AD 672928

CONVERGENCE CONDITIONS FOR NONLINEAR PROGRAMMING ALGORITHMS

By

W. I. Zangwill

Working Paper No. 197
Revised March, 1968

Center for Research in Management Science
University of California
Berkeley

JUN 7 1968

This research was supported (in part) by a grant from the Office of Naval Research Grant No. ONR-NR-047-069 to the University of California, and administered through the Center for Research in Management Science. Reproduction in whole or part is permitted for any purpose of the United States Government.

NONR-222(77)

This document has been approved for public release and sale; its distribution is unlimited.

This [redacted] and is not to [redacted].

Reproduced by the
CLEARINGHOUSE
for Federal Scientific & Technical
Information Springfield Va 22151

24

CONVERGENCE CONDITIONS FOR NONLINEAR PROGRAMMING ALGORITHMS

W. I. Zangwill

School of Business Administration
University of California
Berkeley, California

Abstract

Conditions which are necessary and sufficient for convergence of a nonlinear programming algorithm are stated. It is also shown that the convergence conditions can be easily applied to most programming algorithms. As examples, algorithms by Arrow, Hurwicz and Uzawa; Cauchy; Frank and Wolfe; and Newton-Raphson are proven to converge by direct application of the convergence conditions. Also the Topkis-Veinott convergence conditions for feasible direction algorithms are shown to be a special case of the conditions stated in this paper.

Background and Summary

Nearly twenty years ago F. John [7], and Kuhn and Tucker [10], in brilliant papers, discussed when a given point was optimal for a nonlinear programming problem. Under certain assumptions they gave necessary and sufficient conditions for a point to be optimal. From a practical concave programming orientation the question, "Is a given point optimal?" was now settled. Moreover, their conditions prompted exploration of a broader problem, viz., given a point which is not optimal, how can an optimal point be located. In effect, F. John and Kuhn and Tucker answered the static question of knowing when a given point was optimal, but did not resolve the dynamic question of how to move from a point which is not optimal to an optimal one. As partial answers to the latter question numerous nonlinear programming algorithms have been developed. One of the earliest and best known of these, the Simplex Method [4], actually predates the F. John, and Kuhn-Tucker conditions.

Even a casual glance at the literature reveals the plethora of algorithmic techniques; each one seemingly different from the next, each having its own advantages and disadvantages. As is well known, it is often extremely difficult to prove that an algorithm converges. In fact, only a small percentage of all suggested procedures have ever been proven to converge. And even some which at first were thought to converge were later found to have incorrect or incomplete proofs. Furthermore, each algorithm seemed to have its own unique and different proof.

It is the purpose of this paper to explore the similarities among algorithms. Do the Simplex Method, the Newton-Raphson Method, and in fact, all programming algorithms have a common essence? And if so, do there exist features which insure algorithmic convergence? In an important paper Topkis and Veinott [12], have studied these questions for the class of feasible direction algorithms. It will be shown that their conditions are subsumed by the conditions presented in this paper. But in addition, this paper will not only give conditions that are sufficient to insure convergence, but also will pose conditions that convergent algorithms necessarily satisfy.

The practical impact of these conditions will be illustrated by using them to prove convergence of several well-known algorithms. For example, we correct an error in Uzawa's modification of the Arrow-Hurwicz algorithm. Furthermore, in most cases the convergence proofs are considerably simpler than the original proofs. Even when there is no obvious simplification or improvement, the convergence conditions provide a unified and straightforward method for proving convergence.

The Algorithm as an Iterative Procedure

Consider the general nonlinear programming problem, called problem (P).

$$(P)\quad \begin{array}{ll} (1) & \text{maximize } f(x) \\ (2) & \text{subject to } g_i(x) \geq 0 \qquad i=1,\cdots,m, \end{array}$$

where all functions are real-valued and $x \in E^n$, n-dimensional Euclidean space. It is assumed throughout the paper that f is continuous. Define $F \subset E^n$ as the set of all x which satisfy (2). The set F is the feasible set. Any point $x \in F$ that maximizes f is said to be an optimal point for (P). We assume $F \neq \phi$, where ϕ is the null set.

Our goal is to analyze algorithmic procedures for solving problem (P). For definiteness assume the algorithms are for digital computers, and therefore, the algorithms generate a discrete sequence of points. Furthermore, the algorithm need not operate directly upon the points x, but say on related points z. We may thus view an algorithm as a rather sophisticated iterative procedure, that given a point z^k either stops or generates a successor z^{k+1}. For generality assume that the points z on which the procedure operates need not be in E^n. Merely require that they be defined on a given metric space (V, ρ). Often the metric space will in fact be E^n with the usual metric.

