

EXTREMIZATION OF FUNCTIONS WITH EQUALITY CONSTRAINTS *

H. KWAKERNAAK and R.C.W. STRIJBOS

*Delft University of Technology, Delft, The Netherlands ***

Received 10 September 1970

Revised manuscript received 26 October 1971

By the use of Lagrange multipliers the problem of minimizing or maximizing a function of n variables with m equality constraints is converted into the problem of solving $n + m$ nonlinear equations. A quasi-Newton method is proposed to solve these equations. It is assumed that the gradient of the function and the Jacobian of the constraints are explicitly available. Compact and efficient updating formulas are given. The numerical results are compared to those for a method recently described by Powell [1] for the same problem.

1. Introduction

In recent years several techniques have been developed for the unconstrained minimization of a function $\alpha(\mathbf{x})$ with respect to the vector-valued variable $\mathbf{x} = \text{col}(x_1, x_2, \dots, x_n)$. If the gradient $\text{grad}\{\alpha(\mathbf{x})\}$ is explicitly available for each value of \mathbf{x} , the most effective techniques known are the conjugate gradient method of Fletcher and Reeves (for a description see Pierre [2]) and Davidon's method, as extended by Fletcher and Powell (see also Pierre [2]). If equality constraints of the form

$$\mathbf{g}(\mathbf{x}) = \mathbf{0}$$

are present, where $\mathbf{g}(\mathbf{x})$ is an m -dimensional vector function ($m < n$), a well-known approach is to convert the constrained minimization prob-

* This paper was presented at the 7th Mathematical Programming Symposium 1970, The Hague, The Netherlands.

** Present address: Twente University of Technology, Enschede, The Netherlands.

lem into a sequence of unconstrained problems by the use of penalty functions. The SUMT method of Fiacco and McCormick [3] is a good example of this technique. A modification of this method has recently been proposed by Powell [1]. He suggests to minimize the modified objective function

$$\phi_k(\mathbf{x}) = \alpha(\mathbf{x}) + [\mathbf{g}(\mathbf{x}) + \boldsymbol{\theta}_k]^T \boldsymbol{\Sigma}_k [\mathbf{g}(\mathbf{x}) + \boldsymbol{\theta}_k],$$

where

$$\boldsymbol{\theta}_k = \text{col}(\theta_{k1}, \theta_{k2}, \dots, \theta_{km}),$$

and where $\boldsymbol{\Sigma}_k$ is the diagonal matrix

$$\boldsymbol{\Sigma}_k = \text{diag}(\sigma_{k1}, \sigma_{k2}, \dots, \sigma_{km}).$$

The function $\phi_k(\mathbf{x})$ is minimized for a sequence of values of $\boldsymbol{\theta}_k$ and $\boldsymbol{\Sigma}_k$, where each time the minimizing value of \mathbf{x} is the starting point for the next minimization. As it turns out, moderate values of the parameters θ and $\boldsymbol{\Sigma}$ suffice to obtain the solution. Powell gives simple rules to determine the successive parameter values.

In the present paper an entirely different approach is taken. By the introduction of Lagrange multipliers the problem of minimizing $\alpha(\mathbf{x})$ subject to the constraint $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ is converted to the problem of solving $n + m$ simultaneous nonlinear equations. For the solution of these equations a method is developed of the quasi-Newton type. The method has the property that if $\alpha(\mathbf{x})$ is quadratic and $\mathbf{g}(\mathbf{x})$ linear, a solution is reached in $n + 1$ steps or less. For the application of the method it is assumed that both the gradient of $\alpha(\mathbf{x})$ and the Jacobian of $\mathbf{g}(\mathbf{x})$ are available for each value of \mathbf{x} .

The remainder of the paper is organized as follows: In section 2 a quasi-Newton method for the solution of a set of nonlinear equations is reviewed. Compact and efficient formulas for updating the inverse of the Jacobian are given. Next in section 3 the problem of minimizing $\alpha(\mathbf{x})$ subject to $\mathbf{g}(\mathbf{x}) = \mathbf{0}$ is converted into the problem of solving a set of simultaneous nonlinear equations, and it is shown how the method of section 2 may be applied to this problem. A systematic enumeration of the successive steps in the iterative method concludes this section. In

section 4 a comparison is made between the present method and that of Powell described above by applying the two methods to a number of examples. Section 5 presents the conclusions of the paper.

