ALGORITHMS FOR NONLINEAR PROBLEMS WHICH USE DISCRETE APPROXIMATIONS TO DERIVATIVES*

J.E. Dennis, Jr.+
Cornell University

71 - 98
Technical Report

Department of Computer Science
Cornell University
Ithaca, New York 14850

(*) Given at ACM'71 by invitation from SIGMAP

(+) Research supported by NSFGJ-27528

ALGORITHMS FOR NONLINEAR PROBLEMS WHICH USE DISCRETE APPROXIMATIONS TO DERIVATIVES

by

J.E. Dennis, Jr.*

ABSTRACT

The most desirable algorithms for nonlinear programming problems call for obtaining the gradient of the objective and the Jacobian of the constraint function. The analytic form is often impossible and almost always impractical to obtain. The usual expedient is to use difference quotients to approximate the partial derivatives. This paper is concerned with the theoretical and practical ramifications of such modifications to basic algorithms. Among the methods surveyed are steepest descent, Stewart's modifications of the Davidon-Fletcher-Powell method, the Levenberg-Marquardt method, Newton's method, and the nonlinear reduced gradient method. Numerical results are included in the presentation.

KEY WORDS AND PHRASES: Nonlinear function minimization, numerical differentiation, nonlinear programming.

CR CATEGORIES: 5.15,5.16,5.41

*Research supported by the National Science Foundation under NSF GJ-27528.

ALGORITHMS FOR NONLINEAR PROBLEMS WHICH USE DISCRETE APPROXIMATIONS TO DERIVATIVES

by

J.E. Dennis, Jr.

INTRODUCTION:

Let f and G be scalar and vector valued functions of a vector variable x. Consider the general nonlinear programming problem of the form:

Find a vector x^* such that each component of G is nonpositive and if x is any other vector with the same property then $f(x^*)$ does not exceed $f(x)$.

We are interested here only in the cases when f has no special properties except perhaps convexity. In particular, a problem is neither linear, quadratic, nor separable. Most of the desirable algorithms for the solution of this problem are based on the solution of successive linear problems and so they require the calculation of at least the gradient of f and the matrix of first derivatives, the Jacobian, of G. If, in addition, the Hessian, the matrix of second partials, of f is available then its use is often rewarded by a reduction in the number of approximate problems to be solved.

When faced with the job of finding a computational solution to the nonlinear problem, the analyst is usually willing to assume the existence of the derivatives in question but justifiably reluctant to sit down and crank them out by hand and then code them for the machine. It is also frequently the case that analytic expressions for the functions, much less the first or second partials,

are not available. This is perhaps more often so for the objective function then for the constraints and arises for example in the optimal control of chemical processes and in optimal design problems.

The obvious expedient is the use of difference quotients to replace the desired partial derivatives. Thus when the gradient of a function g is needed one chooses a nonzero vector h and forms the estimate

$$\frac{\partial g(x)}{\partial x_j}\Big|_{\bar{x}} \doteq \frac{g(\bar{x} + h_j \varepsilon_j) - g(\bar{x})}{h} ,$$

where ε_j is the vector whose i th coordinate is δ_{ij} (Kronecker delta). Let us denote such an approximate gradient at x by $\delta g(x;h)$ and an approximate Jacobian by $\Delta G(x;h)$. These, or related approximations hardly need justification when no analytic derivatives are obtainable. Even when they are, the work involved in making the analytic derivatives available for the machine sometimes causes use of the approximations to be more efficient if all else is equal. Recent work has shown that very often for the unconstrained problem, little or nothing is lost if the discrete form of an algorithm is used in conjunction with an intelligent choice of h.

The purpose of this paper is to survey these results for the unconstrained case in the hope that they will prove useful for the constrained problem. To this end we comment on their applications to obtain some elementary generalizations of Wolfe's reduced gradient method to the

case of nonlinear constraints. See Abadie and Carpentier [1] for priority.

In section 2 we present the case when no analytic partials are used at all. As one might suspect, even in the unconstrained problem it is possible to converge to a point which is not even a local minimum.

Section 3 is concerned with methods arising when one is willing to determine first partials analytically but not second partials.

The results of these two sections are related in section 4 via the reduced gradient method to the problem of optimization under nonlinear constraints. In section 5 we look at the significance of solving the discretized Kuhn-Tucker Saddle Point Problem which is the effect of the algorithm of section 4.

DESCENT ALONG DISCRETIZED GRADIENT DIRECTIONS

This section will be concerned with the most primitive of the methods considered. Suppose we wish to minimize a real function f of N real variables $x_1,\ldots,x_N$ subject to no constraints. We will assume throughout this section that f is continuously differentiable and so any minimum of f lies in the set $E = \{x = (x_1,\ldots,x_N)^T : \nabla f(x) = 0\}$. We will also assume that ∇f is unavailable to us and must be approximated by differences. At this point it should be mentioned that none of the methods of this section are really recommended unless the number of variables is fairly large (ten) and then only Stewart's method seems useful. For smaller problems see [17],[19],[20],[24].