Now examine the iterative operation itself. Given a point z^k the procedure yields a point z^{k+1}. It may be possible to actually define a function $A: V \to V$ such that $z^{k+1} = A(z^k)$. The function then defines the iterative procedure. Unfortunately in many cases such a function would not be well defined as there may not be a unique value $A(z)$ for a given z. As an example, consider the Simplex Method and suppose the point z has just been generated. The point z is a basic feasible solution of the constraining linear inequalities. Now assume that the next point y, also a basic solution, is to be generated. The point y, called a successor point,

may not be well defined as there might be a tie in the choice of the variable to enter the basis. That is, there are situations in which several possible y can conceivably be generated from z. Similar ambiguity about the successor to a point z arises in other algorithms. We are therefore forced to consider procedures that may generate a y in some set. The set being the set of all possible successor points that the iterative operation could conceivably generate from a particular z.

It should also be observed that the procedure might depend upon the number of iterations k already taken place. A procedure which does not depend upon k is said to be autonomous. As autonomous procedures are so numerous an autonomous iterative procedure will be defined first. This definition will then motivate the more abstract definition for the general iterative procedure.

The Autonomous Iterative Procedure

Consider a particular problem (P) and a given metric space (V, ρ). Letting $\mathcal{P}(V)$ denote power set, define a point to set mapping $A: V \to \mathcal{P}(V)$. Then the autonomous iterative procedure operates as follows. Given $z^1 \in V$ assume $z^2, \cdots, z^k$ have been generated. Then, if $A(z^k) = \phi$ the procedure stops. Otherwise $y \in A(z^k)$ is a possible value for z^{k+1} and furthermore $z^{k+1} \in A(z^k)$.

The more general definition will now be stated.

The Iterative Procedure

Consider a particular problem (P) and a given metric space (V, ρ). For all $k \geq 1$ define a set $V_k \subset V$. For any point $z^k \in V_k$ define a set

$$A_k(z^k) \subset V_{k+1}.$$

The iterative procedure is as follows. Given $z^1 \in V_1$ assume $z^2, \cdots, z^k$

have been generated. If $\phi = A_k(z^k)$, the procedure stops. Otherwise any $y \in A_k(x^k)$ is a possible value of z^{k+1}, and furthermore, $z^{k+1} \in A_k(x^k)$.

It should be clear that the set $A(z^k)$ is the set of all successors to z^k for the autonomous procedure, while for the general procedure $A_k(z^k)$ is that set.

Before continuing it is useful to develop some notation for subsequences. The letter K, perhaps superscripted, will denote an infinite subsequence of the integers. Any subsequence of $\{z^k\}_{k=1}^{\infty}$ can be denoted $\{z^k\}_K$ for an appropriate K. If the subsequence converges to a point z^{∞} we write $z^k \to z^{\infty}$ $k \in K$. The subsequence $\{z^{k+1}\}_K$ is simply the subsequence formed by adding 1 to each $k \in K$. If $z^{k+1} \to z^{\infty+1}$ $k \in K$, then $z^{\infty+1}$ is the limit of the subsequence $\{z^{k+1}\}_K$. The notation $\{z^k\}_{K^1}$ where $K^1 \subset K$ will mean an infinite subsequence of subsequence $\{z^k\}_K$.

The Convergence Conditions

Our immediate goal is to determine some conditions that are necessary and sufficient for an iterative procedure to be a convergent algorithm. But first the concept of convergence must be clarified. It is difficult to write a foolproof definition of convergence other than the tautology that convergence is the property which all convergent algorithms have. To begin with we specify a set $\Omega \subset V$ called the solution set. Any point $z \in \Omega$ is called a solution point or solution, and the algorithm will seek points in Ω.

The set Ω will be defined by some given property; the property perhaps depending on the problem and the algorithm under consideration. Often Ω will be the set of optimal points to problem (P). However, many other

properties can be used to define Ω. For example Ω could be any one of the following: the set of all points in an ε neighborhood of an optimal point, all roots of an equation, all efficient or Pareto points, all equilibrium points, etc. Nevertheless, it is assumed that for any given problem and algorithm the set of solution points Ω has been defined. Of course, Ω may turn out to be empty for certain problems.

Ideally, we would like an algorithm either to determine a solution point if one exists or to indicate that a solution does not exist. In addition, if a solution does not exist, it should tell us why such a point does not exist. Unfortunately, such properties are far too stringent to impose upon any conceivable algorithm implementable on any conceivable digital computer. We, therefore, adopt a somewhat practical definition of an algorithm based upon the properties of extant convergent algorithms.

A <u>convergent algorithm</u> is an iterative procedure with the following properties:

a) If the procedure stops at a point z, then the algorithm indicates either that no solution exists or that z is a solution. Also if a point z^k is a solution, then either $A_k(z^k) = \phi$ or $y \in A_k(z^k)$ implies y is a solution.

b) If the procedure generates an infinite sequence of points none of which are solutions, then if all points are not on a compact set no solution point exists while if all points are on a compact set the limit of any convergent subsequence is a solution point.

<u>Sufficient Conditions</u>

Certain conditions known as convergence conditions may now be stated such that if an algorithm satisfies these conditions it is a convergent algorithm. The first condition is, roughly speaking, a compactness condition.

