MATHEMATICAL PROGRAMMING
R.W. Cottle, M.L. Kelmanson and B. Korte (Editors)

AUTOMATIC STEP SIZES FOR THE FORTIFIED DESCENT ALGORITHMS IN MONOTROPIC PROGRAMMING

R.T. Rockafellar

Department of Mathematics
University of Washington,
Seattle, U.S.A.

A general monotropic programming problem consists in minimizing, subject to linear constraints, a convex function that is preseparable, i.e. expressible as the sum of finitely many terms, each of which is a linear function composed with a convex function of a single variable.

Any such problem can be reduced to the canonical form:

$$\text{(P)} \qquad \text{minimize} \quad F(x) = \sum_{j=1}^{n} f_j(x_j) \quad \text{subject to}$$
$$x \in \mathcal{C} \ \text{and} \ x_j \in C_j \quad \text{for} \ j = 1,\dots,n,$$

where $\mathcal{C}$ is a linear subspace of R^n (describable by a homogeneous system of linear equations), each C_j is a nonempty interval in R, and f_j is a finite convex function on C_j. The interval C_j is not required to be closed, but the following technical assumption is imposed: under the convention that

$$f_j(x_j) = +\infty \quad \text{for} \quad x_j \notin C_j \tag{1}$$

the function f_j is continuous relative to the closure of C_j. (This ensures that f_j, interpreted via (1) as a proper convex function on all of R, is closed in the sense of [1,§§7,24].)

Linear programming and (convex) quadratic programming are special cases of monotropic programming. Taking the f_j's to be piecewise linear or piecewise quadratic, one gets useful extensions of these classical problem types. Many problems in network programming fit directly into the form (P) too. In that context, x_j is the flow in the jth arc of a certain directed graph, and $\mathcal{C}$ is the space of all flows $x = (x_1,\dots,x_n)$ which are circulations in the graph, i.e. conserved at every node. See [2] for more on such examples.

An important characteristic of monotropic programming problems, which distinguishes them from convex programming problems in general and shows a close kinship with linear and network programming problems, is the fact that they can be solved by descent methods where the descent directions can be determined *combinatorially*, e.g. by pivoting. We have explained this in [2], [3], and demonstrated that directions yielding descent by more than a *pre-assigned* amount ε can actually be generated in such a way. The purpose of the present article is to provide a step size rule which effects this guaranteed amount of descent automatically, without the need for any line search in the direction in question. This property is particularly desirable as a means of implementing the dual approach to solving (P), where descent techniques are applied to a certain dual monotropic programming problem (D). This will be discussed at the end of the article.

Fix any $\varepsilon \geq 0$ and consider for any feasible solution x to (P) the closed intervals

$$[\lambda_j^-(x_j),\lambda_j^+(x_j)] = \{v_j \in R \mid f_j(x_j+s) \geq f_j(x)_j + v_j s-\varepsilon, \quad \forall s \in R\} \qquad (2)$$

where $\lambda_j^+(x_j)$ and $\lambda_j^-(x_j)$ represent extremal slope values depending on ε, as indicated in Figure 1. Note that one just has

$$\lambda_j^+(x_j) = (f_j)'_+(x_j) \quad \text{and} \quad \lambda_j^-(x_j) = (f_j)'_-(x_j) \quad \text{when } \varepsilon = 0 \qquad (3)$$

In the latter case it could happen that $\lambda_j^+(x_j) = \lambda_j^-(x_j) = +\infty$ or $\lambda_j^+(x_j) = \lambda_j^-(x_j) = -\infty$ for some j, so the interval $[\lambda_j^-(x_j), \lambda_j^+(x_j)]$ is empty; then x is said not to be *regularly* feasible. There are no such worries when $\varepsilon > 0$, since then $\lambda_j^+(x_j) > -\infty$ and $\lambda_j^-(x_j) < +\infty$. Of course, one has $\lambda_j^+(x_j) = +\infty$ if and only if x_j is the right endpoint of C_j (the interval where f_j is finite), and $\lambda_j^-(x_j) = -\infty$ if and only if x_j is the left endpoint of C_j. More will be said later about how the values $\lambda_j^+(x_j)$ and $\lambda_j^-(x_j)$ might be determined. For the time being, we simply assume they are readily available.

