minimization. The combination with Newton's method described in the next to last paragraph of the previous section involving iteration (27) and the step-size rule (28) was also tested in one large-scale problem and yielded convergence in very few iterations.

As in the method of steepest descent for unconstrained minimization it may be important to scale the variables of the problem prior to initiating computation. An appropriate scaled version of the gradient projection method with positive definite symmetric scaling matrices $T(x_k)$ (which depend on x_k) is defined by

$$x_{k+1}=x_k(a_k)$$

where $x(a)$ denotes the unique solution of the problem

$$\min_{z\in Q}\left[x_k-aT(x_k)\nabla f(x_k)-z\right]' \cdot T(x_k)^{-1}\left[x_k-aT(x_k)\nabla f(x_K)-z\right],$$

i.e., $x(\alpha)$ is the unique projection of $[x-\alpha T(x_k)\nabla f(x_K)]$ on Q with respect to the norm specified by T^{-1}. The step size a_k is obtained from $a_k=\beta^{m_k}s$ where m_k is the first nonnegative integer m satisfying

$$f(x_k)-f[x_k(\beta^m s)] \geqslant \frac{\sigma[x_k-x_k(\beta^m s)]'T(x_k)^{-1}[x_k-x_k(\beta^m s)]}{\beta^m s}$$

or

$$f(x_k)-f[x_k(\beta^m s)]\geqslant \sigma\nabla f(x_k)'[x_k-x_k(\beta^m s)].$$

By taking $T(x_k)=[\nabla^2 f(x_k)]^{-1}$ above we obtain a Newton-type method. However in general the computation of $x(a)$ requires the solution of a quadratic programming problem and this would be unacceptable for many problems. By contrast for simple constraint sets such as (4) and (5) it is possible to employ diagonal scaling ($T(x_k)$: diagonal) without affecting the convergence properties or the simplicity of the algorithm. Thus for constraint sets such as (4) or (5) the gradient projection iteration can take the form

$$x(a)=P_Q[x-aT(x)\nabla f(x)] \tag{29}$$

where $T(x)$ is diagonal with diagonal elements $T_1(x),\cdots,T_n(x)$ which can be any positive scalars. The corresponding inequality for the Armijo rule takes one of two forms:

$$f(x_k)-f[x_k(\beta^m s)]\geqslant \frac{\sigma}{\beta^m s}\sum_{i=1}^{n} T_i^{-1}(x_k)\left[x_k^i-x_k^i(\beta^m s)\right]^2 \tag{30}$$

$$f(x_k)-f[x_k(\beta^m s)]\geqslant \sigma\nabla f(x_k)'[x_k-x_k(\beta^m s)]. \tag{31}$$

In all our computational experiments involving the steepest descent version of the algorithm we took $T_i(x_k)$ equal to the inverted second derivative $[\partial^2 f(x_k)/\partial x^{i^2}]^{-1}$ evaluated at the current point x_k. For this case the initial step size $s=1$ is a good choice, a fact which substantially contributed to the efficiency of the computation. The choice of scaling factors adopted represents a diagonal approximation of the Hessian matrix and is common in unconstrained minimization.

Example 1: Consider the two-dimensional linear dynamic system

$$\begin{bmatrix}\xi_{i+1}^1\\ \xi_{i+1}^2\end{bmatrix}=\begin{bmatrix}0 & 1\\ -1 & 0\end{bmatrix}\begin{bmatrix}\xi_i^1\\ \xi_i^2\end{bmatrix}+\begin{bmatrix}0\\ 1\end{bmatrix}u_i, \qquad i=0,1,\cdots,N-1$$

where the initial state (ξ_0^1,ξ_0^2) is given. The problem is to find a scalar control sequence $(u_0,\cdots,u_{N-1})$ satisfying $-1\leqslant u_k\leqslant 1$, $k=0,1,\cdots,N-1$ which minimizes

$$J(u_0,\cdots,u_{N-1})=\frac{1}{2}\sum_{i=1}^{N}\left[(\xi_i^1)^2+(\xi_i^2)^2\right].$$