2. A quasi-Newton method for solving simultaneous equations

In this section the quasi-Newton method of solving nonlinear simultaneous equations is reviewed. The presentation is similar to that of Rosen [4]. Let the set of nonlinear equations be given in the form

$$f(\mathbf{x}) = \mathbf{0} , \tag{1}$$

where f is an n -dimensional vector function. Let \mathbf{x}_i be the approximate solution that has been found at the end of i iterations. Then the $(i + 1)$ -th iteration proceeds as follows:

(1) Establish the search direction \mathbf{p}_i from

$$\mathbf{p}_i = - H_i \mathbf{f}_i . \tag{2}$$

Here H_i is an approximation to $J^{-1}(\mathbf{x}_i)$, where $J(\mathbf{x})$ is the Jacobian of $f(\mathbf{x})$. Furthermore, \mathbf{f}_i stands for $f(\mathbf{x}_i)$.

(2) Calculate the $(i + 1)$ -th approximation \mathbf{x}_{i+1} from

$$\mathbf{x}_{i+1} = \mathbf{x}_i + t_i \mathbf{p}_i . \tag{3}$$

Here the step length factor t_i is so chosen that $\mathbf{f}^T(\mathbf{x})f(\mathbf{x})$ is reduced when going from \mathbf{x}_i to \mathbf{x}_{i+1} . This reduction is achieved by computing $\mathbf{f}^T(\mathbf{x}_{i+1})f(\mathbf{x}_{i+1})$ for a few values of t_i . The first value of t_i is taken equal to 1 provided a prescribed maximal change in each of the components of \mathbf{x} is not exceeded; otherwise t_i is adjusted so as to satisfy this condition. If no reduction of $\mathbf{f}^T(\mathbf{x})f(\mathbf{x})$ is obtained successively the values $0.3 t_i$, $0.09 t_i$ and $-0.3 t_i$ are tried.

(3) Update the approximated inverse Jacobian with the formula

$$H_{i+1} = H_i + \frac{(\delta \mathbf{x}_i - H_i \delta \mathbf{f}_i) \mathbf{z}_i^T H_i}{\mathbf{z}_i^T H_i \delta \mathbf{f}_i} , \tag{4}$$

where

$$\delta f_i = f_{i+1} - f_i, \quad (5)$$

$$\delta x_i = t_i p_i = x_{i+1} - x_i. \quad (6)$$

Rosen [4] reviews various possible choices for the n -dimensional column vector z_i . A quite reasonable choice has been suggested by Barnes [5]. He selects z_i orthogonal to the previous $n - 1$ search directions. It may be shown that if the simultaneous equations (1) are *linear* the approximated inverse Jacobian H_i is exact after n iterations with this choice of z_i , and hence convergence to the exact solution is obtained at the end of the $(i + 1)$ -th iteration.

Barnes recommends the Gram-Schmidt method to make z_i orthogonal to the previous $n - 1$ search directions. In the following a different method is suggested that leads to compact and efficient formulas. These formulas are very similar to those given by Murtagh and Sargent [6]. Here the approximated inverse Jacobian is obtained as follows: First a matrix F_i is updated:

$$F_{i+1} = \hat{F}_i + \frac{(e - \hat{F}_i \delta f_i) e^T \hat{F}_i}{e^T \hat{F}_i \delta f_i}. \quad (7)$$

The vector $e = \text{col}(0, 0, \dots, 0, 1)$ is n -dimensional, and \hat{F}_i is obtained from F_i by a cyclic permutation of rows, i.e., if

$$F_i = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{pmatrix}, \quad (8)$$

where the $r_j, j = 1, 2, \dots, n$, are row vectors, then

$$\hat{F}_i = \begin{pmatrix} r_2 \\ r_3 \\ \vdots \\ r_n \\ r_1 \end{pmatrix}. \quad (9)$$

With the aid of F_{i+1} the updated inverse Jacobian may be obtained as

$$H_{i+1} = H_i + (\delta x_i - H_i \delta f_i) e^T F_{i+1} . \quad (10)$$

In the Appendix it is shown how the formulas (7) and (10) may be derived. There it is also proved that updating the inverse Jacobian according to (7) and (10) is equivalent to using (4), and making z_i orthogonal to the previous $n - 1$ directions searched.