It is reasonable to ask why a method is included if it isn't useful. The answer is that aside from a desire for completeness, a method is included because it is used by many people and successfully used by some.

The simple discrete analogue of steepest descent is the following:

Given x^k, choose $h^k = (h_1^k,\ldots,h_N^k)$ and form $\delta f(x^k,h^k)$. Now choose γ^k to minimize $f(x^k - \gamma^k \delta f(x_j^k\ h^k))$. Set $x^{k+1} = x^k - \gamma^k \delta f(x_j^k\ h^k)$.

Of course γ^k need not exist and even if it does γ^k is not computed but always estimated by the particular analyst's favorite technique. The most interesting vagary is the first one, "choose h^k". Is the strategy for choosing h^k important? If it is, what is a good one?

First let us convince ourselves of the importance of choosing h_i^k by some criterion other than that it be small relative to x_i^k.

Example 1

Let $f(x) = kx^T \begin{bmatrix} 1 & -1+.5r \\ -1+.5r & 1 \end{bmatrix} x$

where $k>0$ is arbitrary and r is the ratio such that $h^0 = |r\cdot x^0|$ $\delta f(x^0;r|x^0|)$ is easily shown to be zero if $x^0 = (a,a)$, $a<0$.

This was an extreme example since the iteration broke down before it started. Even when we get off to a good start, it we leave $h^k = h^0$ we can converge to a non minimum point.

Example 2

$$\text{Let } f(x) = x^T \begin{bmatrix} 2.6 & -2.4 \\ -2.4 & 2.6 \end{bmatrix} x$$

$h^0 = h^k = 10^{-3} \circ x^0$. $x^0 = (100,105)^T$. All computations were done in APL on Cornell's IBM360/65. The cosine of the angle between $\nabla f(x^0)$ and $\delta f(x^0;h^0)$ was .9999789715 indicating an excellent h^0. After 42 iterations the process was making excellent progress with a cosine of about .75 and $x^{42} = (.3467820816,.3864009953)$. However $x^{50} = -(.000838720661, .004900663455)$ and the cosine was $-3.701471096 \times 10^{-7}$. The iteration seems to have converged to about 6 places, with an f value of $4.21408585 \times 10^{-5}$.

It is interesting to note that true steepest descent gave $x^{50} = (.6243654846,.5594854005)$ and $f(x^{50}) = .1193714638$. On the other hand, the Euler-trapezoidal predicator-corrector pair with a unit step size $x^{33} = (.004891909778,.004990987333)$ and $f(x^{33}) = 7.30071098 \times 10^{-6}$ when applied to the initial value problem $x'(t) = -\delta f(x,h^0), x(0) = x^0$. In fact $x^{48} = (5.333697222,5.446809675) \times 10^{-5}$, $f(x^{48}) = 8.687145399 \times 10^{-10}$ and the decrease was steady.

These examples illustrate the general results obtained in [2]. Those results, based on an analysis of the asymptotic stability of the initial value problem say that even if f is convex and has Lipschitz continuous second partial derivatives we cannot expect to do better than

$$||x^n - x^*|| \leq \tfrac{1}{2}k\,(H(x^*))\,||h^0||,$$

where $k(H(x^*))$ is the condition number of the Hessian of

f at x^*. This result is independent of n and $||h^0|| > 0$ $||x^* - x^0|| > 0$. In Example 1 this bound is $(2-.5r)||x^0||$ which is off by a factor of $(2-.5r)$. For Example 2 the bound is rather pessimistic at about 1.65 while $||x^{50}-x^*||$ is about 5×10^{-3}.

The bound does tell us that if we decrease h on the iteration proceeds, we will get convergence.

Now, of course anyone using the discretized steepest descent algorithm would expect to have to decrease $||h^k||$ with k. The problem is to decrease it quickly enough so that convergence is not impeded. As $||h^k||$ is decreased the errors in the calculation of $\delta f(x^k;h^k)$ become larger so that they eventually swamp the iteration. One answer to this impass might be to use a higher order analogue of $\delta f(x^k;h^k)$ but this only postpones the problem at the expense of additional function evaluations in the gradient calculations. See [22].

One possible strategy was developed in [6] for the discrete Levenberg-Marquardt and Gauss analogues.