A key complication in nonlinear programming algorithms that may arise due to lack of compactness is that there may be no optimal point to problem (P). In this case the maximum operation on f has to be replaced by a supremum operation. Condition I is intended to circumvent such a problem.

Condition I. a) If for some z and k $A_k(z) = \phi$, then the algorithm indicates either that z is a solution or that no solution exists. Should z^k be a solution, then either $A_k(z^k) = \phi$ or $y \in A_k(z^k)$ implies y is a solution. b) If the procedure generates an infinite sequence of points none of which are solution points, then if a solution exists there is a compact set X such that $z^k \in X$ for all k.

This condition is akin to similar assumptions made for nonlinear programming algorithms [6, 15, 16]. If anything it is somewhat less restrictive than most assumptions of this type.

Condition II is the crucial assumption that guarantees convergence.

Condition II. If $z^k \in X$, a compact set, for all k, then there exists a continuous function $Z: X \to E^1$ such that:

II-a) Given any point z^k then there exists an L_k such that for all $\ell \geq L_k + k$

$$Z(z^\ell) \geq Z(z^k).$$

II-b) Suppose the algorithm generates an infinite sequence of points none of which are solutions. Also suppose there exists a convergent subsequence $z^k \to z^\infty$ $k \in K$ such that z^∞ is not a solution. Then there is a K^1 such that $z^k \to z^*$ $k \in K^1$ and

$$Z(z^*) > Z(z^\infty).$$

The previously developed conditions will now be proven sufficient to insure that an iterative procedure is actually a convergent algorithm, in that it will satisfy the definition of convergence.

Theorem 1. Consider an iterative procedure on a metric space (V, ρ) and assume conditions I and II hold. Then the procedure is a convergent algorithm.

Proof. By I-a we are assured that if the algorithm stops at z then either z is a solution or that no solution point exists. Also if z is a solution any successor point is also a solution.

Consider now that the procedure generates an infinite sequence of points $\{z^k\}_{k=1}^{\infty}$ none of which are solutions. If the points are not all on a compact set, then by I-b no solution point exists. If all points are on a compact set, then any subsequence must contain a convergent subsequence. It only remains to prove that the limit of any convergent subsequence must be a solution.

It first will be shown that the sequence $\{Z(z^k)\}_{k=1}^{\infty}$ itself has a limit. Applying II-a there exists a sequence $\{z^{k*}\}$ such that

$$Z(z^{k*}) = \text{minimum } \{Z(z^{\ell}) \mid \ell \geq k\}. \tag{3}$$

Furthermore, the sequence $\{Z(z^{k*})\}$ is monotonic increasing. Also by compactness of X a convergent subsequence $\{z^{k*}\}_{K*}$ may be extracted from $\{z^{k*}\}$ such that $z^k \to z^*$ $k \in K^*$. By monotonicity

$$\lim_{k^* \to \infty} Z(z^{k*}) = \lim_{k \in K^*} Z(z^k) = Z(z^*) \tag{4}$$

where the final equality is by continuity of the Z.

Now consider any convergent subsequence $z^k \to z'$ $k \in K'$. By continuity $\lim_{k \in K'} Z(z^k) = Z(z')$. Given k^* and using (3) we may select $k' \in K'$ so large that $Z(z^{k'}) \geq Z(z^{k*})$. Hence

$$Z(z') \geq Z(z^*). \tag{5}$$

But by II-a given any $k' \in K'$ there is an L_k such that $\ell \geq L_k + k$ implies

$$Z(z^{k'}) \leq Z(z^{\ell}) .$$

Thus for k^* large enough

$$Z(k') \leq Z(z^{k^*}).$$

Monotonicity of $\{Z(z^{k^*})\}$ implies $Z(z^{k'}) \leq Z(z^*)$. Thus

$$Z(z') \leq Z(z^*). \tag{6}$$

By (5) and (6)

$$Z(z^*) = Z(z').$$

As this holds for any limit point z' it must be that

$$\lim_{k\to\infty} Z(z^k) = Z(z^*). \tag{7}$$

Up to this point II-b has not been employed. It will be used now. Let $z^k \to z^\infty$ $k \in K$. It must be proven that z^∞ is a solution. Assume z^∞ is not a solution. Then by II-b there is a K^1 such that $z^k \to z^*$ $k \in K^1$ and

$$Z(z^*) > Z(z^\infty).$$

But by (7) this is impossible. Hence z^∞ must be a solution.

Q.E.D.

The next corollary will add insight into the previous theorem. It is useful for the autonomous case. But before we state it, we must define a closed map.