To complete the description of the updating process, starting values H_0 and F_0 must be prescribed. These may be selected independently. A computationally convenient but rather arbitrary choice is

$$H_0 = I, \quad F_0 = I, \quad (11)$$

where I is the unit matrix. Better results may be achieved by choosing for H_0 some approximation to the inverse Jacobian, possibly obtained by finite differencing. In this case the choice of F_0 is still arbitrary. The arbitrariness of F_0 reflects the fact that during the first $n - 1$ iterations the prescription of choosing z_i orthogonal to the previous $n - 1$ search directions does not completely determine z_i , since the $n - 1$ previous search directions are not yet available.

3. Extremization with equality constraints

In this section the problem is considered of minimizing the function $\alpha(x)$ subject to the constraint $g(x) = 0$, where x is an n -dimensional column vector and g an m -dimensional vector function, with $m < n$. To find necessary conditions for a stationary point, form the function

$$\alpha(x) + \lambda^T g(x), \quad (12)$$

where $\lambda = \text{col}(\lambda_1, \lambda_2, \dots, \lambda_m)$ is a vector of Lagrange multipliers. Differentiation of this expression with respect to x yields the following necessary condition for a stationary point x^0 :

$$\text{grad}[\alpha(x^0)] + J_g^T(x^0) \lambda^0 = 0. \quad (13)$$

Here $J_g(\mathbf{x})$ denotes the Jacobian of the vector function \mathbf{g} . The n equations (13), together with the m equations

$$\mathbf{g}(\mathbf{x}^0) = \mathbf{0} \quad (14)$$

form a set of $n + m$ equations for \mathbf{x}^0 and the multiplier vector λ^0 . Together \mathbf{x}^0 and λ^0 represent $n + m$ unknowns. Thus, the constrained minimization problem is replaced with the problem of solving the simultaneous nonlinear equations (13) and (14) for \mathbf{x}^0 and λ^0 .

Define the vector function

$$\mathbf{b}(\mathbf{x}, \lambda) = \text{grad}[\alpha(\mathbf{x})] + J_g^T(\mathbf{x})\lambda, \quad (15)$$

and consider the problem of solving the equations

$$\mathbf{b}(\mathbf{x}, \lambda) = \mathbf{0}, \quad \mathbf{g}(\mathbf{x}) = \mathbf{0} \quad (16)$$

for \mathbf{x} and λ . A quasi-Newton method will be developed for this problem. It will be assumed that the function values \mathbf{b} and \mathbf{g} may be explicitly evaluated for any given values of \mathbf{x} and λ . For the evaluation of \mathbf{b} this means that *the gradient* $\text{grad}[\alpha(\mathbf{x})]$ *and the Jacobian* $J_g(\mathbf{x})$ *of* $\mathbf{g}(\mathbf{x})$ *are explicitly available for each value of* \mathbf{x} *and* λ .

The Jacobian J of the combined equations (16) with respect to \mathbf{x} and λ may be partitioned as

$$J(\mathbf{x}, \lambda) = \begin{pmatrix} \mathbf{G}(\mathbf{x}, \lambda) & J_g^T(\mathbf{x}) \\ J_g(\mathbf{x}) & \mathbf{0} \end{pmatrix}. \quad (17)$$

Here $\mathbf{G}(\mathbf{x}, \lambda)$ is the Jacobian of $\mathbf{b}(\mathbf{x}, \lambda)$ with respect to \mathbf{x} , i.e., $\mathbf{G}(\mathbf{x}, \lambda)$ is an $n \times n$ matrix, the (i, j) -th element of which is given by

$$G_{ij}(\mathbf{x}, \lambda) = \frac{\partial h_i(\mathbf{x}, \lambda)}{\partial x_j}. \quad (18)$$

When the Jacobian $J(\mathbf{x}, \lambda)$ is precisely known, Newton's method prescribes the following step to reach an approximate solution of (16) from a tentative solution \mathbf{x} :

$$p = -J^{-1}(x, \lambda) \begin{pmatrix} b(x, \lambda) \\ g(x) \end{pmatrix}. \tag{19}$$