Despite its apparent simplicity this problem is quite ill-conditioned for a large number of stages N. The scaled version (29), (30) of the gradient projection with $s=1$, $\sigma=0.1$, $\beta=0.1$, was employed to solve this problem for a variety of initial states and number of stages. The initial point was $u^0=(0,\cdots,0)$. For $(\xi_0^1,\xi_0^2)=(10^3,10^3)$ and $N=10$, $N=10^2$, $N=10^3$, the optimal control sequence lies entirely on the boundary of the constraint region and was obtained in a single iteration. The same occurred for $(\xi_0^1,\xi_0^2)=(10^2,10^2)$ and $N=10$, $N=100$. For $(\xi_0^1,\xi_0^2)=(40,40)$ and $N=100$, the optimal control sequence does not lie entirely on the boundary. The number of active constraints is 78 and the gradient projection method identified these constraints in 11 iterations. Subsequently the method was performing (scaled) steepest descent on the space of the remaining 22 variables and after an additional 11 iterations it attained the optimal value to within a sufficient degree of accuracy that roundoff error (in single precision) became significant and termination occurred. Notice here that, due to the special nature of the problem, the optimal solution is easy to obtain by essentially analytical means once the set of active constraints is identified.

Example 2: The second example relates to a problem of scheduling water release from a reservoir subject to upper and lower bounds on the total water volume in the reservoir. The volume x^i at period i is governed by the equation

$$x^{i+1}=x^i+d_i-u_i, \qquad i=0,1,\cdots,N-1$$

where u_i is water released, and d_i is a known inflow

$$d_i=6+10\sin\left[\frac{2\pi(i+1)}{N+1}\right], \qquad i=0,1,\cdots,N-1.$$

The constraints are

$$x^0=x^N=8, \qquad 2\leqslant x^i\leqslant 8, \qquad i=1,\cdots,N-1$$

and the two cost functionals that were minimized are given by

$$J(u_0,\cdots,u_{N-1}) = \sum_{i=0}^{N-1} \exp[-0.5u_i]$$
$$= \sum_{i=0}^{N-1} \exp\left[0.5(x^{i+1}-x^i-d_i)\right] \quad (32)$$

$$J(u_0,\cdots,u_{N-1}) = \sum_{i=0}^{N-1} (-42u_i+u_i^2)$$
$$= \sum_{i=0}^{N-1} \left[42(x^{i+1}-x^i-d_i)+(x^{i+1}-x^i-d_i)^2\right]. \quad (33)$$

The scaled version (29), (30) of the gradient projection method with $s=1$, $\sigma=0.1$, $\beta=0.1$ was used to solve the problem for the initial point $x^i=5$, $i=1,\cdots,N-1$, the two cost functionals (32), (33), and $N=12$, $N=52$, $N=104$. The results are summarized in Table I. A star in the k-column indicates the iteration where all active constraints at the solution were identified. After that iteration the method was equivalent to steepest descent on the subspace of the variables corresponding to inactive constraints. For the case where $N=365$, the method was making slow progress and the computation was not carried to completion.

Subsequently we tried a combination of steepest descent and Newton's method for solving the problem for the case of the quadratic cost functional (33). In this combination at the current point x_k the set of indices corresponding to active constraints $A(x_k)$ was first determined. Subsequently the point $x_k(1)$ was determined via (29) with each scaling factor $T_i(x_k)$ equal to the corresponding inverse second derivative $[\partial^2 f(x_k)/\partial x^{i^2}]^{-1}$. If $A(x_k) \neq A[x_k(1)]$ the next point x_{k+1} was determined via (29), (30) with $s=1$, $\sigma=0.1$, $\beta=0.1$. If $A(x_k) = A[x_k(1)]$, the Newton direction z_k on the space of binding constraints was computed and x_{k+1} was determined via (27), (28) with $\sigma=0.1$, $\beta=0.1$. The results of the computation for the starting point $x^i=5$, $i=1,\cdots,N-1$ are shown in Table II. An (S) in the iteration number column indicates a steepest descent step while an (N) indicates a Newton step. As can be seen from the table, the method converged (to the exact minimum) in very few iterations even for large dimensions.