A map $A: V \to \mathcal{P}(V)$ is said to be <u>closed</u> at z^∞ if

a) $z^k \to z^\infty$ $\quad k \in K,$

b) $y^k \to y^\infty$ $\quad k \in K$

and

c) $y^k \in A(z^k) \qquad k \in K$

imply

$$y^\infty \in A(z^\infty).$$

Corollary 1-1. This is the same as Theorem 1 except that Condition II is replaced by II' where:

Condition II': If $z^k \in X$, a compact set, for all k, then there exists a continuous function $Z: X \to E^1$ such that

II'-a) If z is a solution, then either the procedure terminates or $y \in A(z)$ implies

$$Z(y) \geq Z(z).$$

While if z is not a solution, $y \in A(z)$ implies

$$Z(y) > Z(z).$$

II'-b) The map $A(z)$ is closed at any z not a solution.

Proof. Clearly II'-a) implies II-a). We now show that II-b holds. Let

$$z^k \to z^\infty \qquad k \in K$$

where z^∞ is not a solution. By compactness of X, there is a $K^1 \subset K$ such that the subsequence $(z^{k+1})_{K^1}$ converges.

$$z^{k+1} \to z^* \qquad k \in K^1 .$$

However,

$$z^{k+1} \in A(z^k) \qquad k \in K^1 .$$

Using the definition of closedness

$$z^* \in A(z^\infty).$$

But z^∞ by assumption is not a solution. Therefore from II'-a

$$Z(z^*) > Z(z^\infty).$$

Q.E.D.

For further implications of this theorem see [16].

It should be remarked that if V is a finite set, I-a and II'-a insure finite convergence.

Necessary Conditions for Convergence

It will now be shown that using the previous definition of convergent algorithm, I and II necessarily follow.

Theorem 2: Consider an iterative procedure on (V, ρ) which is a convergent algorithm. Let Ω, the set of all solution points, be closed. Then conditions I and II necessarily follow.

Proof: Condition I-a holds easily as it is a) of the definition of convergence. Assume that an infinite sequence of points is generated none of which are solutions. If all points are not on a compact set no solution point exists so that I-b holds.

Assume therefore that $z^k \in X$ for all k where X is compact. It must now be shown that II-a holds.

(9) $$\text{Define } Z(z) = -\inf\{\rho(z, y); y \in \Omega\}.$$

It is straightforward to show that Z is a continuous function $Z: X \to E^1$. Note that $Z(z) = 0$ implies $z \in \Omega$ as Ω is closed.

Consider the sequence $\{Z(z^k)\}_{k=1}^{\infty}$. It will be established that

(10) $$\lim_{k\to\infty} Z(z^k) = 0.$$

But this must be true for consider any subsequence $z^k \to z^\infty$ $k \in K$. Then

$$\lim_{k\in K} Z(z^k) = Z(z^\infty) = 0.$$

employing the continuity of Z and the fact that by hypothesis any limit point z^∞ is a solution.

Now analyze II-a. Let z^k be optimal. Then by a) of the convergence definition, if there exists a $y \in A_k(z^k)$, then y is also optimal. Hence for any $\ell \geq k$

$$Z(z^k) = Z(z^{k+1}) = Z(z^\ell) = 0.$$

Assume now z^k is not optimal. Then $Z(z^k) < 0$. But by (10) there exists an L_k such that for all $\ell \geq L_k + k$, $Z(z^k) < Z(z^\ell) \leq 0$.

Only Condition II-b remains. But II-b holds vacuously as any limit point of any convergent subsequence must be a solution.

Q.E.D.

Application of the Convergence Conditions

In this section a representative sample of the better known algorithms will be proved to converge using the convergence conditions. In several cases the convergence proofs are considerably simpler than the original proofs. But given any algorithm the convergence conditions provide a framework from which to start a convergence proof. Presumably such a framework is a better place to commence than starting each proof from scratch. Proving convergence has not yet been reduced to filling in the blanks. But it is hoped that the convergence conditions will simplify the problem.

Occasionally in the following algorithms certain assumptions are made which are slightly stronger than the corresponding assumptions made by the algorithm's originators. The purpose of this is solely for clarity. The proofs also hold with the weaker assumptions.

Unconstrained Maxima

Some of the simplest, yet most useful, algorithms seek the unconstrained maximum of f over E^n. An important class of these algorithms, known as unconstrained feasible direction algorithms, are easily proven to satisfy

conditions I and II of Corollary 1-1 and hence are convergent algorithms, as is now shown.

Each algorithm in this class possesses a continuous function $b: E^n \to E^n$ that serves as a direction. Briefly given a point x^k, the successor x^{k+1} is generated by maximizing f in the direction $b(x^k)$.

Denoting $\nabla f(x)$ as the gradient of f evaluated at x, a point x is termed a solution if $\nabla f(x)b(x) = 0$. We specify the map A by $x' \in A(x)$ if and only if x' is an optimal solution to

$$\max\{f(x + \tau b(x)) \mid \tau \geq 0\}. \tag{11}$$

The unconstrained feasible direction algorithm operates as follows. If x^k is a solution stop. Otherwise define the successor by calculating $x^{k+1} \in A(x^k)$. For simplicity let τ^k satisfy

$$x^{k+1} = x^k + \tau^k b(x^k). \tag{12}$$

To prove convergence the following two assumptions will be needed.

i) Either f has no solution point or the set $\{x \mid f(x) \geq f(x^0)\}$ is compact for any x^0

and

ii) If x is not a solution, then $x' \in A(x)$ implies $f(x') > f(x)$.

