A family of gradient methods for optimization

J. O. Ramsay

McGill University, Montreal, Canada

The path leading to the optimum can be defined as a system of differential equations. The application of ODE solving methods to optimization and some convergence acceleration possibilities are discussed. This approach seems to have promise for the initial stages of difficult optimization problems and some nonlinear programming tasks.

(Received December 1969)

1. Introduction

The approaches to the problem of maximising or minimising a nonlinear function of several variables can be put into three categories according to how much information is extracted from the function.

1. Only the function value. The more well-known methods in this category are those of Nelder and Mead (1965), Powell (1965), Rosenbrock (1960), and Swann (1964).

2. The gradient. The main technique in this category is the method of steepest ascent, also known as the 'gradient method'.

3. The matrix of second derivatives. This matrix may be computed directly, as in the method of successive approximations or Newton–Raphson method, or may be approximated as in the methods of Barnes (1965), Fletcher and Powell (1963), Fletcher and Reeves (1964), and Powell (1965).

The advantages and disadvantages of these methods have been discussed in detail elsewhere (for example, Box, 1966, Wilde and Beightler, 1967). The methods in the first category are especially useful when computed values are subject to error since the function value generally is less unstable than derivatives of the first or higher order. They are also essential where computation of values other than the function value is impractical or impossible. Otherwise, the slow convergence and sensitivity to sudden changes in slope of the surface make these methods generally inferior to those in the other two categories.

For the majority of unconstrained optimization problems, the methods in category three will be easily superior to the gradient methods. Their convergence for nearly quadratic functions is very rapid and they provide valuable information about the curvature at the optimum. Even for severely non-quadratic functions, the availability of good initial approximations to the solution can often insure that these methods will come up with the final solution quickly.

This paper is concerned with those functions which may cause difficulties in the use of second derivative approaches. The presence of sharp ridges may make round-off error in the computer a severe problem with these methods and sudden changes in direction of the ridge may make their convergence slow since the ridges of quadratic functions, for which these methods are usually convergent, are straight. On the other hand, the function might be well behaved but there may be simply too many variables to allow convenient storage of the matrix of second derivatives. Finally, there is the large class of problems for which constraints on the variables exist. Methods which rely on finding the optimum along a specified line in parameter space will tend to strike the boundaries of the admissible region more often than those which take small steps. Transformations of the kind suggested by Box (1966) imply that once a constraint has been tightened in this way, it cannot be loosened again except by some form of external intervention.

For these problems, the gradient method is very useful and references on nonlinear programming treat it as a widely applicable method for problems of this type (Hadley, 1964). This paper defines a class of methods which fall into the second category. They do not attempt to compute or approximate second derivatives and do not require the location of an optimum on a line. They contain the gradient method as a special case and it will be shown that in general the gradient method is not an efficient member of this class. The important advantages which some other members in this class possess will be outlined in the next section. Section 3 discusses some convergence accelerating possibilities. Section 4 provides an example of a function for which these methods are particularly appropriate.

2. The gradient path methods

The rationale behind the gradient method is that if one moves in the direction in which the function is increasing most rapidly, and if one's step sizes are small enough, then one can guarantee that the function is always being increased and will eventually reach a point where it cannot be increased any further, assuming that a maximum exists. In the limit as step sizes get smaller but more numerous, a continuous path is defined which begins at some initial point in the parameter space and passes through the maximum.

This path can be characterised in the following way:

Let the initial point on the path be p. For an arbitrary point, x, which is on the path, let the gradient be $s(x)$. Now there is an arc length along the path from the initial point, p, to the arbitrary point, x, which can be called r. It is then possible to define the point, x, as a function of arc length. That is,

$$x = y(r) \tag{1}$$

Naturally, this implies that

$$y(0) = p \tag{2}$$

Since the path is defined in such a way that its tangent, $y'(r)$, lies along the gradient vector, and since it is also true that the tangent of any path defined as a function of arc length has unit length, it follows that

$$y'(r) = \frac{s(y)}{||s(y)||} \tag{3}$$

Thus, the path is characterised by the set of ordinary differential equations (3), and the initial value condition (2).