Let D denote the linear subspace of R^n orthogonally complementary to C. (This can be described by a homogeneous system of linear equations dual to the system used to describe C, see [3].) For any $x \in C$ such that the "rectangle"

$$\Lambda(x) = \prod_{j=1}^{n} [\lambda_j^-(x_j),\lambda_j^+(x_j)] \qquad (4)$$

is nonempty (this being true for any feasible x when $\varepsilon > 0$, but only for regularly feasible x when $\varepsilon = 0$), one will have either that $\Lambda(x) \cap D \neq \emptyset$ or that $\Lambda(x)$ can be separated strongly from D. In other words, either

$$\exists v \in D \quad \text{satisfying} \quad \lambda_j^-(x_j) \leq v_j \leq \lambda_j^+(x_j) \quad \text{for all } j \qquad (5)$$

or else

$$\exists z \in C \quad \text{satisfying} \quad \sum_{j:z_j>0} \lambda_j^+(x_j)z_j + \sum_{j:z_j>0} \lambda_j^-(x_j)z_j < 0 \qquad (6)$$

Indeed, there exist combinatorial algorithms which in finitely many steps can resolve the question constructively, by producing either a v as in (5) or a z as in (6), see [2]. Moreover, this can be accomplished in such a manner that in the case of (6), the vector z which is obtained points in one of the finitely many so-called *elementary* directions of C.

For our purposes here, we need not go into the nature of such algorithms nor into the properties of "elementary" vectors. We wish merely to take for granted that a routine can be called forth, whenever required, to determine constructively whether (5) or (6) holds. The following result, which we proved in [3], indicates how such a routine can be made on the basis of a descent algorithm for (P).

THEOREM 1 [3]

Let x *be any feasible solution to* (P) *(regularly feasible if the case* $\varepsilon = 0$ *is under consideration). If* (5) *holds, then* x *is approximately optimal for* (P) *in the sense that*

$$F(x) \leq \inf(P) + \delta \quad \text{for} \quad \delta = n\varepsilon \tag{7}$$

If (6) *holds, on the other hand, the vector* z *gives a direction of* ε*-descent in the sense that*

$$\begin{aligned} &\exists t > 0 \quad \text{with} \quad x + tz \quad \text{again feasible for} \quad (P) \quad \text{and} \\ &F(x + tz) < F(x) - \varepsilon \end{aligned} \tag{8}$$

In the case of $\varepsilon = 0$, condition (7) means that x is an optimal solution to (P), while (8) means that x can be improved in the direction of z. It can be shown that the improved feasible solution $x+tz$ will again be regularly feasible in this case. When $\varepsilon > 0$, however, the conclusions are stronger: if x is not already δ-optimal, an improvement by more than ε, through line search in the direction of z, is guaranteed. Here, of course, $\delta > 0$ could be the number specified in advance, and ε would then be taken equal to δ/n.

It follows that for arbitrary $\varepsilon > 0$ and $\alpha \in R$, we can start with any feasible solution x^0 to (P) and determine in finitely many steps an x which is either δ-optimal for (P), or feasible for (P) with $F(x) < \alpha$. In the kth iteration, if it is not already true that the current feasible solution x^{k-1} satisfies $F(x^{k-1}) < \alpha$, we apply a combinatorial routine of the sort mentioned above in order to decide constructively whether (5) or (6) holds for $x = x^{k-1}$. If (5) holds, we terminate, because (7) is true for $x = x^{k-1}$. If (6) holds, we get a vector z^k, and by a line search in the direction of z^k, if necessary, we can find a t_k such that the vector $x^k = x^{k-1} + t_k z^k$ is again feasible and satisfies $F(x^k) < F(x^{k-1}) - \varepsilon$. We then proceed with iteration $k+1$. Note that the number of iterations before termination cannot exceed the *a priori* bound

$$1 + (n/\delta)[F(x^0) - \max\{\alpha, \inf(P)\}]$$

Obviously, when termination does come, the algorithm can be restarted with lower values of ε, δ, and α, if so desired.

We speak of the preceding method, with $\varepsilon > 0$, as *fortified descent* in monotropic programming. The corresponding method with $\varepsilon = 0$ generates an improving sequence of feasible solutions x^k, but apart from special situations there is no assurance even that $F(x^k) \to \inf(P)$, much less that termination will come in a finite number of iterations, regardless how many.