It will now be shown that any unconstrained feasible direction algorithm which satisfies i) and ii) also satisfies conditions I and II'. Assumption i) insures that I holds since if no x' satisfies equation (11) then there is no solution point. Condition II'-a is verified by letting $V = E^n$, $x^k = z^k$, and $Z(z) = f(x)$, because (11) insures that $f(x^k)$ is monotonic. Also ii) provides that if x^k is not a solution

$$f(x^{k+1}) > f(x^k).$$

Condition II'-b will now be established. Let $x^k \to x^\infty$ and $x^{k+1} \to x^{\infty+1}$ $k \in K$. By continuity $b(x^k) \to b(x^\infty)$, $k \in K$. Assume x^∞ is not a solution, then (by definition of solution) $b(x^\infty) \neq 0$. It then follows from (12) that for some $\tau^\infty \geq 0$, $\tau^k \to \tau^\infty$ $k \in K$. Using (11) for any $\tau \geq 0$

$$f(x^{k+1}) = f(x^k + \tau^k b(x^k)) \geq f(x^k + \tau b(x^k)).$$

Taking limits

$$f(x^{\infty+1}) \doteq f(x^{\infty}+\tau^{\infty}b(x^{\infty})) \geq f(x^{\infty}+\tau b(x^{\infty})).$$

As this holds for any $\tau \geq 0$

$$f(x^{\infty+1}) = \max\{f(x^{\infty} + \tau b(x^{\infty}))|\tau \geq 0\} .$$

Therefore the map A is closed and condition II'-b holds. The algorithm converges.

By applying the above reasoning two popular algorithms are seen to converge. The first is the Cauchy [2] procedure. This procedure assumes the f is continuously differentiable and defines $b(x) = \nabla f(x)$. The second is a modified Newton-Raphson algorithm. For this algorithm f is assumed to have continuous second partial derivatives and its Hessian matrix at x, $H(x)$, is assumed negative definite for all x. Defining $H^{-1}(x)$ as the inverse of the Hessian, $b(x) = -H^{-1}(x)\nabla f(x)$. It might also be noted that the above reasoning provides an alternative to Theorem 2 of Topkis and Veinott [12].

<u>The Frank and Wolfe Algorithm</u>

The Frank and Wolfe algorithm [6] is for problem (P) when all constraints g_i are linear and f is continuously differentiable. Assume the feasible region F is compact. Given any $x \in F$ define $A(x)$ as follows. Solve the linear programming problem where x is fixed via the Simplex Method

$$\max\{\nabla f(x)w|w \in F\} \tag{13}$$

for an optimal w^1. Then $x' \in A(x)$ if and only if x' is an optimal solution to

$$\max\{f(x + \tau(w^1-x))|0 \leq \tau \leq 1\}. \tag{14}$$

Note that for $0 \leq \tau \leq 1$, $x + \tau(w-x) \in F$. Therefore given $x^1 \in F$ all successor points will also be feasible.

A point x is called a solution if $\nabla f(x)(w^1-x) = 0$ where w^1 solves (13). Should x be a solution the procedure stops and $A(x) = \phi$. The algorithm is now specified and we will prove its convergence using conditions I and II.

Condition I holds immediately as F is compact. Letting $V = F$, $z = x$, and $Z(z) = f(z)$, equation (14) insures that II-a holds.

Now condition II-b must be established. Let $x^k \to x^\infty$ and $x^{k+1} \to x^{\infty+1}$ for $k \in K$. Assume x^∞ is not a solution. Define w^k as the optimal point to problem (13) when $x = x^k$. From the theory of linear programming and the fact that f has continuous derivatives, there must exist a $K^1 \subset K$ such that $w^k \to w^\infty$ $k \in K^1$ where w^∞ solves (13) for $x = x^\infty$. Given any fixed τ, $0 \leq \tau \leq 1$

$$f(x^{k+1}) \geq f(x^k + \tau(w^k - x^k)).$$

From the continuity

$$f(x^{\infty+1}) \geq \max\{f(x^\infty + \tau(w^\infty - x^\infty)) \mid 0 \leq \tau \leq 1\} > f(x^\infty),$$

where the final inequality holds as x^∞ not optimal implies

$$\nabla f(x^\infty)(w^\infty - x^\infty) > 0.$$

Hence, condition II-b holds. The algorithm can also be proved to converge via (Corollary 1.1.)

Modifications of the above reasoning can be used to validate that many similar algorithms satisfy the convergence condition. In particular, the decomposable nonlinear programming method of Zangwill [14], Zoutendijk's methods of feasible directions [17], and the Convex Simplex Method of Zangwill [15] fall into this category.