Set $h_j^n = \min\{|f(x_n)|, \delta_j^n\}$ where

$$\delta_j^n = \begin{cases} 10^{-9} \text{ if } |x_j^n| < 10^{-6} \\ \\ 10^{-3} \times |x_j^n| \text{ if } 10^{-6} \leq |x_j^n|. \end{cases}$$

Example 3 The above choice of h was tried in conjunction with Example 2. The reader will easily see that h^0 is the same in both cases since $f(100,105) = 3162.5$. The iteration had the dubious distinction of being as bad as true steepest descent. Of more interest to us is the fact that in 57 itera-

tions the cosine of the angle between $\nabla f(x^k)$ and $\delta f(x^k;h^k)$ was .9999815959 and had been constant in five decimal places throughout.

An extremely interesting method which uses a discrete gradient has been proposed by Stewart [22]. The idea of the method is rather uninteresting since it is simply to use the Davidon-Fletcher-Powell Hessian approximation on $\delta f(x^k;h^k)$. The real interest is in the fact that the clever h-strategy devised by Stewart works so well.

Suppose that we have $\delta f(x^k;h^k)$, H_k, an approximation to $H(x^k)$ and by some method we have determined x^{k+1}. Stewart sets $x^{k+1} = x^k - \gamma^k H_k^{-1}\delta f(x^k;h^k)$ where γ^k is chosen by a linear search routine to minimize

$$f(x^k - \gamma H_k^{-1}\delta f(x^k;h^k))$$

but this is irrelevant to the choice of h^{k+1}. Now consider the equation

$$f(x^{k+1}+h_i^{k+1}\varepsilon_i) = f(x^{k+1})+\nabla f(x^{k+1})h_i^{k+1}\varepsilon_i+\frac{1}{2}(h_i^{k+1}\varepsilon_i)^T H_k(h_i^{k+1}\varepsilon_i)$$

$$= f(x^{k+1}+h_i^{k+1}\frac{\partial f(x^{k+1})}{\partial x_i} + \frac{1}{2}(h_i^{k+1})^2(H_k)_{ii}.$$

If h_i^{k+1} could be determined by this equation then

$$\frac{(\delta f(x^{k+1};h^{k+1}))_i - \frac{\partial f(x^{k+1})}{\partial x_i}}{\frac{\partial f(x^{k+1})}{\partial x_i}} = \frac{h_i^{k+1}(H_k)_{ii}}{2\frac{\partial f(x^{k+1})}{\partial x_i}}$$

would be the relative error in $(\delta f(x^{k+1};h^{k+1}))_i$ due to the inadequacy in the quadratic approximation to $f(x^{k+1}+h_i^{k+1}\varepsilon_i)$. Stewart estimates the error in calculation

of $(\delta f(x^{k+1};h^{k+1}))_i$ as coming from cancellation in $f(x^{k+1}+h_i^{k+1}\varepsilon_i) - f(x^{k+1})$. Assuming that for given x, the machine value $f(x)^* = f(x)(1+\varepsilon(x))$ where $|\varepsilon(x)| \leq \eta$ and that $f(x^{k+1}+h_i^{k+1}\varepsilon_i) \doteq f(x^{k+1})$, we get $(\delta f(x^{k+1};h^{k+1}))_i^* - (\delta f(x^{k+1};h^{k+1}))_i = 2\varepsilon f(x^{k+1})/h_i^{k+1}$. Hence the relative error in the machine calculation of $(\delta f(x^{k+1};h^{k+1}))_i$ is bounded in absolute value by

$$2\eta \left| \frac{f(x^{k+1})}{(\delta f(x^{k+1};h^{k+1}))_i h_i^{k+1}} \right| .$$

Notice that the truncation error increases with increasing h while the machine error decreases. Stewart sets the absolute values of the two errors equal and determines h_i^{k+1} from the equations which result when $(\delta f(x^{k+1};h^{k+1}))_i$ and $\frac{\partial f(x^{k+1})}{\partial x_i}$ are estimated. Our bounds are in a slightly different form than his but if we make the estimation $(\delta f(x^{k+1};h^{k+1}))_i$ are $\frac{\partial f(x^{k+1})}{\partial x_i}$ we get

$$h_i^{k+1} = 2\left[\eta \left| \frac{\delta(x^{k+1})}{(H_k)_{ii}} \right|\right]^{\frac{1}{2}}, \text{ one of his choices.}$$

The details of choosing h^{k+1} can be found in [22] along with some impressive numerical results. Lill [14] has published an Algol program implementing the method with a more sophisticated linear minimization routine than Stewart used. The author believes that Stewart's h-strategy is far too interesting to be allowed to accompany the Davidon-Fletcher-Powell method into absolescence.