V. Conclusions

This paper has provided proper extensions of the Armijo step-size rule for use with the gradient projection method for minimization over a closed convex set. The rules are easy to implement and, at least for simple constraints, they lead to convergence under weaker assumptions than those of Goldstein, Levitin, and Polyak. Furthermore, under sufficiency assumptions, they allow the combination of the gradient projection method with higher order methods. It is to be noted that other step-size rules with similar properties may be constructed. For

TABLE I
Computational Results—Example 2 (Steepest Descent Version)

N=12 (# of act. constraints=5)				N=52 (# of act. constraints=33)				N=104 (# of act. constraints=71)			
k	C.F. (32)	k	C.F. (33)	k	C.F. (32)	k	C.F. (33)	k	C.F. (32)	k	C.F. (33)
0	19.3472	0	-1868.23	0	72.1201	0	-8549.80	0	142.555	0	-17188.8
1	15.4261	1	-1941.98	1	69.9509	1	-8582.00	1	141.381	1	-17210.3
3*	12.9163	3*	-1974.43	5	62.4588	5	-8687.19	9	132.960	9	-17332.3
5	12.6992	5	-1975.37	9	57.8171	9	-8725.74	17	127.106	17	-17385.0
7	12.6556	7	-1975.57	13	56.7295	13	-8730.40	25	125.318	25	-17391.7
9	12.6448	9	-1975.63	18*	56.5678	18*	-8730.99	33	124.866	33	-17393.2
11	12.6420	11	-1975.64	22	56.5606	20	-8731.01	40*	124.779	40*	-17393.4
13	12.6414	13	-1975.64	26	56.5602	22	-8731.02	45	124.764	45	-17393.5
15	12.6412			30	56.5602	24	-8731.02	50	124.760	50	-17393.5
17	12.6411							55	124.758		
19	12.6411							60	124.758		

TABLE II
Computational Results—Example 2 (Combined Steepest Descent and Newton's Method)

N = 12		N = 52		N = 104		N = 365		N = 365 (cont.)	
k	Value of C.F. (33)	k	Value of C.F. (33)	k	Value of C.F. (33)	k	Value of C.F. (33)	k	Value of C.F. (33)
0	-1868.23	0	-8549.81	0	-17188.8	0	-60519.9	14(S)	-60749.9
1(N)	-1942.00	1(N)	-8582.00	1(N)	-17214.0	1(N)	-60540.0	15(S)	-60749.9
2(S)	-1963.80	2(S)	-8597.06	2(S)	-17222.5	2(S)	-60542.6	16(N)	-60750.4
3(S)	-1969.81	3(S)	-8610.23	3(S)	-17230.5	3(S)	-60545.2	17(S)	-60750.4
4(N)	-1975.65	4(N)	-8690.08	4(N)	-17331.0	4(N)	-60666.6	18(S)	-60750.4
		5(S)	-8724.81	5(S)	-17378.1	5(S)	-60719.2	19(N)	-60750.5
		6(S)	-8729.12	6(S)	-17386.5	6(S)	-60731.3	20(S)	-60750.5
		7(S)	-8730.27	7(S)	-17389.6	7(S)	-60736.9	21(N)	-60750.5
		8(N)	-8731.03	8(S)	-17391.1	8(S)	-60740.2	22(S)	-60750.5
				9(N)	-17393.4	9(S)	-60742.3	23(N)	-60750.5
				10(S)	-17393.5	10(S)	-60743.8		
				11(N)	-17393.6	11(S)	-60744.9		
				12(S)	-17393.6	12(N)	-60749.4		
				13(N)	-17393.6	13(S)	-60749.7		

example, it is possible to prove similar convergence results for an analog of a step-size rule due to Goldstein (see [16]), as well as other rules along the same lines (see, e.g., [12]).

Both the analysis and the computational results suggest that the gradient projection method, particularly when combined with Newton's method, can be extremely effective in solving multidimensional problems with many simple constraints such as lower and/or upper bounds on the variables. The computational results suggest also that quadratic programming problems with many simple constraints can be handled very efficiently by the methods of this paper.