Automatic Step Size Rule

Let us now look more closely at the task of determining a step size t which meets the prescription in (8). What we know in general is that $x \in C$, $F(x) < \infty$ (this means $x_j \in C_j$ for all j, according to the convention in (1)), and z is a vector in C such that

$$\inf_{t>0} F(x+tz) < F(x) - \varepsilon$$

Since $F(x+tz)$ is convex as a function of t, the approximate minimization of $F(x+tz)$ with respect to $t > 0$ could be carried out relatively easily, at least in principle, so as to get a value of t with $F(x+tz) < F(x)$. Then $x+tz$ would be another vector in C, in fact a feasible solution to (P) (because

$F(x+tz) < \infty$), and our task would be finished.

There are reasons, however, for wishing to avoid a line search for t, of the kind just described. For one thing, $F(x+tz)$ is the sum of a possibly large number of terms $f_j(x_j+tz_j)$. All these would have to be dealt with explicitly by any search method, and this could be tedious. Another potential drawback is that in certain situations which will be discussed later, the functions f_j to which the algorithm is applied may only be known in a somewhat indirect manner, making their values troublesome to compute, even though the slopes $\lambda_j^-(x_j)$ and $\lambda_j^+(x_j)$ may themselves be available.

We may suppose that along with $\lambda_j^-(x_j)$ and $\lambda_j^+(x_j)$ we have at our disposal certain numbers s_j^+ and s_j^- (not necessarily unique) as indicated in Figure 1. Specifically, with the assumption henceforth that $\varepsilon > 0$, let us note that

$$\begin{aligned} \lambda_j^+(x_j) &= \inf_{s>0} [f_j(x_j+s) - f_j(x_j) - \varepsilon]/s \\ \lambda_j^-(x_j) &= \sup_{s<0} [f_j(x_j+s) - f_j(x_j) - \varepsilon]/s \end{aligned} \tag{9}$$

and select arbitrary numbers

$$\begin{aligned} s_j^+ &\in \operatorname*{argmin}_{s>0}[f_j(x_j+s) - f_j(x_j) - \varepsilon]/s \\ s_j^- &\in \operatorname*{argmin}_{s<0}[f_j(x_j+s) - f_j(x_j) - \varepsilon]/s \end{aligned} \tag{10}$$

using the following conventions to extend those formulas:

$$\begin{aligned} s_j^+ &= 0 \quad \text{if } f_j(x_j+s) = \infty \text{ for all } s > 0 \quad (\text{i.e. } \lambda_j^+(x_j) = +\infty) \\ s_j^- &= 0 \quad \text{if } f_j(x_j+s) = \infty \text{ for all } s < 0 \quad (\text{i.e. } \lambda_j^-(x_j) = -\infty) \end{aligned} \tag{11}$$

and

$$\begin{aligned} &s_j^+ \text{ is allowed to be } +\infty \text{ if} \\ &\quad [f_j(x_j+s) - f_j(x_j) - \varepsilon]/s \to \lambda_j^+(x_j) \text{ as } s \to +\infty \end{aligned} \tag{12}$$

$$\begin{aligned} &s_j^- \text{ is allowed to be } -\infty \text{ if} \\ &\quad [f_j(x_j+s) - f_j(x_j) - \varepsilon]/s \to \lambda_j^-(x_j) \text{ as } s \to -\infty \end{aligned} \tag{13}$$

The latter conventions are engendered by the circumstance that the difference quotient $[f_j(x_j+s) - f_j(x_j) - \varepsilon]/s$ is unimodal in s, in fact as a function of $r = 1/s$ it is convex for $r > 0$ and concave for $r < 0$. In minimizing a convex function of r over $0 < r < \infty$, it is perfectly natural to consider $r = 0$ as a possible solution in the obvious limiting case, and this explains (12); similarly for (13). (The reader may again wish to refer to Figure 1 to see that we are alluding here to cases where $\lambda_j^+(x_j)$ and $\lambda_j^-(x_j)$ turn out, one or the other, to be limiting slopes for the graph of f_j.)