Using the formula for the inverse of a partitioned matrix (see e.g. Noble [7], p. 25) this may be rewritten as

$$p = - \begin{pmatrix} G^{-1} - G^{-1} J_g^T (J_g G^{-1} J_g^T)^{-1} J_g G^{-1} & G^{-1} J_g^T (J_g G^{-1} J_g^T)^{-1} \\ (J_g G^{-1} J_g^T)^{-1} J_g G^{-1} & -(J_g G^{-1} J_g^T)^{-1} \end{pmatrix} \begin{pmatrix} b \\ g \end{pmatrix} \tag{20}$$

where the arguments x and λ have been omitted. The advantage of this expression is that the inversion of the $(n + m) \times (n + m)$ matrix J is reduced to the inversion of the $m \times m$ matrix $J_g G^{-1} J_g^T$. Since the matrix J_g and the functions b and g are supposed to be explicitly available at each stage of the iteration, the only quantity that requires additional computation is the matrix G , which only occurs in (20) in the form of its inverse G^{-1} . A method will be given to estimate G^{-1} by an updating technique similar to that described in section 2 of this paper.

From (15) it follows that

$$b(x + \delta x, \lambda + \delta \lambda) = b(x, \lambda) + G(x, \lambda) \delta x + J_g^T(x, \lambda) \delta \lambda,$$

where second-order terms have been neglected, Apparently the matrix G^{-1} obeys the (approximate) equality

$$G^{-1}(\delta b - J_g^T \delta \lambda) = \delta x, \tag{21}$$

where

$$\delta b = b(x + \delta x, \lambda + \delta \lambda) - b(x, \lambda). \tag{22}$$

Equation (21) forms the counterpart of the equation

$$J^{-1} \delta f = \delta x$$

that forms the starting point for the updating technique of section 2.

In analogy with (7) and (10), equation (21) results in the updating formulas

$$K_{i+1} = \hat{K}_i + \frac{[e - \hat{K}_i(\delta b_i - J_{gi}^T \delta \lambda_i)] e^T \hat{K}_i}{e^T \hat{K}_i (\delta b_i - J_{gi}^T \delta \lambda_i)}, \quad (23)$$

$$L_{i+1} = L_i + [\delta x_i - L_i(\delta b_i - J_{gi}^T \delta \lambda_i)] e^T K_{i+1}. \quad (24)$$

Here K_i is an auxiliary matrix, \hat{K}_i is obtained from K_i by cyclic permutation of the rows, and L_i is the approximation to $G^{-1}(x_i, \lambda_i)$. J_{gi} and δb_i denote J_g and δb , evaluated at x_i and λ_i , the approximate solution that has been obtained at the end of the i -th iteration. When L_i is substituted for G^{-1} into (20), the p that is thus computed is used as a search direction rather than as a full step.

Using the results of this section, the complete algorithm for determining a stationary point of $\alpha(x)$ subject to $g(x) = 0$ may be summarized as follows:

1. Set the iteration index i equal to 0. Choose x_0 , λ_0 , K_0 and L_0 . Compute J_{g0} , g_0 and b_0 . Here J_{gi} , g_i and b_i denote $J_g(x_i, \lambda_i)$, $g(x_i)$ and $b(x_i, \lambda_i)$.
2. Compute the search direction

$$p_i = - \begin{pmatrix} L_i - L_i J_{gi}^T (J_{gi} L_i J_{gi}^T)^{-1} J_{gi} L_i & L_i J_{gi}^T (J_{gi} L_i J_{gi}^T)^{-1} \\ (J_{gi} L_i J_{gi}^T)^{-1} J_{gi} L_i & -(J_{gi} L_i J_{gi}^T)^{-1} \end{pmatrix} \begin{pmatrix} b_i \\ g_i \end{pmatrix}.$$

3. Calculate x_{i+1} and λ_{i+1} from

$$\begin{pmatrix} x_{i+1} \\ \lambda_{i+1} \end{pmatrix} = \begin{pmatrix} x_i \\ \lambda_i \end{pmatrix} + t_i p_i,$$

where the step length factor t_i is so chosen that $g^T g + k \cdot b^T b$ is reduced. It has been found favourable to keep the constant k equal to 0 as long as a good reduction of $g^T g$ is achieved. Next k is set equal to 1. Retain $J_{g,i+1}$, g_{i+1} and b_{i+1} . If no improvement is found, retain these quan-

tities for the last step length that has been attempted. If the extremum has been found stop, otherwise proceed.

4. Update K_i and L_i , using (23) and (24). If no improvement was obtained in step 3, use the quantities corresponding to the last step length attempted for updating.