The Direction Function

Topkis and Veinott [12] consider the concept of a direction function.

This concept is quite useful for proving that feasible direction algorithms converge. Let F, the feasible region, be compact. A direction $b \in E^n$ is called a feasible direction at $x \in F$ if for some $t > 0$, $x + \tau b \in F$, for all $0 < \tau \leq t$. If b is feasible at x and, in addition, $f(x) < f(x+\tau b)$ for all $0 < \tau \leq t$, then b is also said to be usable at x. Should no usable direction exist at $x \in F$, the point x is termed a solution. As f is continuous and F is compact a solution point clearly exists. For every sequence $\{x^1, x^2, x^3, \cdots, x^k, \cdots\}$ in F we assume there is a direction function b which assigns to $\{x^1, \cdots, x^k\}$ a feasible direction $b^k = b(x^1, x^2, \cdots, x^k)$ at x^k. All b^k are contained in a compact set.

Let $(z^k, b^k) \to (z^\infty, b^\infty)$ $k \in K$, then the Topkis-Veinott conditions specify that

i) For some $t > 0$, $x^k + \tau b^k \in F$ for all $k \in K$ and all $0 < \tau \leq t$.

ii) If b^∞ is feasible but not usable at x^∞, then x^∞ is a solution.

Also there is a real-valued lower semi-continuous step size function $f^1(x, w)$ defined on the Cartesian product $F \otimes F$ such that $f^1(x, x+\tau b)$ is continuous in τ. This function satisfies conditions

iii) $f^1(x, x) = f(x)$ and $f^1(x, w) \leq f(w)$ for $w, x \in F$,

and

iv) if b is usable at x for f, then b is usable for $f^1(x, \cdot)$ at x.

The algorithmic procedure is defined as follows. Given $x^1, x^2, \cdots, x^k$, if x^k is a solution the procedure stops. Otherwise the point x^{k+1} is generated by

$$(15) \qquad f^1(x^k, x^{k+1}) = \max\{f^1(x^k, x^k + \tau b^k) \mid \tau \geq 0,\ x^k + \tau b^k \in F\}.$$

It must now be shown that any procedure which satisfies the above four conditions also satisfies I and II. Let E^n be the metric space and define $V = F$ and $V_k = \{x^k\}$ for all k. The algorithm is then defined as follows

$$A_k(x^k) = \begin{cases} \phi & \text{if } x^k \text{ is a solution} \\ \{x^{k+1}\} & \text{otherwise} \end{cases}$$

Also let $Z(x) = f(x)$.

Condition I holds as the set F is compact. Condition II-a holds because equation (15) and condition iii) insure that f is monotonic.

Condition II-b now must be established. Let $x^k \to x^\infty$ $k \in K$, $x^{k+1} \to x^{\infty+1}$ $k \in K$ and $b^k \to b^\infty$ $k \in K$. Assume x^∞ is not a solution. It must be proved that

$$f(x^{\infty+1}) > f(x^\infty).$$

By i) and the compactness of F, $x^\infty + \tau b^\infty \in F$ for all $0 \leq \tau \leq t$, and thus b^∞ is feasible. Since b^∞ is feasible but x^∞ is not a solution, b^∞ must be usable by ii). Then via iv) there exists a $0 < \tau^1 \leq t$ such that

$$f^1(x^\infty, x^\infty + \tau^1 b^\infty) > f^1(x^\infty, x^\infty) = f(x^\infty). \tag{16}$$

Furthermore, because $x^k \to x^\infty$ $k \in K$ and $b^k \to b^\infty$ $k \in K$

$$x^k + \tau^1 b^k \to x^\infty + \tau^1 b^\infty \quad k \in K.$$

In addition, by i) as $\tau^1 \leq t$, $x^k + \tau^1 x^k \in F$ for all $k \in K$.

Using (15) and iii)

$$f(x^{k+1}) \geq f^1(x^k, x^{k+1}) \geq f^1(x^k, x^k + \tau^1 b^k).$$

Exploiting the continuity of f and lower semi-continuity of f^1,

$$f(x^{\infty+1}) = \lim_{k\in K} f(x^{k+1}) \geq \liminf_{k\in K} f^1(x^k, x^{k+1})$$

$$\geq \liminf_{k\in K} f^1(x^k, x^k + \tau^1 b^k)$$

$$\geq f^1(x^\infty, x^\infty + \tau^1 b^\infty).$$

Hence II-b holds for using (16)

$$f(x^{\infty+1}) > f(x^\infty).$$

The fact that the Topkis-Veinott conditions are a special case of the convergence conditions I and II establishes that the many algorithms proved by their techniques can also be proved by the convergence conditions. One class of algorithms subsumed by these conditions are the so-called cyclic coordinate ascent methods. These methods optimize one coordinate at a time.