At any rate, it is clear that with the stated conventions there always do exist numbers s_j^+ and s_j^- corresponding to $\lambda_j^+(x_j)$ and $\lambda_j^-(x_j)$ as described and satisfying

$$\begin{aligned} 0 < s_j^+ \leq \infty \quad &\text{when} \quad \lambda_j^+(x_j) < \infty \\ 0 > s_j^- \geq -\infty \quad &\text{when} \quad \lambda_j^-(x_j) > -\infty \end{aligned} \tag{14}$$

Moreover, they can, at worst, be calculated along with $\lambda_j^+(x_j)$ and $\lambda_j^-(x_j)$ by way of the line searches in (9), and if necessary these searches can be executed in terms of relatively elementary and accessible convex functions of a single variable s. Sometimes this is terrifically easy; for instance, when f_j is piecewise quadratic relative to C_j, answers that are virtually in "closed form" can be given, see [2, Chap. 9].

We show now that such numbers s_j^+ and s_j^- furnish a value $\bar{t}$ which can be used without further ado as the step size in the fortified descent algorithm. This makes it possible to avoid a line search of the potentially more complicated kind suggested by (8).

THEOREM 2

With $\varepsilon > 0$, *let* x *be any feasible solution to problem* (P), *and suppose* z *is a vector as in* (6). *For arbitrary numbers* s_j^+ *and* s_j^- *corresponding to* $\lambda_j^+(x_j)$ *and* $\lambda_j^-(x_j)$ *as just described, let*

$$\bar{t} = \min \begin{cases} s_j^+/z_j & \textit{for } j \textit{ such that } z_j > 0 \\ s_j^-/z_j & \textit{for } j \textit{ such that } z_j < 0 \end{cases} \tag{15}$$

Then $\qquad 0 < \bar{t} \leq \infty$ *and*

$$F(x+\bar{t}z) < F(x) - \varepsilon \quad \textit{if} \quad \bar{t} < \infty \tag{16}$$

In the degenerate case where $\bar{t} = \infty$, $F(x+tz)$ *is nonincreasing as a function of* $t \in (0,\infty)$, *and*

$$\lim_{t\to\infty} F(x+tz) < F(x) - \varepsilon$$

Proof:

Observe at the outset that the inequality in (6) implies $\lambda_j^+(x_j) < \infty$ for indices j with $z_j > 0$, and $\lambda_j^-(x_j) > -\infty$ for j with $z_j < 0$. From (14), then, it is clear that none of the numbers s_j^+ or s_j^- involved in (15) can be 0, and therefore $\bar{t} > 0$ (since each of the ratios has the same sign in both numerator and denominator). Next note that for each j with $z_j > 0$ (hence $s_j^+ > 0$) one has by the choice of s_j

$$\begin{aligned} [f_j(x_j+s_j^+) - f_j(x_j) - \varepsilon]/s_j^+ &= \lambda_j^+(x_j) \quad \text{if} \quad s_j^+ < \infty \\ \lim_{s\to\infty} [f_j(x_j+s) - f_j(x_j) - \varepsilon]/s &= \lambda_j^+(x_j) \text{ if } \quad s_j^+ = \infty \end{aligned} \tag{18}$$

The first of the cases (18) yields

$$[f_j(x_j+tz_j) - f_j(x_j)]/t = \lambda_j^+(x_j)z_j - (\varepsilon/t) \quad \text{for} \quad t = s_j^+/z_j$$

and hence by the monotonicity of the difference quotient of the convex function f_j:

$$[f_j(x_j+tz_j) - f_j(x_j)]/t \le \lambda_j^+(x_j)z_j - \varepsilon(z_j/s_j^+) \tag{19}$$

$$\text{for} \quad 0 < t \le s_j^+/z_j \quad \text{when} \quad z_j > 0.$$

The second case in (18) yields the same inequality, provided only that we interpret z_j/s_j^+ as 0 when $s_j^+ = \infty$ (and correspondingly interpret the difference quotient for $t = \infty$ as a limit). By a parallel argument, we have

$$[f_j(x_j+tz_j) - f_j(x_j)]/t \le \lambda_j^-(x_j)z_j - \varepsilon(z_j/s_j^-) \tag{20}$$

$$\text{for} \quad 0 < t \le s_j^-/z_j \quad \text{when} \quad z_j < 0.$$

Adding up the inequalities in (19) for indices j with $z_j > 0$ and the ones in (20) for indices j with $z_j < 0$ in (6)), we obtain