The last two methods to cover in this section are generally associated with the nonlinear least squares problem,

ie. $f(x) = ||F(x)||_2^2$ where $F:E^N \to E^M$, $M \geq N$. Let $J_F(x) = (\frac{\partial f_i}{\partial x_j}(x))$ and it is easily seen that $\nabla f(x) = 2J_F(x)^T F(x)$. An excellent iteration for finding stationary points is the so-called Levenberg-Marquardt method [13],[16].

$$x^{k+1} = x^k - (\alpha^k + J_F(x^k)^T J_F(x^k))^{-1} J_F(x^k)^T F(x^k).$$

α^k is a nonnegative scalar chosen by a descent criterion or to improve the conditioning of the linear system which must be solved to get $x^{k+1} - x^k$. If the scalar α^k is always zero, the iteration is called the Gauss-Newton method. Our interest is in the iterations which result from using $\Delta F(x^k, h^k)$ in place of $J_F(x^k)$ in the Levenberg and Gauss algorithms.

Once again a result in [2] based on asymptotic stability theory for ordinary differential equations warns us that we must decrease h^k. In [6] several convergence theorems are given for these processes. We chose to summarize the results informally. Assume that J_F satisfies an ℓ_2 Lipschitz condition of order 1 with constant K and let x^* be a zero of $J_F(x^*)^T F(x^*)$ with $K||F(x^*)||_2$ a strict lower bound on the spectrum of $J_F(x^*)^T J_F(x^*)$. Then the discrete Levenberg-Marquardt algorithm converges locally as long as $\{h^k\} \to_k 0$. An interesting special case is when $F(x^*) = 0$, ie. the fit is exact. Then if $\{\alpha^n\}$ and $\{h^n\}$ are $0(||F(x_n)||_{any})$ the method is second order. The results form the basis for an (α, h) - strategy.

Let δ^k be defined as above Example 3.

Set $\alpha^n = c \cdot ||F(x^n)||_\infty$ where

$$c = \begin{cases} 10 \text{ whenever } 10 \leq ||F(x^n)||_\infty \\ 1 \text{ whenever } 1 < ||F(x^n)|| < 10 \\ .01 \text{ when } ||F(x_n)|| \leq 1. \end{cases}$$

Set $h_i^n = \min \{||F(x_n)||_{any}, \delta_i^n\}$.

[6] contains test results which indicate the excellence of these strategies. There seems to be no advantage in using analytic derivatives for these methods. When $||F(x^*)|| = 0$ the method does indeed exhibit second order convergence.

DISCRETE NEWTON METHODS

In this section we will assume that ∇f is available to us but $H(x) = (\frac{\partial f(x)}{\partial x_i \partial x_j})$, the Hessian of f at x is not. The two methods considered here both attempt to speed convergence by using discrete approximations to second partials of f but they are quite dissimiliar.

Goldstein and Price [12] considered the straight-forward discrete analogue of Newton's method for unconstrained minimization, ie

$$\Delta\nabla f(x^k;h^k)\ d^k = \nabla f(x^k)$$

$$x^{k+1} = x^k - \gamma^k d^k.$$

γ^k was chosen to insure $f(x^{k+1}) < f(x^k)$, in particular by the technique [11],[18] usually associated with Goldstein which ensures that eventually $\gamma^k = 1$. Our interest is in the choice of h^k. $||h^k||$ was chosen $0(||\nabla f(x^k)||)$ and the method was shown to be super-linear under the assumption that f be strictly convex and continuously second differentable.

The authors furthermore remark that if f is C^3, the method is second order. Samanskii [21] at about the same time gave a Kantorovich Theorem for the more general problem of finding zero's of a nonlinear vector valued function F. He showed, if F is C^2, and $||h^k||$ chosen $0(||F(x^k)||)$, the discrete Newton method has its true error vectors bounded by a second order sequence of error bounds. (Ortega and Rheinboldt [18] would call this "of r-order 2"). These two papers seem to be the first to suggest this "Steffensen" strategy for choosing h. The author in [9] gave a Kantorovich Theorem and a proof of "true"(q-order 2, [18]) second order convergence under the hypothesis that the first partials of F are Lipschitz continuous.

Now in the case of a general F, this discretized Newton method requires N scalar function evaluations for $F(x^k)$ and N^2 more for $\Delta F(x^k,h^k)$. In the case where $F=\nabla f$, we could assume symmetry of $H(x^k)$ and use only $(N^2+N)/2$ scalar function values for $\Delta\nabla f(x^k,h^k)$ or a total of $(N^2+3N)/2$ as opposed to N^2+N for the non symmetric case. Brown's method, the other method to be considered in this section requires only $(N^2+3N)/2$ function values in either case besides having some other desirable qualities.

Brown's method is somewhat difficult to describe in general so we will illustrate it by giving the case of three nonlinear equations in three unknowns in complete detail.