Appendix

Proof of Proposition 2

Let us first define for every $x \in Q, i=1,\cdots,n$,

$$a_i(x) = \inf\left\{a>0 \,|\, x^i - a\frac{\partial f(x)}{\partial x^i} \leqslant 0\right\} \quad (A1)$$

where $a_i(x)=\infty$ if the set above is empty. Then we have for every $x\in Q$ the equations

$$x^i(a)=x^i-\min[a,a_i(x)]\frac{\partial f(x)}{\partial x^i},\qquad i=1,\cdots,n \quad \text{(A2)}$$

$$\nabla f(x)'[x-x(a)]=\sum_{i=1}^n \min[a,a_i(x)]\left[\frac{\partial f(x)}{\partial x^i}\right]^2. \quad \text{(A3)}$$

Consider first the case of sequences $\{x_k\}$ generated by (19), (20) with the step-size rule of Section III. Now, if x_k is a stationary point for some k we are done. So assume that each point $x_k, k=0,1,\cdots$, is nonstationary. Let $\{x_k\}_{k\in K}$ be a subsequence of $\{x_k\}$ converging to a point $\bar{x}$. It follows from (21) that

$$\nabla f(x_k)'[x_k-x_k(a_k)]\to 0. \quad \text{(A4)}$$

Assume now that $\bar{x}$ is a nonstationary point. Then there exists an index j such that

$$\bar{x}^j=0 \quad\text{and}\quad \frac{\partial f(\bar{x})}{\partial x^j}<0$$

or

$$\bar{x}^j>0 \quad\text{and}\quad \frac{\partial f(\bar{x})}{\partial x^j}\neq 0. \quad \text{(A5)}$$

In either case above we have that there exists a positive integer $\bar{k}$ and positive scalars $\bar{a}$ and θ such that

$$a_j(x_k)\geqslant \bar{a}>0,\quad \left[\frac{\partial f(x_k)}{\partial x^j}\right]^2\geqslant\bar{\theta}>0,\qquad \text{for all } k\in K,\ k\geqslant\bar{k}. \quad \text{(A6)}$$

Using (A3) and (A6) we have

$$\nabla f(x_k)'[x_k-x_k(a_k)]\geqslant \min\left[\inf_{k\in K}a_k,\bar{a}\right]\bar{\theta}\geqslant 0,\qquad \forall k\in K,\ k\geqslant\bar{k}$$

and taking limits and using (A4) we obtain

$$\inf_{k\in K} a_k=0. \quad \text{(A7)}$$

It follows that there exists a subsequence $\{x_k,a_k\}_{k\in K'}$, $K'\subset K$ such that

$$\{x_k,a_k\}_{k\in K'}\to(\bar{x},0). \quad \text{(A8)}$$

From the above relation and the form of the generalized Armijo rule, it follows that for some index $\bar{k}'$ we have

$$f(x_k)-f\left[x_k(\beta^{-1}a_k)\right]<\sigma\nabla f(x_k)'\left[x_k-x_k(\beta^{-1}a_k)\right],\qquad \forall k\in K',\ k\geqslant\bar{k}', \quad \text{(A9)}$$

i.e., the test (22) fails at least once for $k\in K', k\geqslant\bar{k}'$. Define the index set

$$\bar{I}=\left\{i\,|\,\bar{x}^i=0 \text{ and } \frac{\partial f(\bar{x})}{\partial \bar{x}^i}<0, \text{ or } \bar{x}^i>0 \text{ and } \frac{\partial f(\bar{x})}{\partial \bar{x}^i}\neq 0,\qquad i=1,\cdots,n\right\}. \quad \text{(A10)}$$