The essence of the method is simple. Since it is successive points on this gradient path that one is after and since the right side of (3) is computable, the problem becomes the solution of a set of simultaneous differential equations by numerical means. Hence, any of the standard methods for this problem are also applicable to the optimising of a function of several variables if only the gradient is to be used. Sample references are Henrici (1962) and Ralston (1965).

The gradient method in this scheme is equivalent to the Euler or point-slope method which is not generally proposed as a practical method for solving ODE's but is mainly useful for illustration and proofs of convergence. The application of ODE solving methods such as predictor-corrector or deferred limit algorithms to this problem brings a number of important benefits. In the first place, they are usually demonstrably stable in the sense of not accumulating error in the estimation of points on the path. Moreover, extrapolations are made on the basis of more than one previous point which in general means a larger step size. Finally, estimates of extrapolation error are available and can be used to control step size. This transforms the problem of choosing step size into the more meaningful problem of choosing tolerance limits on the error of extrapolation.

Precisely which ODE method is best in this context probably depends on the problem. The equations (3) are by their nature stiff; that is, they are highly stable and this results in severe instability of the numerical method if step size becomes too large. This emphasises the advantage of multi-step or deferred limit methods over the classical steepest ascent algorithm since the parameters of the more complex procedures can be chosen to maximise stability. The highly stable methods of Gear (1967), Nordsieck (1962) and Widlund (1967), may be especially useful in this application.

The equations (3) are by no means unique in characterising the gradient path. Any transformation of the type

$$t = \int_0^r \frac{||s(y)||}{h(||s(y)||)} dx \tag{4}$$

where h is a positive function of $||s(y)||$ and the integration is over the gradient path, will produce the equivalent system:

$$y'(t) = \frac{s(y)}{h(||s(y)||)} \tag{5}$$

The choice of h in effect determines the step size as a function of gradient length. This is best viewed in the context of the path following the top of a ridge. If $h = 1$, then the step size when the current location is on the steep side of the ridge will be much more than that when the location is near the less steeply sloped top of the ridge. This will tend to lead to severe oscillation about the line of the true gradient path and is likely to impede progress. This characterisation of the gradient path was proposed by Arrow, Hurwicz and Uzawa, 1958. On the other hand, if h is a too steeply increasing function of $s(y)$, then the increased step size when on the top of the ridge will be likely to lead to trouble in terms of the next location. The author has generally found that use of the equations (3) implying $h = ||s(y)||$ and a uniform step size to be the most generally useful.

Special cases in which the differential equations (3) are explicitly solvable can be easily constructed. One of these situations is provided by the quadratic having a diagonal matrix of second derivatives. Although such a problem can be solved by analytic means, it may be instructive to consider the functions, $y(r)$, as a possible nonlinear path in parameter space which can be followed for any function and which, in this instance, leads to the optimum.

If one of the equations (3) is chosen, say equation j, and for each of the remainder the ratio,

$$\frac{y_i'(r)}{y_j'(r)} = \frac{dy_i}{dy_j} = \frac{s_i(y)}{s_j(y)}, \quad i = 1, \ldots, j-1, j+1, \ldots, n, \tag{6}$$

is taken, then the points on the path are defined as functions of parameter j. This reduces the number of differential equations to be solved by one but does not alter the number of quantities to be computed. For the following quadratic function,

$$f(x) = a'x + 1/2x'Dx \tag{7}$$

where D is diagonal, (6) becomes

$$\frac{dy_i}{dy_j} = \frac{a_i + d_{ii}y_i}{a_j + d_{jj}y_j} \tag{8}$$

The solution to (8) is given by:

$$y_i = \frac{c_i}{d_{ii}}(a_j + d_{jj}y_j)^{\frac{dii}{djj}} - \frac{a_i}{djj} \tag{9}$$

If two successive points and their gradients are indicated by $y^{(1)}$, $s^{(1)}$, $y^{(2)}$, and $s^{(2)}$, then solutions for a, D, and the integration constants, c_i, are given by:

$$\left.\begin{aligned} d_{ii} &= \frac{s_i^{(2)} - s_i^{(1)}}{y_i^{(2)} - y_i^{(1)}} \\ a_i &= s_i - d_{ii}\, y_i^{(1)} \\ c_i &= s_i^{(1)} \frac{d_i}{[s_i^{(1)}]^{dj}} \end{aligned}\right\} \tag{10}$$

It may be that optimising the function along the nonlinear path specified by (9) and (10) will be more efficient for

some functions than following the linear path specified in the gradient method.