5. Increase i by 1 and return to step 2.

The following comments are appropriate:

(a) If the process converges to a solution, this solution is a stationary point, and not necessarily a constrained extremum. In specific problems it may be possible to build into step 3 a mechanism that steers the process towards a constrained minimum or maximum.

(b) If the function $\alpha(\mathbf{x})$ is quadratic in \mathbf{x} , and the constraint functions $\mathbf{g}(\mathbf{x})$ are linear in \mathbf{x} , the matrix \mathbf{G} is constant. This means that at the end of n iterations L_n is precisely \mathbf{G}^{-1} , and hence, that at the end of the $(n + 1)$ -th iteration the exact solution is reached.

(c) The Jacobian \mathbf{J} (eq. (17)) may be invertible while \mathbf{G} is not. In that case eq. (20) is not correct. This slightly reduces the range of applicability of the method.

4. Numerical results and comparison with Powell's method

In this section numerical experiences with the constrained extremization technique proposed in section 3 are reported, and a comparison is made with the method of Powell that is described in the introduction. To achieve a fair comparison, the unconstrained minimization problem that must be solved at each stage of Powell's method is attacked with a quasi-Newton technique as described in section 2. To give Powell's method every advantage, the computer program that has been written to implement this method includes the feature of carrying over the estimate of \mathbf{H} to the next unconstrained minimization, as suggested by Powell.

The following problems have been considered:

a. *Rosenbrock's problem*

The function to be minimized is given by

$$\alpha(\mathbf{x}) = (x_2 - x_1^2)^2 + 0.01(1 - x_1)^2 .$$

Table 1
Results for Rosenbrock's problem

	Powell's method	Present method
maximal change in x_1 and x_2	total number of calculations	
0.2	61	37
1	16	12
3	10	18

The constraint is a parabola

$$g(\mathbf{x}) = x_1(x_1 - 4) - 2x_2 + 12 = 0.$$

In table 1 the results are given for the method suggested by Powell and the method proposed in this paper. A comparison is given of the total number of calculations necessary to reach the minimum. "One calculation" includes the evaluation of the value of α , of the gradient of α , of the constraint function g and of the Jacobian of g . Both methods are supplied with this information. The parameter in table 1 is the maximal allowed change in either component of \mathbf{x} . The initial value for \mathbf{x} is $x_1 = -1.2$, $x_2 = 1.0$. For the final value $x_1 = 1.99938$, $x_2 = 4.00000$ was found.

b. Powell's problem

The second example is taken from the paper of Powell in which he describes his method. The function to be minimized is

$$\alpha(\mathbf{x}) = \exp(x_1 x_2 x_3 x_4 x_5).$$

There are three equality constraints:

$$g_1(\mathbf{x}) = x_1^2 + x_2^2 + x_3^2 + x_4^2 + x_5^2 - 10 = 0,$$

$$g_2(\mathbf{x}) = x_2 x_3 - 5x_4 x_5 = 0,$$

$$g_3(\mathbf{x}) = x_1^3 + x_2^3 + 1 = 0.$$

Table 2
Results for Powell's problem

	Powell's method	Present method
maximal change in x_1 through x_5	total number of calculations	
0.1	301	41
0.5	192	31
3	232	26

Starting with the values

$$x_1 = x_4 = -2, \quad x_2 = x_3 = 2, \quad x_5 = -1,$$

the following solution has been found:

$$x_1 = -1.71714, \quad x_2 = 1.59571, \quad x_3 = 1.82725, \\ x_4 = x_5 = -0.76364.$$

Table 2 presents the number of calculations for both methods.

c. Pendulum problem

The constrained extremization technique proposed may be applied to solve certain control problems. The last example will illustrate this. The motion of the mathematical pendulum depicted in fig. 1 is described by the differential equation

$$\ddot{\phi}(t) = -k_1 \sin \phi(t) + k_2 u(t) \cos \phi(t),$$

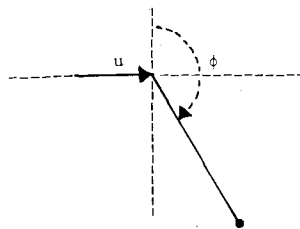


Fig. 1. Mathematical pendulum.

where $\phi(t)$ is the angle with the vertical, k_1 and k_2 are constants depending on the length of the pendulum and the gravitational acceleration, and $u(t)$ is the horizontal acceleration of the pivot.