Since $\bar{x}$ is nonstationary, the set $\bar{I}$ is nonempty. Furthermore since $\{a_k\}_{k\in K'}\to 0$ we have for some positive integer $\tilde{k}$ and some positive scalars $\tilde{a},\tilde{\theta}$

$$a_i(x_k)\geqslant\tilde{a}\geqslant\beta^{-1}a_k>0,\qquad \left[\frac{\partial f(x_k)}{\partial x^i}\right]^2\geqslant\tilde{\theta}>0,\qquad \forall k\in K',\ k\geqslant\tilde{k},\ i\in\bar{I} \quad \text{(A11)}$$

where $a_i(x_k)$ is defined by (A1). Now by the mean value theorem we have

$$f(x_k)-f\left[x_k(\beta^{-1}a_k)\right]=\nabla f(x_k)'\left[x_k-x_k(\beta^{-1}a_k)\right]+\left[\nabla f(\xi_k)-\nabla f(x_k)\right]'\left[x_k-x_k(\beta^{-1}a_k)\right] \quad \text{(A12)}$$

where ξ_k is some vector on the line segment joining x_k and $x_k(\beta^{-1}a_k)$. Hence, by (A9), for all $k\in K', k\geqslant\bar{k}'$,

$$0\leqslant(1-\sigma)\nabla f(x_k)'\frac{\left[x_k-x_k(\beta^{-1}a_k)\right]}{|x_k-x_k(\beta^{-1}a_k)|}<\left[\nabla f(x_k)-\nabla f(\xi_k)\right]\frac{\left[x_k-x_k(\beta^{-1}a_k)\right]}{|x_k-x_k(\beta^{-1}a_k)|}. \quad \text{(A13)}$$

Since $\{a_k\}_{k\in K'}\to 0$, the right-hand side above tends to zero and as a result

$$\nabla f(x_k)'\frac{\left[x_k-x_k(\beta^{-1}a_k)\right]}{|x_k-x_k(\beta^{-1}a_k)|}\to 0,\qquad \text{as } k\to\infty,\ k\in K'. \quad \text{(A14)}$$

Equivalently we have, using (A3),

$$\frac{\sum_{i=1}^n \min\left[\beta^{-1}a_k,a_i(x_k)\right]\left[\frac{\partial f(x_k)}{\partial x^i}\right]^2}{\left(\sum_{i=1}^n\left(\min\left[\beta^{-1}a_k,a_i(x_k)\right]\right)^2\left[\frac{\partial f(x_k)}{\partial x^i}\right]^2\right)^{\frac{1}{2}}}\to 0,\qquad \text{as } k\to\infty,\ k\in K' \quad \text{(A15)}$$

from which we obtain

$$\frac{\sum_{i=1}^n \min\left[1,\left[\frac{a_i(x_k)}{\beta^{-1}a_k}\right]\right]\left[\frac{\partial f(x_k)}{\partial x^i}\right]^2}{|\nabla f(x_k)|}\to 0,\qquad \text{as } k\to\infty,\ k\in K' \quad \text{(A16)}$$

and, using (A11),

$$\frac{\sum_{i\in\bar{I}}\left[\frac{\partial f(x_k)}{\partial x^i}\right]^2}{|\nabla f(x_k)|}\to 0,\qquad \text{as } k\to\infty,\ k\in K'. \quad \text{(A17)}$$

Since

$$|\nabla f(x_k)|=\left[\sum_{i\in\bar{I}}\left[\frac{\partial f(x_k)}{\partial x^i}\right]^2+\sum_{i\notin\bar{I}}\left[\frac{\partial f(x_k)}{\partial x^i}\right]^2\right]^{\frac{1}{2}} \quad \text{(A18)}$$

and $\Sigma_{i\notin \bar{I}}[\partial f(x_k)/\partial x^i]^2$ is bounded, it follows from (A17) that

$$\sum_{i\in \bar{I}}\left[\frac{\partial f(x_k)}{\partial x^i}\right]^2 \to 0, \qquad \text{as } k\to\infty,\ k\in K' \tag{A19}$$

or equivalently

$$\sum_{i\in \bar{I}}\left[\frac{\partial f(\bar{x})}{\partial x^i}\right]^2 = 0 \tag{A20}$$