3. Acceleration techniques

As the discussion of differential equation solving methods in this context indicated, the problem of choosing step size can be transformed to the intuitively meaningful problem of setting bounds on the error in estimating points along the gradient path. Too liberal bounds will usually set the estimates to oscillating about the line along the top of the ridge with relatively little forward progress. Too stringent bounds may produce more accuracy than is worthwhile. It is probably useful to choose the error bounds in order to maintain a certain minimal smoothness of the estimated path without allowing this smoothness to become so great that forward progress is sacrificed. One might, for example, update the error bounds on the basis of the average cosine of the angle between successive steps taken over twenty or so interations so as to keep this average within the limits of 0·90 to 0·95. This tends to ensure a relative smooth rate of forward progress with the necessary slowing when the ridge takes a sharp turn or falls off very steeply on either side.

The definition of each parameter as a function of the single variable, arc length along the gradient path from the initial point, suggests that an examination of the behaviour of this relationship and some form of extrapolation might be beneficial. Two procedures will be outlined here:

The first approach is to devise an extrapolating function, $Y_i(r)$, for the ith parameter function, $y_i(r)$. For some interval, $[r_0, r_1]$ over which approximations to y_i have been computed, it is possible to fit an interpolating polynomial of degree n, $p_i^{(n)}$, by some means. It is well known that use of this polynomial for extrapolating from r_1 to some value, r_2, is unlikely to be successful since the error is an accelerating function of $r_2 - r_1$. A more useful function is

$$Y_i = w_i p_i^{(n)} + (1 - w_i)p^{(1)}, \tag{11}$$

where

$$w_i = \min\left[1, \frac{p_i^{(n)\prime}(r_1) - p_i^{(n)\prime}(r_0)}{r_1 - r_0} \Big/ \frac{p_i^{(n)\prime}(r_2) - p_i^{(n)\prime}(r_1)}{r_2 - r_1}\right] \tag{12}$$

That is, the extrapolating function is a weighted sum of the linear and nth degree interpolating polynomials such that the average change in slope of the function over the interval of interpolation, $[r_0, r_1]$, is greater than or equal to the average slope change over the extrapolation interval, $[r_1, r_2]$. The success of this extrapolation therefore depends on the function $y_i(r)$, not undergoing a radical change in slope at some point. Such a radical change can be expected to occur in the initial stages of following the gradient path as the path ascends to the side of a steep ridge and then turns to follow the top of the ridge. It is advisable, therefore, to provide a test of whether such a sharp change has occurred in a particular interval and to do the interpolation after such a change.

A second extrapolation possibility is to use the function, $s_i(r)$, which is the ith gradient element considered as a function of arc length. If the interpolating function over the interval, $[r_0, r_1]$, is

$$S_i(r) = (r - a)(b_i r - c_i) \tag{13}$$

where the factor, $r - a$, common to all s_i is necessary to make them all vanish for the same value of r, then

$$y_i'(r) = \frac{b_i r - c_i}{r^2 \sum_i b_i^2 - 2r \sum_i b_i c_i + \sum c_i^2} \tag{14}$$

The obvious interval of extrapolation in this case is $[r_1, a]$ since the predicted optimum is at $r = a$. The integral of (14) over this interval can be evaluated directly to provide an estimate of the optimum point.

One possibility for the estimation of a, b_i, and c_i in (14) is to fit a quadratic in r to the computed values of $s_i(r)$ in the interpolation interval. For each such quadratic, the root which is real, greater than r_1, and accompanied by a negative or positive slope depending on whether a maximum or minimum is sought is an estimate of a. The mean of these estimates can then be used as the final estimate and the values of b_i and c_i estimated by a linear approximation to the computed values of $s_i/(r - a)$.