Let $F = (f_1,f_2,f_3)^T$ and let x^k be given. Then in the usual fashion, we set

$$0 = f_1(x^k) + \sum_{j=1}^{3} f_{1j}(x^k)(x_j - x_j^k).$$

A basic idea of Brown's method is that this equation is not so much viewed as a linearization of $F(x) = 0$ as it is as a linearization of the relationship between the components of the solution vector. Thus we solve this linear equation for some x_{j_0} where j_0 is chosen such that $f_{1j_0}(x^k)$ is the component of $\nabla f_1(x^k)$ which has maximum modulus. Let us assume that $j_0 = 3$. The above equation now yields

$$x_3 = x_3^k - \frac{f_1(x^k)}{f_{13}(x^k)} - \sum_{j=1}^{2} \frac{f_{1j}(x^k)}{f_{13}(x^k)} (x_j - x_j^k).$$

Now define $\hat{f}_2^{(k)}(x_1,x_2)$ by setting it equal to $f_2(x_1,x_2,x_3(x_1,x_2))$ where the bivariate function x_3 is defined above. Now set

$$0 = \hat{f}_2^{(k)}(x_1^k,x_2^k) + \sum_{j=1}^{2} \hat{f}_{2j}^{(k)}(x_1^k,x_2^k)(x_j - x_j^k)$$

and under the assumption that $f_{2,2}^{(k)}(x^k)$ is the element of the 2-vector $\nabla \hat{f}_2^{(k)}(x^k)$ which has maximum modulus we obtain

$$x_2(x_1) = x_2^k - \frac{\hat{f}_2^{(k)}(x_1^k,x_2^k)}{\hat{f}_{2,2}^{(k)}(x_1^k,x_2^k)} - \frac{\hat{f}_{21}^{(k)}(x_1^k,x_2^k)}{\hat{f}_{2,2}^{(k)}(x_1^k,x_2^k)} (x_1 - x_1^k)$$

Now define $\hat{f}_3(x_1) = f_3(x_1, x_2(x_1) x_3(x_1, x_2(x_1)))$ set $0 = \hat{f}_3(x_1^k) + \hat{f}_{31}(x_1^k)(x_1 - x_1^k)$ and solve for $x_1^{k+1} = x_1$. Then $x_2^{k+1} = x_2(x_1^{k+1})$, $x_3^{k+1} = x_3(x_1^{k+1}, x_2^{k+1})$. Clearly we could have incorporated descent in each component separately or in the conventional way. So far we have been describing the analytic form of Brown's method. The discrete form is so far superior that one generally calls it "Brown's method" and specifies "analytic form of Brown's method" when he wants true derivatives - just the reverse of the usual naming procedure.

Let us look more closely at the partials used.

$$\hat{f}_{2j}^{(k)}(x_1^k, x_2^k) = \frac{\partial f_2}{\partial x_j}(x_1^k, x_2^k, x_3(x_1^k, x_2^k))$$

$$= f_{2,j}(x_1^k, x_2^k, x_3(x_1^k, x_2^k)) +$$

$$+ f_{2,3}(x_1^k, x_2^k, x_3(x_1^k, x_2^k)) \frac{\partial x_3}{\partial x_j}(x_1^k, x_2^k) .$$

The partial of x_3 with respect to x_j is just $\frac{f_{1j}(x^k)}{f_{13}(x^k)}$ but notice that in this analytic form, although we have $f_{1,2}(x^k)$ and in the minimization problem $f_{12} = f_{21}$, f_{21} is needed with a different argument. Hence symmetry doesn't help us to the extent it does in Newton's method. On the other hand, for say $j=2$,

$$\hat{f}_2^{(k)}(x_1^k, x_2^k + h_2^k) - \hat{f}_2^{(k)}(x_1^k, x_2^k) = f_2(x_1^k, x_2^k + h_2^k, x_3^k - \frac{f_1(x^k)}{f_{13}(x^k)} -$$

$$h_2^k \frac{f_{12}(x^k)}{f_{13}(x^k)}) - f_2(x_1^k, x_2^k, x_3^k - \frac{f_1(x^k)}{f_{13}(x^k)})$$

and, of course $f_{12}(x^k)$, $f_{13}(x^k)$ would have been approximated by difference quotients. Thus it follows that whether or not $(f_{ij}(x))$ is symmetric, $x^{k+1} - x^k$ requires only $(N^2 + 3N)/2$ scalar function values. See [7] for a proof that the method is second order when h is chosen by the Steffensen strategy.

EXAMPLE 4

Rosenbrock's function is the sum of squares function for

$$f_1(x_1,x_2) = 10(x_2 - x_1^2)$$

$$f_2(x_1,x_2) = 1 - x_1.$$

A standard problem is to find $x^* = (1,1)$ from $x^0=(-1.2,1)$. The discrete Newton's method [5] required 24 function values to reduce $||F||_2$ to 7.451×10^{-8}. Brown's method with $h_j^k = x_j^k \times 10^{-3}$ required 15 function values to reduce $||F||_2$ to zero. Thus Brown's method not only required 5 function values per iteration as opposed to 6 for Newton but with a rather naive h-strategy it required fewer iterations, 3 as opposed to 4. The method always behaves this way when some of the equations are linear and these are placed first in the order. The reason is probably that if M of the N equations are linear then Brown's method essentially reduces the problem to an $(N-M) \times (N-M)$ nonlinear system. If all the equations are linear then the method reduces to Gaussian elimination with row pivoting. It is this exploitation of any linearity in the problem that makes the method of special interest to us here. See [3] in

this context, especially Example 7.1 and for an Algol program with the Steffenson strategy see [4].