which contradicts the assumption that $\bar{x}$ is a nonstationary point. Hence $\bar{x}$ must be stationary and Proposition 2 is proved for the case of the step-size rule of Section III. The proof of Proposition 2 for the case of the step-size rule of Section II is similar. If $\{x_k\}_{k\in K}\to\bar{x}$ and $\bar{x}$ is not stationary, then (A5), (A6) hold and in addition

$$\frac{1}{a_k}|x_k - x_k(a_k)|^2 \to 0.$$

Since

$$\frac{1}{a_k}|x_k - x(a_k)|^2 = \frac{1}{a_k}\sum_{i=1}^{n}\min\left[a_k^2, a_i(x_k)^2\right]\left[\frac{\partial f(x_k)}{\partial x^i}\right]^2,$$

we obtain again (A7) as well as the analog of (A9) which now takes the form

$$f(x_k) - f\left[x_k(\beta^{-1}a_k)\right] < \frac{\sigma}{\beta^{-1}a_k}|x_k - x_k(\beta^{-1}a_k)|^2.$$

By Lemma 1 we have

$$\frac{1}{\beta^{-1}a_k}|x_k - x_k(\beta^{-1}a_k)|^2 \leqslant \nabla f(x_k)'\left[x_k - x_k(\beta^{-1}a_k)\right].$$

Combining the above two inequalities we obtain (A9) and from this point the proof proceeds exactly as for the case of the step-size rule of Section II. Q.E.D.

Proof of Proposition 3

First we observe that by (A1) and the continuity of $\nabla f(x)$ there exists a sphere $S(\bar{x};\delta_1)$ such that

$$\frac{\partial f(x)}{\partial x^i} > 0, \qquad \forall i\in A(\bar{x}) = \{i|\bar{x}^i = 0\},\ x\in Q\cap S(\bar{x};\delta_1).$$

It follows from known results on the method of steepest descent and Assumption 2 that there exists a scalar $\delta_2 > 0$ such that if a point x_m generated by the algorithm satisfies

$$x_m \in Q\cap S(\bar{x};\delta_2)\cap\{z|z^i = 0,\ i\in A(\bar{x})\}$$

then $x_k\to\bar{x}$ and $x_k^i = 0$ for all $i\in A(\bar{x}), k\geqslant m$. Thus the proposition will be proved if we can demonstrate the existence of a scalar $\delta > 0$ such that if a point x_l generated by the algorithm satisfies $|x_l - \bar{x}| \leqslant \delta$, then

$$x_{l+1}\in Q\cap S(\bar{x};\delta_2)\cap\{z|z^i = 0,\ i\in A(\bar{x})\}. \tag{A21}$$

Indeed by the Lipschitz condition (23), Lemma 2, and $a_l \leqslant s$, we have that the step size a_l is bounded above and below by positive numbers provided $|x_l - \bar{x}| \leqslant \epsilon'$ where ϵ' is a sufficiently small positive number to guarantee that $|x_{l+1} - \bar{x}| \leqslant \epsilon$. It follows, in view of Assumption 1, that if the coordinates $x_l^i, i\in A(\bar{x})$ are sufficiently small nonnegative numbers then $x_l^i(a_l) = x_{l+1}^i = 0$ for all $i\in A(\bar{x})$, and if the coordinates x_l^i, $i\notin A(\bar{x})$ are sufficiently close to $\bar{x}^i$, $i\notin A(\bar{x})$ then we will have $|x_{l+1} - \bar{x}| \leqslant \delta_2$ and x_{l+1} will satisfy (A21). Hence the existence of a scalar $\delta > 0$ such that (A21) is satisfied for every $x_l\in Q\cap S(\bar{x};\delta)$ is clear and the proposition is proved. Q.E.D.

References

[1] A. A. Goldstein, "Convex programming in Hilbert space," *Bull. Amer. Math. Soc.*, vol. 70, no. 5, pp. 709–710, 1964.

[2] ——, *Constructive Real Analysis*. New York: Harper & Row, 1967.

[3] E. S. Levitin and B. T. Polyak, "Constrained minimization problems," *USSR Comput. Math. Math. Phys.*, vol. 6, pp. 1–50, 1966 (English transl. of paper in *Zh. Vychisl. Mat. i Mat. Fiz.*, vol. 6, pp. 787–823, 1965).