4. An example

A mathematical model of learning proposed by Audley and Jonckheere (1956) provides an illustration of the kind of problem for which the gradient path methods are especially suited. This model defines a stochastic process which specifies the probabilities of making a correct response at each trial in a sequence in which the subject is supposed to be learning a correct choice. If a random variable, X_i, is defined which takes on the value of one on trial i if the correct response was made and zero otherwise, then the five parameter model for the probability of this variable being one is:

$$P\{X_i = 1\} = \frac{\rho + (\alpha - \beta)K_i + (i - 1)\beta}{1 + (\gamma_1 - \gamma_2)K_i + (i - 1)\gamma_2}, \tag{15}$$

where $K_i = \sum_{j=1}^{i-1} X_j$ is the number of correct responses before trial i. Interpretations of the five parameters, ρ, α, β, γ_1, and γ_2, can be found in the reference cited above.

The problem is to find maximum likelihood estimates of these parameters by maximising the following log likelihood function:

$$\log L = \sum_{i=1}^{N} [X_i \log p_i + (1 - X_i) \log (1 - p_i)] \tag{16}$$

where p_i is the probability of a correct response on the ith trial specified by (15) and there are a total of N trials. Since these probabilities must be in the interval (0, 1), it is necessary to place the following restrictions on the parameters:

$$\begin{aligned} 0 < p < 1 \\ \alpha \geqslant 0 \\ \beta \geqslant 0 \\ \gamma_1 - \alpha \geqslant 0 \\ \gamma_2 - \beta \geqslant 0 \end{aligned}$$

In order to avoid phrasing this problem as one in nonlinear programming, the parameters were transformed in

the following way:

$$t^2 = \rho$$
$$a^2 = \alpha$$
$$b^2 = \beta$$
$$c^2 = \gamma_1 - \alpha$$
$$d^2 = \gamma_2 - \beta$$

The transformed parameters are now unconstrained with the exception of t which, by the nature of the data, is very unlikely to exceed one. However, this is achieved only at the expense of making the boundaries absorbing in the sense that, if any of these parameters is given the value of zero, its derivative vanishes and it cannot lose that value. Hence, it is important that the optimisation method compute estimated points which have as little danger as possible of prematurely striking a boundary. Approaches which maximise the function along a line are clearly not desirable for this reason.