The problem is to bring the pendulum from its stable to its unstable equilibrium position in a given length of time T while minimizing the total acceleration required. Furthermore it is required that the pivot returns to its initial position with velocity equal to zero at time $t = T$.

When time is discretized and the horizontal acceleration is supposed to be piecewise constant, i.e.,

$$u(t) = x_i = \text{constant for } (i-1)T/N \leq t < iT/N,$$

with N a suitable integer, the optimal control problem mathematically can be stated as follows:

Minimize

$$\alpha(\mathbf{x}) = \sum_{i=1}^N x_i^2$$

subject to the constraints

$$g_1(\mathbf{x}) = \phi(T) = 0,$$

$$g_2(\mathbf{x}) = \dot{\phi}(T) = 0,$$

$$g_3(\mathbf{x}) = \int_0^T u(t) dt = (T/N) \sum_{i=1}^N x_i = 0,$$

$$g_4(\mathbf{x}) = \int_0^T dt \int_0^T u(\tau) d\tau = (T/N)^2 \sum_{i=1}^N (N + 1/2 - i)x_i = 0.$$

The linear constraints $g_3 = 0$ and $g_4 = 0$ may be used to eliminate two variables. The values of $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$, are computed from analytical expressions for the solution of the differential equations involving elliptic functions.

In fig. 2 solutions for several values of N are sketched for the parameter values $k_1 = 11.65$, $k_2 = 1.188$, $T = 1$. Dotted horizontal lines de-

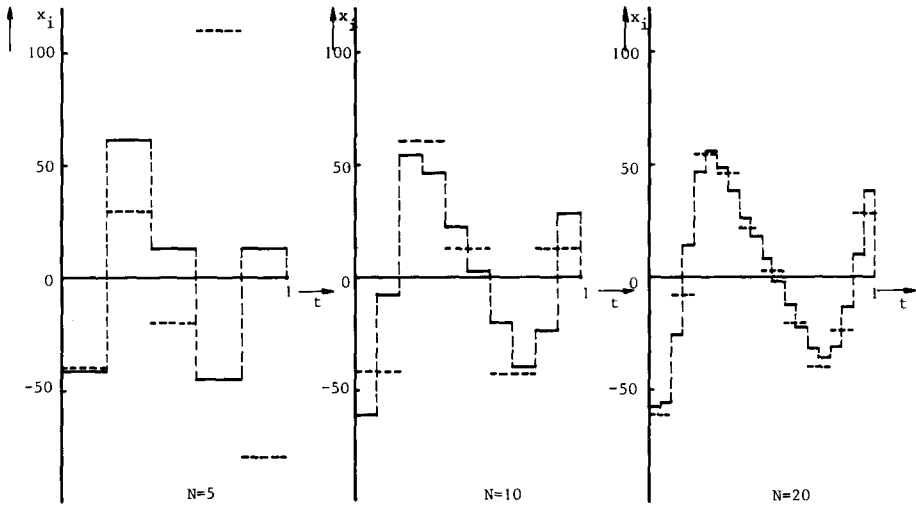


Fig. 2. Results for the pendulum problem.

note the starting values used for the control inputs x , the uninterrupted lines indicate the optimal values finally obtained. For $N = 5$ the optimal solution was obtained after 20 calculations, for $N = 10$ after 81 calculations and for $N = 20$ after 60 calculations. Fig. 3 gives a “stroboscopic” representation of the successive positions of the pendulum for $N = 20$.

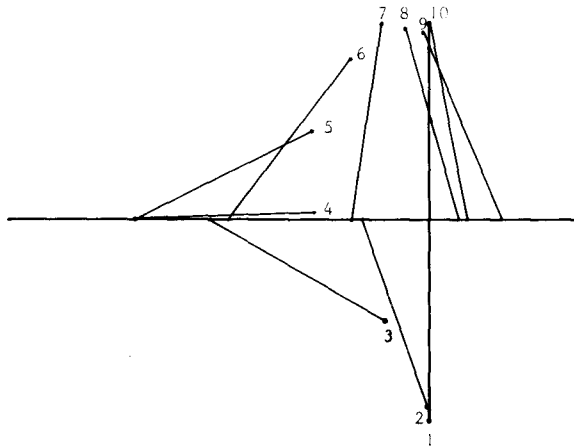


Fig. 3. Stroboscopic representation of the successive positions of the pendulum during the optimal motion. The positions are given at intervals of 0.1 second.