$$\sum_{j=1}^n [f_j(x_j+tz_j) - f_j(x_j)]/t < \sum_{j:z_j>0} [\lambda_j^+(x_j)z_j - \varepsilon(z_j/s_j^+)]$$

$$+ \sum_{j:z_j<0} [\lambda_j^-(x_j)z_j - \varepsilon(z_j/s_j^-)]$$

This holds for $0 < t \le \bar{t}$ by the choice (15) of $\bar{t}$. Thus, invoking the inequality in (6), we get

$$[F(x+tz) - F(x)]/t < -\varepsilon\Big[\sum_{j:z_j>0} (z_j/s_j^+) + \sum_{j:z_j<0} (z_j/s_j^-)\Big] \tag{21}$$

$$\text{for} \quad 0 < t \le \bar{t}$$

In the degenerate case where $\bar{t} = \infty$, the ratios in (15) must all be $+\infty$, and each term in the sum on the right side of (21) must be 0. Then

$$[F(x+tz) - F(x)]/t < 0 \qquad \text{for} \quad 0 < t < \infty$$

and since $F(x+tz)$ is a convex function of t it must actually be a nonincreasing function of $t \in (0,\infty)$. Then (17) is correct, because the fact that z satisfies (6) implies by Theorem 1 that (8) holds.

In the remaining case, where $0 < \bar{t} < \infty$, $\bar{t}$ is equal to one of the positive ratios in (15) (the reciprocals of which appear in the sum of the right side of (21)), so that

$$[\sum_{j:z_j>0} (z_j/s_j^+) + \sum_{j:z_j<0} (z_j/s_j^-)] \geq 1/\bar{t} > 0 \tag{22}$$

Then

$$[F(x+tz) - F(x)]/t \leq -\varepsilon/\bar{t} < 0 \quad \text{for } 0 < t \leq \bar{t}$$

Again this tells us that $F(x+tz)$ is nonincreasing in t for $0 < t \leq \bar{t}$, and in particular, by setting $t = \bar{t}$, that (16) is correct. The proof of Theorem 2 is thereby completed.

Note that the degenerate case where $\bar{t}$ might be ∞ is impossible if the set of feasible solutions to (P) is bounded, or indeed even if the set of optimal solutions to (P) is nonempty and bounded. This is true because F is a closed, proper convex function on R^n. For any such function, if z is a vector such that, for some $x \in R^n$ with $F(x) < \infty$, $F(x+tz)$ is nonincreasing as a function of $t \in (0,\infty)$, then the latter must actually hold for *every* $x \in R^n$ (cf. [1,§8]).

Application to Dual Descent

For each of the functions f_j with associated finiteness interval C_j, there is a so-called conjugate convex function g_j on R with an associated finiteness interval D_j. We shall not go into the details of the symmetric relationship between the pair f_j, C_j and the pair g_j, D_j, and how one can easily be determined from the other. This is explained in [3], [4] and [1,§24].

The monotropic programming problem dual to (P) is

$$\text{maximize} \quad -G(v) = -\sum_{j=1}^{n} g_j(v_j) \quad \text{subject to} \tag{D}$$

$$v \in \mathcal{D} \quad \text{and} \quad v_j \in D_j \quad \text{for } j = 1,\ldots,n$$

This has exactly the same character as (P) (one could just as well minimize G as maximize $-G$). The role of problem (D) has not been mentioned here before, but in fact it underlies the theory of the descent algorithms for (P) as developed in [3]. In particular, as shown in [3], the following assertion can be added to our Theorem 1 above: *if (5) holds, then the* v *in (5) is approximately optimal for* (D) *in the sense that*

$$-G(v) \geq \sup\,(D) - \delta \quad \text{for } \delta = n\varepsilon \tag{23}$$

This extended version of Theorem 1 leads to a constructive proof [3] of the following duality theorem for monotropic programming.

THEOREM 3 [4]

If either (P) *or* (D) *has a feasible solution, then* inf (P) = sup (D).