Arrow-Hurwicz-Uzawa Algorithm

The Uzawa iterative adaption [1] of the Arrow-Hurwicz gradient method considers the following modification of (P)

$$\begin{aligned} \text{maximize} \quad & f(x) \\ \text{subject to} \quad & g(x) \geq 0 \\ & x \geq 0 \end{aligned}$$

where f is strictly concave and $g(x) = (g_1(x), \cdots, g_m(x))$ is a vector of the constraints g_i each of which is assumed concave. All functions are continuously differentiable on E^n. It is also supposed that a vector x^0 exists such that $x^0 \geq 0$ and $g_i(x^0) > 0$ for all i.

Define a Lagrangean function

$$\Psi(x, u) = f(x) + ug(x)$$

where u is an m vector. Consider a point $(\bar{x}, \bar{u})$ assumed to exist, called a saddle point, such that

$$\Psi(\bar{x}, \bar{u}) = \max_{x \geq 0} = (x, \bar{u}) = \min_{u \geq 0} \Psi(\bar{x}, u). \tag{19}$$

By strict concavity there is a unique $\bar{x}$ which solves (19). Let $U = \{u | (\bar{x}, u)$ is a saddle point of $\Psi(x, u)\}$. It is easy to show that U is compact [1, p. 155].

The algorithm is defined by the difference equations

$$x^{k+1} = \max[0, x^k + \tau\Psi_x(x^k, u^k)] \tag{20a}$$

$$u^{k+1} = \max[0, u^k - \tau\Psi_u(x^k, u^k)] \tag{20b}$$

where $x^1 \geq 0$ and $u^1 \geq 0$, τ is a positive scalar to be specified subsequently, and $\Psi_x = \nabla_x\Psi(x^k, u^k)$ and $\Psi_u = g(x^k)$ are respectively the partial derivatives of Ψ with respect to x and u.

Given $\varepsilon > 0$ a point $z = (x, u)$ is termed a solution if

$$|\bar{x} - x|^2 < \varepsilon.$$

The algorithm commences with an initial point $z^1 = (x^1, u^1)$ and recursively generates points $z^k = (x^k, u^k)$ via equations (20a) and (20b) until a solution is obtained. The algorithm stops at a solution.

To specify τ first define

$$Z(x, u) = \min_{\bar{u} \in U} \{|x - \bar{x}|^2 + |u - \bar{u}|^2\}.$$

Then

$$\tau = \min \left\{ \frac{(\bar{x}-x)\Psi_x - (\bar{u}-u)\Psi_u}{|\Psi_x|^2 + |\Psi_u|^2} \middle| \frac{\varepsilon}{2} \leq |x-\bar{x}|^2,\ Z(x,u) \leq J,\ \bar{u} \in U \right\}$$

where $J = Z(x^1, u^1)$. The value of τ will be positive because the numerator of the function being minimized is positive as the next lemma proves, also the function is continuous and the region of minimization is compact.

Lemma 3.

$$(\bar{x} - x)\Psi_x - (\bar{u} - u)\Psi_u > 0 \qquad \text{if } x \neq \bar{x}.$$

Proof: By concavity of Ψ in x

$$\Psi(\bar{x}, u) \leq \Psi(x, u) + (\bar{x} - x)\Psi_x .$$

Since $\Psi_u = g$,

$$\Psi(x, \bar{u}) - (\bar{u} - u)\Psi_u = \Psi(x, u).$$

Therefore

$$\Psi(\bar{x}, u) - \Psi(\bar{x}, u) \leq (\bar{x} - x)\Psi_x - (\bar{u} - u)\Psi_u. \tag{1}$$

Now by definition of a saddle point, and since f is strictly concave

$$\Psi(x, \bar{u}) < \Psi(\bar{x}, \bar{u}) \leq \Psi(\bar{x} - u)$$

Therefore from (1)

$$0 < \Psi(\bar{x}, u) - \Psi(x, \bar{u}) \leq (\bar{x} - u)\Psi_x - (\bar{u} - u)\Psi_u. \qquad \text{Q.E.D.}$$

We now prove convergence via Corollary 1.1. Condition II will be established first. If $z = (x, u)$ is a solution the procedure terminates. Now suppose $z^k = (x^k, u^k)$ is not a solution. After some manipulation it can be shown that [1, page 156]

$$|x^{k+1} - \bar{x}|^2 + |u^{k+1} - \bar{u}|^2 \leq |x^k - \bar{x}|^2 + |u^k - \bar{u}|^2$$

$$- \tau\{2[(\bar{x} - x^k)\Psi_x - (\bar{u} - u)\Psi_u] - \tau[|\Psi_x|^2 + |\Psi_u|^2]\} \tag{2}$$

We will validate II'-a (reversing inequalities) by proving that

$$Z(x^{k+1}, u^{k+1}) < Z(x^k, u^k) \tag{3}$$

However, to prove (3) we see from (2) that it is only necessary to show

$$2[(\bar{x}-x^k)\Psi_x - (\bar{u}-u)\Psi_u] - \tau[|\Psi_x|^2 + |\Psi_u|^2] > 0. \tag{4}$$