5. Conclusions

In this paper it is proposed to attack extremization problems with equality constraints by conversion to the problem of solving a set of nonlinear equations. An algorithm is described to solve these equations by a quasi-Newton method for which compact and efficient updating formulas are given. The algorithm has successfully been applied to several examples. It appears out that this method may sometimes achieve faster results than conversion of the constrained extremization problem into a sequence of unconstrained problems.

Appendix

In this appendix the origin of the updating formulas (7) and (10) is explained. The argument goes as follows: At the end of the $(i + 1)$ -th iteration the approximate solution of $f(\mathbf{x}) = \mathbf{0}$ is given by \mathbf{x}_{i+1} . Using a Taylor expansion about the point \mathbf{x}_{i+1} the following holds

$$f(\mathbf{x}_i) = f(\mathbf{x}_{i+1}) + J(\mathbf{x}_{i+1})(\mathbf{x}_i - \mathbf{x}_{i+1}), \quad (\text{A.1})$$

where terms of order two and higher have been neglected. $J(\mathbf{x})$ denotes the Jacobian of $f(\mathbf{x})$. In abbreviated form this may be written as

$$\delta f_i = J_{i+1} \delta \mathbf{x}_i. \quad (\text{A.2})$$

Defining H_{i+1} as the inverse of J_{i+1} this equation is identical to

$$\delta \mathbf{x}_i = H_{i+1} \delta f_i. \quad (\text{A.3})$$

Suppose now that H_{i+1} is calculated by assuming that the inverse Jacobian $H(\mathbf{x})$ has not changed during the last n iterations (n is the dimension of \mathbf{x}). Then the following relations must be satisfied

$$\begin{aligned} H_{i+1} \delta f_i &= \delta \mathbf{x}_i, \\ H_{i+1} \delta f_{i-1} &= \delta \mathbf{x}_{i-1}, \\ &\dots\dots \\ H_{i+1} \delta f_{i-n+1} &= \delta \mathbf{x}_{i-n+1}. \end{aligned} \quad (\text{A.4})$$

In matrix form these equations may be condensed as

$$H_{i+1}(\delta f_{i-n+1}, \delta f_{i-n+2}, \dots, \delta f_i) = (\delta x_{i-n+1}, \delta x_{i-n+2}, \dots, \delta x_i) . \tag{A.5}$$

Now, let H_{i+1} be represented as

$$H_{i+1} = H_i + \delta H_i . \tag{A.6}$$

If it is assumed that H_i satisfies the relations (A.4) with $i + 1$ replaced with i , it is easily found by substituting (A.6) into (A.5) that δH_i satisfies

$$\begin{aligned} \delta H_i(\delta f_{i-n+1}, \delta f_{i-n+2}, \dots, \delta f_i) &= (\mathbf{0}, \dots, \mathbf{0}, \delta x_i - H_i \delta f_i) \\ &= (\delta x_i - H_i \delta f_i) e^T , \end{aligned} \tag{A.7}$$

where $e = \text{col}(0, \dots, 0, 1)$. Now define

$$P_{i+1} = (\delta f_{i-n+1}, \delta f_{i-n+2}, \dots, \delta f_i) \tag{A.8}$$

and

$$F_{i+1} = P_{i+1}^{-1} . \tag{A.9}$$

Then it immediately follows from (A.7) that

$$\delta H_i = (\delta x_i - H_i \delta f_i) e^T F_{i+1} ,$$

and hence from (A.6)

$$H_{i+1} = H_i + (\delta x_i - H_i \delta f_i) e^T F_{i+1} . \tag{A.10}$$

To derive the iterative formula for F_{i+1} (Chidambara [8] has given a similar result) it is first noted that

$$P_{i+1} = \tilde{P}_i + (\delta f_i - \delta f_{i-n}) e^T , \tag{A.11}$$

where \tilde{P}_i is obtained from P_i by cyclic permutation of the columns, i.e., if

$$P = (p_1, p_2, \dots, p_n),$$

then

$$\tilde{P} = (p_2, \dots, p_n, p_1).$$

It follows from (A.9) that

$$F_{i+1} = P_{i+1}^{-1} = \tilde{P}_i^{-1} [I + (\delta f_i - \delta f_{i-n}) e^T \tilde{P}_i^{-1}]^{-1}. \quad (\text{A.12})$$