From a practical point of view, one of the most interesting features of the duality is that it provides an alternative method of solving (P), where descent is applied to (D). To see how this can work, consider any feasible solution v to (D) and (for our fixed ε) the associated intervals

$$[\mu_j^-(v_j),\mu_j^+(v_j)] = \{x_j \in R \mid g_j(v_j+r) \geq g_j(v_j) + v_j r - \varepsilon, \ \forall r \in R\} \quad (24)$$

Assuming that none of these is empty (as is true if $\varepsilon > 0$ or if v is regularly feasible for (D)), we can determine constructively (by the sort of combinatorial algorithm mentioned earlier) whether

$$\exists x \in C \text{ satisfying } \mu_j^-(v_j) \leq x_j \leq \mu_j^+(v_j) \text{ for all } j \quad (25)$$

or instead

$$\exists w \in D \text{ satisfying } \sum_{j:w_j>0} \mu_j^+(v_j)w_j + \sum_{j:w_j<0} \mu_j^-(v_j)w_j < 0 \quad (26)$$

In the first case we conclude from the extended version of Theorem 1, with (P) and (D) in reversed roles, not only that v is δ-optimal for (D), but *also* x *is* δ*-optimal for* (P). In the second case, on the other hand, w gives a *direction of* ε*-ascent for* (D):

$$\exists t > 0 \text{ with } v+tw \text{ again feasible for (D)} \quad (27)$$
$$\text{but } -G(v+tw) > -G(v) + \varepsilon$$

So it is that, in general, the monotropic descent algorithms already discussed can be applied to the dual problem (D) (actually in terms of ascent, or equivalently in terms of minimizing G rather than maximizing $-G$), and, upon termination, can produce a δ-optimal solution to the primal problem (P).

The catch is that for $\delta = 0$ ($\varepsilon = 0$), termination cannot be guaranteed. Without termination, it is not clear what, if anything, has been learned about (P), even though an optimizing sequence may have been generated for (D). This is where the fortified version of the algorithm with $\delta > 0$ ($\varepsilon > 0$) takes on an obvious importance, since for it, termination is not only guaranteed, but the number of iterations obeys an *a priori* bound.

Even with the fortified algorithm applied in (D), however, there may in some situations be difficulty or inconvenience in having to deal with the conjugate functions g_j and intervals D_j. For instance, we may not wish to go through the exercise of constructing these functions and intervals from the given f_j's and C_j's and storing them in sufficient detail on a computer. Our aim now is to show that this is not really necessary, by virtue of the automatic step size rule in Theorem 2. *In fact, the fortified descent algorithm, as applied to* (D), *can be executed entirely in terms of the original problem* (P).

The automatic step size rule, as invoked for the dual problem, requires not only the values $\mu_j^+(v_j)$ and $\mu_j^-(v_j)$, but also certain associated values

$$\begin{aligned} r_j^+ &\in \arg\min_{r>0} [g_j(v_j+r) - g_j(v_j) - \varepsilon]/r \\ r_j^- &\in \arg\min_{r<0} [g_j(v_j+r) - g_j(v_j) - \varepsilon]/r \end{aligned} \quad (28)$$

(these formulas being extended by conventions parallel to (11), (12) and (13)).

Having determined a vector w as in (26), we can get improvement by at least ε in the feasible solution v simply by passing to $v+\bar{t}w$, where

$$\bar{t} = \min \begin{cases} r_j^+/w_j & \text{for } j \text{ such that } w_j > 0 \\ r_j^-/w_j & \text{for } j \text{ such that } w_j < 0 \end{cases} \tag{29}$$

The critical issue is just this, then: *given* v_j, *how can we determine* $\mu_j^+(v_j)$, $\mu_j^-(v_j)$ and also r_j^+ and r_j^- directly from f_j and C_j ? If this can be done, there is no need ever to look at g_j and D_j .

The answer is illustrated in Figure 2. The values $\mu_j^-(v_j)$ and $\mu_j^+(v_j)$ are simply the endpoints of a certain level set of f_j:

$$[\mu_j^-(v_j),\mu_j^+(v_j)] = \{x_j \in C_j \,|\, f_j(x_j) - x_j v \leq \inf_{\xi \in C_j} [f_j(\xi) - \xi v] + \varepsilon\} \tag{30}$$

The values r_j^+ and r_j^- are characterized by the relations

$$\begin{aligned} r_j^+ + v_j &\in \partial f_j(\mu_j^+(v_j)) = [f'_{j-}(\mu_j^+(v_j)),\ f'_{j+}(\mu_j^+(v_j))] \\ r_j^- + v_j &\in \partial f_j(\mu_j^-(v_j)) = [f'_{j-}(\mu_j^-(v_j)),\ f'_{j+}(\mu_j^-(v_j))] \end{aligned} \tag{31}$$

with obvious conventions to cover cases where $\mu_j^+(v_j) = \infty$ or $\mu_j^-(v_j) = -\infty$. These facts are elementary consequences of the theory of ε-subgradients of convex functions [1,§23].