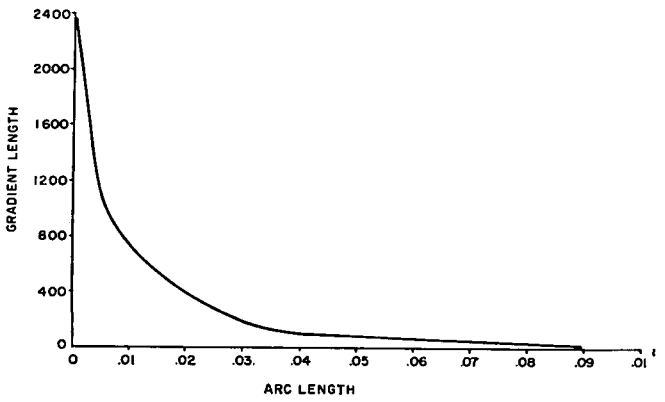

Fig. 1. Gradient length as a function of arc length along the gradient path

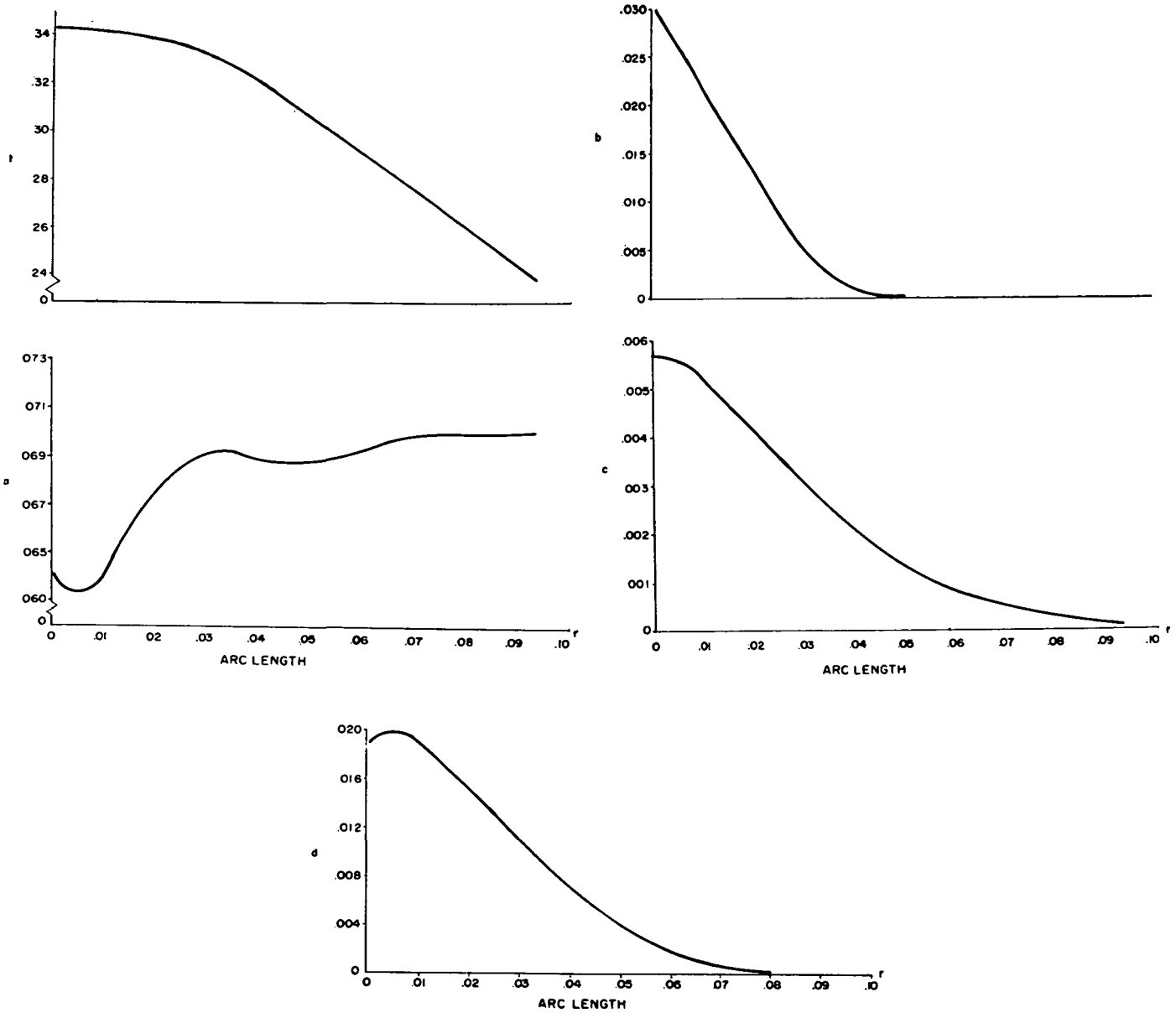

Fig. 2. The parameters as functions of arc length

The differential equation solving method used in this case was Hamming's method. Error bounds were adjusted every twenty iterations to keep the average cosine of the angle between successive gradients between 0·90 and 0·95. None of the extrapolation methods mentioned previously were employed. It was found that once sufficiently close to the optimum, a small number of successive approximation iterations would converge and thereby save considerable time over going all the way to the maximum by the gradient path method. In fact, it has generally been the author's experience that the predictor-corrector algorithm makes very slow progress in the immediate neighbourhood of the optimum and that it is usually worthwhile to switch to some other algorithm.