Since $\tilde{P}_i^{-1} = \hat{F}_i$, where \hat{F}_i is obtained from F_i by cyclic permutation of the rows, it follows from (A.12)

$$F_{i+1} = \hat{F}_i [I + (\delta f_i - \delta f_{i-n}) e^T \hat{F}_i]^{-1}. \quad (\text{A.13})$$

Using the inversion formula (Noble [7], p. 147)

$$(I + ab^T)^{-1} = I - ab^T(1 + b^T a), \quad (\text{A.14})$$

where a and b are arbitrary column vectors, (A.13) may be rewritten as

$$F_{i+1} = \hat{F}_i \left[I - \frac{(\delta f_i - \delta f_{i-n}) e^T \hat{F}_i}{1 + e^T F_i (\delta f_i - \delta f_{i-n})} \right]. \quad (\text{A.15})$$

From this result the updating formula

$$F_{i+1} = \hat{F}_i + \frac{(e - \hat{F}_i \delta f_i) e^T \hat{F}_i}{e^T \hat{F}_i \delta f_i} \quad (\text{A.16})$$

is easily obtained, using the fact that $\hat{F}_i \delta f_{i-n} = e$.

If the function $f(x)$ is linear, its Jacobian, and also hence its inverse Jacobian, is constant. In this case it is clear that if for H_0 the correct value is chosen, H_i will remain correct. If H_0 is not correctly chosen,

after n iterations the correct value is reached, and maintained after this. Barnes [5] discusses the convergence properties of the method if $f(\mathbf{x})$ is not linear. It remains to show that the present formulas correspond to choosing \mathbf{z}_i in (4) orthogonal to the previous $n - 1$ search directions. Comparing (4) and (10) it is seen that

$$\mathbf{z}_i^T \mathbf{H}_i = \mu \mathbf{e}^T \mathbf{F}_{i+1} , \quad (\text{A.17})$$

where $\mu = \mathbf{z}_i^T \mathbf{H}_i \delta \mathbf{f}_i$. Using the fact that \mathbf{H}_i satisfies (A.4) with $i + 1$ replaced with i , it easily follows by operating upon $\delta \mathbf{f}_{i-1}, \delta \mathbf{f}_{i-2}, \dots, \delta \mathbf{f}_{i-n+1}$ with both sides of (A.17) that

$$\begin{aligned} \mathbf{z}_i^T \delta \mathbf{x}_{i-1} &= 0 , \\ \mathbf{z}_i^T \delta \mathbf{x}_{i-2} &= 0 , \\ &\dots \dots \\ \mathbf{z}_i^T \delta \mathbf{x}_{i-n+1} &= 0 . \end{aligned}$$

This proves that \mathbf{z}_i is orthogonal to the previous $n - 1$ directions searched.

References

- [1] M.J.D. Powell, "A method for non-linear constraints in minimization problems," in: *Optimization, Symposium of the Institute of Mathematics and its Applications, Keele*, Ed. R. Fletcher (Academic Press, New York, 1969).
- [2] D.A. Pierre, *Optimization theory with applications* (Wiley, New York, 1969).
- [3] A.V. Fiacco and G.P. McCormick, *Nonlinear programming: sequential unconstrained minimization techniques* (Wiley, New York, 1968).
- [4] E.M. Rosen, "A review of quasi-Newton methods in nonlinear equation solving and unconstrained optimization," in: *Proceedings of the 21st National Conference of the Association for Computing Machinery* (Academic Press, New York, 1966) pp. 37-42.
- [5] J.G.P. Barnes, "An algorithm for solving non-linear equations based on the secant method," *Computer Journal* 8 (1965) 66-72.
- [6] B.A. Murtagh and R.W.H. Sargent, "A constrained minimization method with quadratic convergence," in: *Optimization, Symposium of the Institute of Mathematics and its Application, Keele*, Ed. R. Fletcher (Academic Press, New York, 1969).
- [7] B. Noble, *Applied linear algebra* (Prentice-Hall, Englewoods-Cliffs, N.J., 1969).
- [8] M.R. Chidambara, "On the inverse of certain matrices," *IEEE Transactions on Automatic Control*, Vol. AC-12 (1967) 214-215.