Acknowledgement:

This research was sponsored in part by the Air Force Office of Scientific Research, Air Force Systems Command, U.S. Air Force, under grant no. F49620-82-K-0012 at the University of Washington, Seattle.

References

1. R.T. Rockafellar, Convex Analysis, Princeton University Press, 1970.

2. R.T. Rockafellar, Network Flows and Monotropic Optimization, Wiley Interscience, 1984.

3. R.T. Rockafellar, "Monotropic programming : Descent algorithms and duality", in Nonlinear Programming 4 (O.L. Mangasarian et al, editors), Academic Press, 1981, 327-366.

4. R.T. Rockafellar, "Convex programming and systems of elementary monotonic relations", J. Math. Anal. Appl., 19 (1967), 167-187.

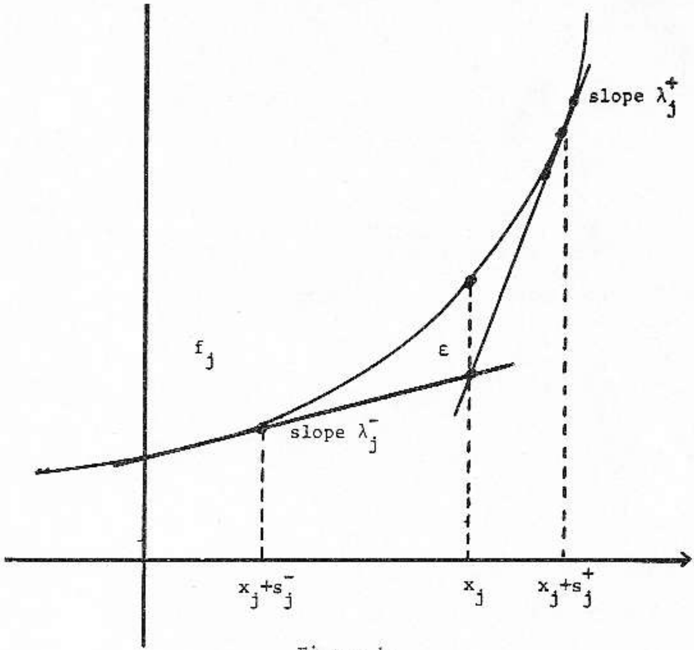

Figure 1.

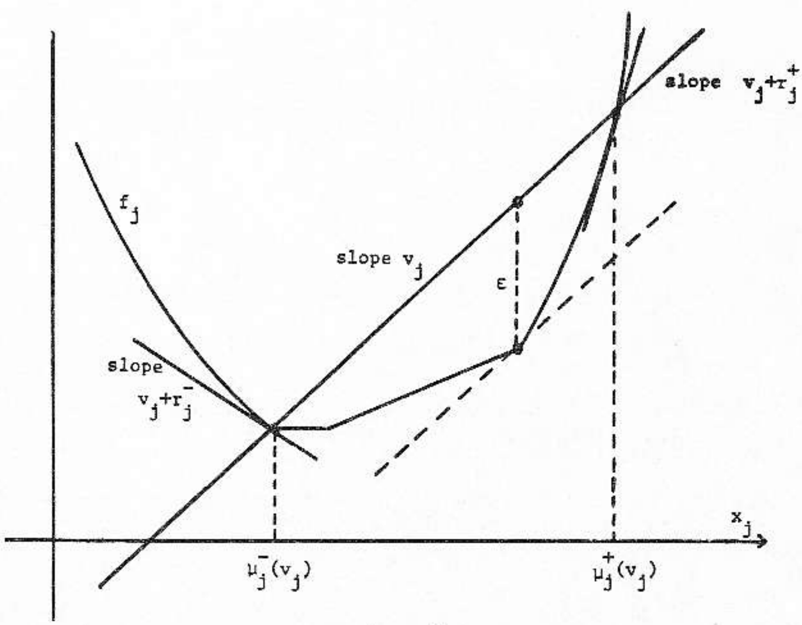

Figure 2.