[4] V. F. Demyanov and A. M. Rubinov, *Approximate Methods in Optimization Problems*. New York: American Elsevier, 1971.

[5] B. T. Polyak, "Gradient methods for the minimization of functionals," *USSR J. Comput. Math. Math. Phys.*, vol. 3, pp. 643–653, 1963.

[6] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Academic, 1969.

[7] E. Polak, *Computational Methods in Optimization: A Unified Approach*. New York: Academic, 1971.

[8] D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1973.

[9] W. I. Zangwill, *Nonlinear Programming: A Unified Approach*. Englewood Cliffs, NJ: Prentice-Hall, 1969.

[10] G. P. McCormick, "Anti-zig-zagging by bending," *Management Sci.*, vol. 15, pp. 315–319, 1969.

[11] ——, "The variable reduction method for nonlinear programming," *Management Sci.*, vol. 17, no. 3, pp. 146–160, 1970.

[12] J. W. Daniel, *The Approximate Minimization of Functionals*. Englewood Cliffs, NJ: Prentice-Hall, 1971.

[13] L. S. Lasdon, *Optimization Theory for Large Systems*. New York: Macmillan, 1970.

[14] L. Armijo, "Minimization of functions having continuous partial derivatives," *Pac. J. Math.*, vol. 16, pp. 1–3, 1966.

[15] V. H. Quintana and E. J. Davison, "Clipping-off gradient algorithms to compute optimal controls with constrained magnitude," *Int. J. Contr.*, vol. 20, pp. 243–255, 1974.

[16] A. A. Goldstein, "On gradient projection," in *Proc. 12th Ann. Allerton Conf. Circuit and System Theory*, Univ. of Illinois, Oct. 1974, pp. 38–40.

[17] S. K. Mitter, "Successive approximation methods for the solution of optimal control problems," *Automatica*, vol. 3, 1966.

[18] P. Dyer and S. R. McReynolds, *The Computation and Theory of Optimal Control*. New York: Academic, 1970.

[19] G. P. McCormick and R. A. Tapia, "The gradient projection method under mild differentiability conditions," *SIAM J. Contr.*, vol. 10, pp. 93–98, 1972.

[20] A. A. Goldstein and R. A. Tapia, "Kantorovich estimates for some convex programs," *SIAM J. Contr.*, vol. 11, pp. 375–377, 1973.

[21] W. Murray, "Second derivative methods," in *Numerical Methods for Unconstrained Optimization*, W. Murray, Ed. New York: Academic, 1972, pp. 57–71.

Dimitri P. Bertsekas was born in Athens, Greece, in 1942. He received the Mechanical and Electrical Engineering Diploma from the National Technical University of Athens, Athens, Greece, in 1965, the M.S.E.E. degree from George Washington University, Washington, DC, in 1969, and the Ph.D. degree in system science from the Massachusetts Institute of Technology, Cambridge, in 1971.

From 1966 to 1967 he performed research at the National Technical University of Athens, Athens, Greece, and from 1967 to 1969 he was with the U.S. Army Research Laboratories, Fort Belvoir, VA. In the summer of 1971 he

worked at Systems Control, Inc., Palo Alto, CA. From September 1971 until December 1973 he was with the faculty of the Engineering-Economic Systems Department of Stanford University, Stanford, CA. In January 1974 he joined the Coordinated Science Laboratory and the Department of Electrical Engineering at the University of Illinois, Urbana, where he is currently an Associate Professor. He has done research in estimation and control of uncertain systems, dynamic programming, nonlinear programming, and economic analysis and he is the author of the graduate text *Dynamic Programming and Stochastic Control* (New York: Academic, 1976).