THE REDUCED GRADIENT METHOD FOR NONLINEAR CONSTRAINTS

Here we will consider the general problem:

minimize $f(x)$ for $x \in E^N$

subject to $G(x) \leq 0 \in E^M$. $M > N$

For simplicity we will assume nondegeneracy; for any x, $J_G(x)$ is an MxN matrix whose rows satisfy the Haar condition and that J_G is Lipschitz continuous.

We will assume a familiarity with Wolfe's reduced gradient method [23] and present a special case of Abadie and Carpentier's generalization to nonlinear problems [1]. The idea is to find the binding constraints for the problem and so in a standard way we introduce a variable for each constraint. The equality constraints are now $G(x) + y = 0$, and the M new constraints are $y \geq 0$. Pick a feasible point x^0 and compute $y^0 = G(x^0)$. Now partition y^0 into $y^0_A \oplus y^0_B$, independent and dependent variables, where $y^0_A \in E^N$ contains every zero component of y^0.

The constraints can be linearized as

$$G(x^0) + G'(x^0)(x - x^0) = -y$$

$$\text{or } G'(x^0)(x - x^0) = -(y - y^0).$$

If we partition

$$G'(x^0) = G'(x^0)_A \oplus G'(x^0)_B$$

we get $(x - x^0) = - G'(x^0)^{-1}_A (y - y^0)_A$

$$(y - y^0)_B = G'(x^0)_B \, G'(x^0)^{-1}_A (y - y^0)_A.$$

Hence $\nabla_{y_0} f(x^0) = \nabla_x f(x) \nabla_{y_0} x \mid x^0 = - \nabla_x f(x^0) J_G(x^0)^{-1}_A$.

$\nabla_{y_0} f(x^0)$ is called the "reduced gradient".

The basic procedure now defines

$\Delta^0_{y_A}$ by $(\Delta y^0{}_A)_j = - (\nabla_{y_A} f(x^0))_j$;

if $(\nabla_{y_0} f(x^0))_j < 0$ or $(y^0_A)_j > 0$ and

$(\Delta^0_{y_A})_j = 0$, otherwise.

In order to prevent "jamming," this step is often modified, for example so that Δy^0_A has at the most one nonzero component.

Suppose $\Delta y^0_A = 0$, then $y^0_A = 0$ and all the constraints $G(x^0)_A$ are binding. But $\nabla_{y_A} f(x^0) \geq 0$ so we must decrease at least one of the $(y_A)_j$ in order to move in the direction $-\nabla y_A f(x^0)$, ie, we must violate a constraint in order to decrease f so we are done.

If $\Delta y^0_A \neq 0$, compute $\Delta x^0 = -G'(x_0)^{-1}_A \Delta y^0_A$ and $\Delta y^0_B = -G'(x^0)_B G'(x^0)^{-1}_A \Delta y^0_A = -G'(x^0)_B \Delta x^0$. Now compute Θ_A, Θ_B to be the largest positive numbers such that

$$y^0_A + \Theta_A \Delta y^0_A \geq 0$$

$$y^0_B + \Theta_B \Delta y^0_B \geq 0 .$$

Maximize $f(x^0 + t\Delta x^0)$ where $t \in (0, \bar{\Theta}]$, $\bar{\Theta} = \min\{\Theta_A, \Theta_B\}$.

Let $\bar{t}$ be the minimizing t. Set $y^1_A = y^0_A + \bar{t}\Delta y^0_A$ and use Newton's method with initial guess

$$x = x^0 + \bar{t}\Delta x^0, \; y_B = y^0_B + \bar{t}\Delta y^0_B$$

to solve the nonlinear system

$$G(x) + y^1_A + y_B = 0 \quad \text{for} \quad x^1 \quad \text{and} \quad y^1_B.$$

If any $y^1_B = 0$, interchange that coordinate into y_A and one from y_A back into y_B. If no $y^1_B = 0$, then proceed.

We have suppressed a great deal of essential detail, some of which can be found in [1] along with testimonials as to the utility of the method. See also [8] and [10] for recommendations of the method.