Employing the definition of τ, the left side of (4) is larger than

$$2[(\bar{x} - x^k)\Psi_x - (\bar{u}-u)\Psi_u] - \left(\frac{(\bar{x}-x^k)\Psi_x - (\bar{u}-u^k)\Psi_u}{|\Psi_x|^2 + |\Psi_u|^2}\right)(|\Psi_x|^2 + |\Psi_u|^2)$$

$$= [(\bar{x} - x^k)\Psi_x - (\bar{u} - u^k)\Psi_u]$$

$$> 0$$

where the final inequality holds via Lemma 3 as z^k is not a solution. Thus II'-a) holds.

Condition II'-b) holds immediately as the recursions (20a) and (20b) are continuous functions.

We observe that from equation (3) for all k

$$Z(x^k, u^k) \leq J.$$

Therefore all $z^k = (x^k, u^k)$ generated are on a compact set. Furthermore, by assumption a saddle point exists, therefore condition I also holds. The algorithm converges.

Other Algorithms

Several other algorithms have been proved by application of the convergence conditions. In particular, loss function methods such as Zangwill's penalty function method [13] and Fiacco and McCormick's sequential unconstrained approach [5] have been established using the convergence conditions. Also cutting plane methods [8] have been considered.

Conclusion

This paper has presented an attempt to unify the convergence proofs of nonlinear programming algorithms. Both necessary and sufficient conditions for convergence were discussed. The methods presented seem related to but are actually somewhat different than the differential equation stability theory of Liapunov [11]. For the special case in which $A(x)$ is a function, that is, $A(x_k) = x_{k+1}$, and in addition $A(x)$ is continuous, the convergence conditions may be considered as Liapunov conditions for the nonlinear programming case. Finally, it should also be clear by selecting the solution set Ω astutely, algorithms other than the nonlinear programming algorithm can be considered.

Acknowledgment

The author expresses his appreciation to Professor R. van Slyke for his comments on a previous version of this paper. Professor James Friedman suggested an important simplification of Condition II-b.

References

1. Arrow, K. J., Hurwicz, L., and Uzawa, H. Studies in Linear and Nonlinear Programming, Stanford University Press, Stanford, California, 1958.

2. Cauchy, A. L., "Method General pour la Resolution des Systems d'Equations Simultanees, C.R.R. Sci., Paris, 1847, Ch. 25, pp. 536-538.

3. Cheney, E. W. and Goldstein, A. A., "Newton's Method of Convex Programming and Tchebycheff Approximation," Numer. Math., Vol. 1, 1959, pp. 253-268.

4. Dantzig, G. B., Linear Programming and Extensions, Princeton University Press, Princeton, N. J., 1963.

5. Fiacco, A. V. and McCormick, G. P., "The Sequential Unconstrained Minimization Technique for Nonlinear Programming, a Primal-Dual Method," Management Science, Vol. 10, No. 2, pp. 360-364, 1964.

6. Frank, M. and P. Wolfe, "An Algorithm for Quadratic Programming," Naval Research Logistics Quarterly, Vol. 3, 1956, pp. 95-110.

7. John, F., "Extremum Problems with Inequalities as Subsidiary Conditions," Studies and Essays, Courant Anniversary Volume, Interscience, New York, 1948.

8. Kelley, J. E., "The Cutting Plane Method for Solving Convex Programs," J. Soc. Indust. Appl. Math., Vol. 8, No. 4, December 1960, pp. 703-712.

9. Kelley, J. L., General Topology, D. Van Nostrand Co., Inc., Princeton, New Jersey, 1963.

10. Kuhn, H. W. and Tucker, A. W., "Non-Linear Programming," Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, J. Neyman (ed.), University of California Press, Berkeley, 1951, pp. 481-493.

11. La Salle, J. and Lefschetz, S., Stability by Liapunov's Direct Method, Academic Press, N. Y., London, 1961.

12. Topkis, D. M. and Veinott, Jr., A. F., "On the Convergence of Some Feasible Direction Algorithms for Nonlinear Programming," SIAM Journal on Control, Vol. 5, No. 2, 1967, pp. 268-279.

13. Zangwill, W. I., "Non-Linear Programming via Penalty Functions," Management Science-A, Vol. 13, No. 5, pp. 344-358.

14. _____, "A Decomposable Nonlinear Programming Approach," Operations Research, Vol. 15, No. 6, Nov-Dec 1967, pp. 1068-1087.

15. _____, "The Convex Simplex Method," Management Science-A, Vol. 14, No. 3, Nov. 1967, pp. 221-238.

16. _____, "Application of the Convergence Conditions," Working Paper No. 231, Center for Research in Management Science, University of California, Berkeley, August 1967.

17. Zoutendijk, G., Methods of Feasible Directions, Else ier Publ. Co., Amsterdam, 1960.