The data analysed in this were collected by H. Gulliksen from a single cat which responded on 1,294 trials in a simple two-choice task. The initial point for the transformed parameters (t, a, b, c, d), was (0·34403, 0·06431, 0·03041, 0·00581, 0·01905) and was generated by a simpler estimation procedure described elsewhere (Ramsay, 1969). The initial gradient length was 2355 and the initial function value was −758·079. After 275 iterations, the gradient length was 2·642 and the successive approximation iterations began. After five of these, the gradient length was 0·0002174 and this value was the lower limit that could be attained using double precision on the IBM 360, Model 75. The final point for the transformed parameters was (0·23891, 0·07211, 0·00000, 0·00000, 0·00000) and the final function value was −732·419. Thus, the optimum was on the boundary of the admissible region. The matrix of second derivative values at this point is given in **Table 1**. **Fig. 1** indicates the gradient length as a function of arc length along the path and **Fig. 2** shows the parameters as functions of arc length. These figures display the features of the gradient path common to most optimization problems. There is an initial sharp rise in the function value as the path ascends the nearest ridge. Once the top of the ridge is attained, at about arc length 0·01, the path changes direction as indicated by the curvature of the plots for parameters a, c, and d in that region. A second shallower ridge is joined at arc length 0·04 when the parameter b is driven to its boundary value. From that point progress is relatively slow and the parameters as functions of arc length are roughly linear. At arc length 0·093 successive approximations commenced and produced the final estimate in five iterations.

Table 1
Second derivatives of the log likelihood function at the optimum

	r	a	b	c	d
r	−4281	−15900	0	0	0
a	−15900	−155900	0	0	0
b	0	0	−7678	0	0
c	0	0	0	−272·6	0
d	0	0	0	0	−3147

References

ARROW, K. J., HURWICZ, L., and UZAWA, H. (1958). *Studies in linear and nonlinear programming*, Stanford: Stanford University Press.

AUDLEY, R. J., and JONCKHEERE, A. R. (1956). The statistical analysis of the learning process, *British Journal of Statistical Psychology*, Vol. 9, p. 87.

BARNES, J. G. P. (1965). An algorithm for solving non-linear equations based on the secant method, *The Computer Journal*, Vol. 8, p. 66.

BOX, M. J. (1966). A comparison of several current optimization methods, and the use of transformations in constrained problems, *The Computer Journal*, Vol. 9, p. 67.

FLETCHER, R., and POWELL, M. J. D. (1963). A rapidly convergent descent method for minimization, *The Computer Journal*, Vol. 6, p. 163.

FLETCHER, R., and REEVES, C. M. (1964). Function minimization by conjugate gradients, *The Computer Journal*, Vol. 7, p. 149.

GEAR, C. W. (1967). The numerical integration of ordinary differential equations, *Math. of Computation*, Vol. 21, p. 146.

HADLEY, G. (1964). *Nonlinear and dynamic programming*, Reading: Addison-Wesley.

NELDER, J. A., and MEAD, R. (1965). A simpler method for function minimization, *The Computer Journal*, Vol. 7, p. 308.

NORDSIECK, A. (1962). On numerical integration of ordinary differential equations, *Math. of Computation*, Vol. 16, p. 22.

POWELL, M. J. D. (1965). A method for minimizing a sum of squares of non-linear functions without calculating derivatives, *The Computer Journal*, Vol. 7, p. 303.

RALSTON, A. (1965). *A first course in numerical analysis*, New York: McGraw-Hill.

RAMSAY, J. O. (1969). Parameter estimation in the Audley-Jonckheere learning model, Unpublished manuscript.

ROSENBROCK, H. H. (1960). An automatic method for finding the greatest or least value of a function, *The Computer Journal*, Vol. 3, p. 175.

SWANN, W. H. (1964). Report on the development of a new direct searching method of optimization, I.C.I. Ltd., Central Instrument Laboratory Research Note 64/3.

WIDLUND, O. B. (1962). A note on unconditionally stable linear multi-step methods, *BIT*, Vol. 7, p. 65.

WILDE, D. J., and BEIGHTLER, C. S. (1967). *Foundations of optimization*, Englewood Cliffs: Prentice-Hall.