The most obvious way to tie in the results of the two previous sections with the reduced gradient method would be to replace all the deivatives by differences and let the reduced gradient be

$$\delta f(x^0;h^0)\Delta G(x^0;h^0)_A^{-1}.$$

This raises the question as to what the significance of $\Delta y_A^0 = 0$ would be. We postpone consideration until section 5 since it is a question of having solved some problem other than the Kuhn-Tucker saddle point problem. Even if we want to continue to use analytic derivatives in order to be sure we solve the Kuhn-Tucker problem, Brown's method should be very useful in solving the system

$$G(x) + y_A^1 + y_B = 0$$

for x^1 and y_B^1. Here, we are trying to solve M nonlinear equations in M unknowns in order to get a feasible x^1,y^1. Furthermore it seems quite possible that some of the constraints in G might be linear. Brown's method could take advantage of this fact as noted in the previous section.

These modifications of the reduced gradient method are all based on trying to do less work per iteration. We finish this section by suggesting a method which would involve more work per iteration but perhaps fewer iterations. First use Brown's method with the obvious modifications to solve for $\nabla_{y^0} f(x^0)$. Define Δy_A^0 as before and use

Brown's method to find Δx^0. The idea here is that two systems mentioned really represent a linearization of relationships between variables and so Brown's method should give more accurate relationships which utilize any existing linearity of the problem. Clearly there is much testing to be done before more is said.

A DISCRETIZED KUHN-TUCKER STATIONARY-POINT PROBLEM

In the last section the nonlinear reduced gradient method was terminated when $\Delta y_A^0 = 0$. This is because $x^0, (\nabla_y f(x^0) | \theta_{M-N})$ in the analytic case solve the Kuhn-Tucker stationary point problem. We will show that in the discrete form of the method, the same halting condition signals the solution of a discrete analogue of the problem and that this solution has certain minimization properties.

Let us suppose that we have reached a point x^0, y^0 such that $\Delta y_A^0 = 0$ and

$$(\Delta y_A^0)_j = \begin{cases} -(\nabla_{y_A} f(x^0))_j, \text{ if } (\nabla_{y_A} f(x^0)_j) < \text{ or } (y_A^0)_j > 0 \\ \\ 0 \text{ , otherwise} \end{cases}$$

with $\nabla_{y_A} f(x^0) = -\delta f(x^0; h^f) \Delta G(x^0; h^G)_A^{-1}$. By reordering if necessary we can assume that $(y_1^1, \ldots, y_N^1) = y_A^1$. Set $u^T = (\nabla_{y_A} f(x^0) | \theta_{M-N})^T \in E^M$.

Now by the definition of Δy_A^0, we see that it can be zero only if for every j between 1 and N, $\nabla_{y_A} f(x^0)_j \geq 0$ and $(y^0_A)_j = 0$. Hence $u \geq 0$ and

$u^T G(x^0) = 0$. x^0 is feasible and so $G(x^0) \leq 0$. Furthermore, $\delta f(x^0;h^f) + u^T \Delta G(x^0,h^G) = \delta f(x^0;h^f) + \nabla_{y_A} f(x^0)\Delta G(x^0;h^G)_A = \delta f(x^0;h^f) - \delta f(x^0;h^f)\Delta G(x^0,h^G)_A^{-1}\Delta G(x^0,h^G)_A$.

$= 0$.

Of course it is well known [15] that in the analytic case with f and G convex, x^0 is a minimum for the nonlinear programming problem. Let us assume that $h^G \geq 0$ and $h^f \geq 0$. In this case we will show that x^0 has the property that if $x^0 + h^G_j \varepsilon_j$ is feasible then $f(x^0) \leq f(x^0 + h^f_j \varepsilon_j)$. If, in addition, f is convex, then $f(x^0) \leq f(x^0 + t\varepsilon_j)$ for every $t \geq h^f_j$. Let j_0 be arbitrary between 1 and N such that $x^0 + h^G_{j_0}\varepsilon_{j_0}$ is feasible. Then

$$\frac{f(x^0+h^f_{j_0}\varepsilon_{j_0})-f(x^0)}{h^f_{j_0}} = -\sum_{i=1}^{M} u_i \frac{g_i(x^0+h^G_{j_0}\varepsilon_{j_0})-g_i(x^0)}{h^G_{j_0}}$$

Now either $u_i = 0$ or $g_i(x^0)$ is so $f(x^0+h^f_{j_0}\varepsilon_{j_0})-f(x^0)$

$$= \sum_{i=1}^{M} \left(\frac{h^f_{j_0}}{h^G_{j_0}} u_i\right)(-g_i(x^0 + h^0_{j0}\varepsilon_{j0})) \geq 0, \text{ since}$$

$G(x^0+h_{j_0}\varepsilon_{j_0}) \leq 0$. Suppose now that f is convex and $t \geq h^f_{j_0}$. Then

$$\left(\frac{h^f_{j_0}}{t}\right) f(x^0 + t\varepsilon_{j0}) + \left(1 - \frac{h^f_{j_0}}{t}\right) f(x^0)$$

$$\geq f(x^0 \pm h^f_{j_0}\varepsilon_{j_0})$$

$$\text{or} \quad \left(\frac{h^f_{j_0}}{t}\right) f(x^0 + t\varepsilon_{j_0}) \geq \left(\frac{h^f_{j_0}}{t}\right) f(x^0)$$

since $f(x^0 + h^f_{j_0} \varepsilon_{j_0}) \geq f(x^0)$.