An Algorithm for Optimization Problems with Functional Inequality Constraints

ELIJAH POLAK, SENIOR MEMBER, IEEE, AND DAVID Q. MAYNE, MEMBER, IEEE

Abstract—**This paper presents an algorithm for minimizing an objective function subject to conventional inequality constraints as well as to inequality constraints of the functional type: $\max_{\omega\in\Omega}\phi(z,\omega)\leqslant 0$, where Ω is a closed interval in R, and $z\in R^n$ is the parameter vector to be optimized. The algorithm is motivated by a standard earthquake engineering problem and the problem of designing linear multivariable systems. The stability condition (Nyquist criterion) and disturbance suppression condition for such systems are easily expressed as a functional inequality constraint.**

I. Introduction

THIS PAPER presents an algorithm for solving problems of the form

$$\min\left\{f^0(z)|f^j(z)\leqslant 0, j=1,\cdots,m;\ g^j(z)\leqslant 0, j=1,\cdots,p\right\}$$

where $z\in R^n$ is the parameter vector to be optimized, $f^0:R^n\to R$ is the criterion, $g^j:R^n\to R$, $j=1,\cdots,p$, are conventional inequality constraints, and $f^j:R^n\to R$, $j=1,\cdots,m$, are inequality constraints of the *functional* type, i.e.,

$$f^j(z)\triangleq\max_{\omega\in\Omega}\phi^j(z,\omega),\qquad j=1,\cdots,m \tag{1}$$

where Ω is a compact interval in R. The algorithm is motivated by problems arising in the design of earthquake resistant buildings [10], and in designing controllers for linear multivariable systems using frequency response techniques. For the former problem, the cost f^0 is, for example, the weight of the building, and a typical constraint is the maximum deviation between floors during an earthquake.

Manuscript received September 20, 1974; revised April 28, 1975 and November 4, 1975. Paper recommended by E. J. Davison, Chairman of the IEEE S-CS Computational Methods and Digital Systems Committee. This work was supported by the U.S. Army Research Office under Durham Contract DAHC04-73-C-0025 and by the National Science Foundation under Grant GI-36387.

E. Polak is with the Department of Electrical Engineering and Computer Sciences and the Electronics Research Laboratory, University of California, Berkeley, CA 94720.

D. Q. Mayne is with the Department of Computing and Control, Imperial College, London, England.

For the latter problem we shall only discuss the formulation of constraints of the form (1). The cost would normally be an error function in the time or the frequency domain, or a combination of both, with possibly an energy term added on. Conventional inequality constraints arise on gains, torques, rudder angles, etc. Thus, to return to (1), it is well known [3]–[5] that stability and performance can be expressed in terms of the matrix return difference $T(s,z)=I+G(s,z)$, where $G(s,z)$ is the matrix loop gain and z is the vector of controller parameters. Both T and G are $m\times m$ matrices of rational functions, m being the number of outputs. If the open-loop system is stable, then the closed-loop system is stable if the locus $\det(T(i\Omega',z))\triangleq\{\det(T(i\omega,z)):\ \omega\in\Omega'\triangleq[0,\infty)\}$ does not pass through or encircle the origin in the complex plane C. To achieve a degree of "relative" stability, a subset B of the complex plane can be specified so that $\det(T(i\omega,z))\not\in B$, for all $\omega\in\Omega'$, implies that the locus $\det(T(i\omega',z))$ does not encircle the origin and, in addition, does not approach "too closely" the origin (Fig. 1). If, instead, the locus $\det(T(\alpha+i\Omega',z))\triangleq\{\det(T(\alpha+i\omega,z));\ \omega\in\Omega'\}$, $\alpha<0$, is considered, and B is chosen, as in Fig. 2, simply to prevent encirclement of the origin, then $\det(T(\alpha+i\omega,z))\not\in B$, for all $\omega\in\Omega'$, implies that all the closed-loop poles have real parts less than or equal to α, thus, automatically ensuring a degree of relative stability. B can be specified in terms of several functions: $\theta^i:C\to R$, $i\in I$ as follows:

$$B\triangleq\left\{s\in C|\theta^i(s)>0, i\in I\right\};$$

then the closed-loop system is stable, if its parameters z satisfy, for all $i\in I$:

$$f^i(z)\leqslant 0 \tag{2a}$$

where

$$f^i(z)\triangleq\max_{\omega\in\Omega'}\phi^i(z,\omega) \tag{2b}$$