Hence $f(x^0 + t\varepsilon_{j_0}) \geq f(x^0)$.

Notice that we made no assumptions about G.

This result gives us a guide as to how small we need ultimately to make $||h^G||$ and $||h^f||$. $||h^f||$ needs to get small enough so that we aren't concerned about f having a smaller value at any feasible point closer to x^0 than $||h^f||$ along a coordinate direction. $||h^G||$ needs to be small enough to allow the result to apply in any little bubble on the set of feasible points which is too large for us to want to disregard.

There is really nothing special about the set of directions $\varepsilon_1, \ldots \varepsilon_N$. We could have used any other, possibly larger set of directions. One strategy might be to solve the problem on $\varepsilon_1, \ldots, \varepsilon_N$ and then on say $\bar{\varepsilon}_1, \ldots, \bar{\varepsilon}_N$ where $\bar{\varepsilon}_i$ is a "45^0 rotation" of ε_i. The previous answer should provide an excellent initial guess.

REFERENCES

[1] Abadie,J. and Carpentier,J., "Generalization of the Wolfe reduced gradient method to the case of nonlinear constraints", in Optimization edited by R. Fletcher, Academic Press, London 1969.

[2] Boggs,P.T. and Dennis,J.E., "Error bounds on the discretized steepest descent and the discretized Levenberg-Marquardt algorithms", to appear.

[3] Brown,K.M., "A quadratically convergent Newton-like method based on Gaussian elimination", SIAM J. Numer Anal 6, p 560, 1969.

[4] Brown,K.M., "Solution of simultaneous nonlinear equations II", to appear.

[5] Brown,K.M.,and Conte,S.D. "The solution of simultaneous nonlinear equations", Proc. 22nd Nat. Conf Assoc Comp Mach. p 111 Thompson Book Co., Washington, D.C., 1967.

[6] Brown,K.M., and Dennis, J.E., "Derivative free analogues of the Levenberg-Marquardt and Gauss algorithms for nonlinear least squares approximation", IBM Philadelphia Scientific Center Tech. Rpt. No. 320-2994, 1970.

[7] Brown,K.M. and Dennis,J.E., "On the second order convergence of Brown's method for solving simultaneous nonlinear equations", to appear.

[8] Colville,A.R., "A comparative study of nonlinear programming codes", Proc. Princeton Symp. on Math. Prog. - edited by H.W. Kuhn 1970.

[9] Dennis,J.E., "On the convergence of Newton-like methods", Numerical Methods for Nonlinear Algebraic Equations edited by P. Rabinowitz, Gordon-Breach, London, 1970.

[10] Fletcher,R., "Methods for the solution of optimization problems", United Kingdom Atomic Energy Research Establishment T.P. 432.

[11] Goldstein,A.A., Constructive Real Analysis, Harper-Row, New York, 1977.

[12] Goldstein,A.A. and Price,J., "An effective algorithm for minimization", Numer. Math. 10, p 184, 1967.

[13] Levenberg,K., "A method for the solution of certain nonlinear problems in least squares, Quart. Appl. Math, 2, p 164, 1944.

[14] Lill,S., "A modified Davidon method for finding the minimum of a function using difference approximations for derivatives", Computer J. 13, p 111, 1970.

[15] Mangasarian,O.L., Nonlinear Programming, McGraw-Hill, New York, 1969.

[16] Marquardt,D.W., "An algorithm for least squares estimation of nonlinear parameters", SIAM J. Appl. Math 11, p 431, 1963.

[17] Nelder,J.A. and Mead,R., "A simplex method for function minimization", Computer J. 7, p 308, 1965.

[18] Ortega,J.M. and Rheinboldt,W.C. Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

[19] Powell,M.J.D., "An efficient method of finding the minimum of a function of several variables without calculating derivatives", Computer J. 7, p 155, 1964.

[20] Powell,M.J.D., "Recent advances in unconstrained optimization , United Kingdom Atomic Energy Research Establishment T.P. 430.

[21] Samanskii,V.E., "On a modification of the Newton method" (Russian) Ukrain. Math. Z. 19, p 133, 1967.

[22] Stewart,G.W., "A modification of Davidon's minimization method to accept difference approximations of derivatives", J.A.C.M., 14, p 72, 1967.

[23] Wolfe,P., "Methods of nonlinear programming", Nonlinear Programming edited by J. Abadie, John Wiley & Sons, New York, 1967.

[24] Zangwill,W.I., "Minimizing a function without calculating derivatives", Computer J., 10, p 293, 1967.