RICE UNIVERSITY

A FIRST-ORDER METHOD FOR THE EXTREMIZATION OF CONSTRAINED AND UNCONSTRAINED FUNCTIONS

by

SARWAR NAQVI

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

Thesis Director's signature:

Ho-Yi Huang

Houston, Texas
May 1971

Abstract

A First-Order Method for the Extremization

of Constrained and Unconstrained Functions

by

SARWAR NAQVI

The problem of extremizing a function $f(x)$ subject to the constraint $\varphi(x) = 0$ is considered. Here, f is a scalar, x an n-vector, and φ a q-vector, where $0 \leq q < n$. This problem is transformed into that of minimizing the unconstrained function $R(x, \lambda)$, where x and λ are regarded as independent variables. The q-vector λ is the Lagrange multiplier associated with the constraint and the function $R(x, \lambda)$ is the performance index measuring the cumulative error in the optimum condition and the constraint. The minimum $R(x, \lambda) = 0$ of the performance index is sought by applying quadratically convergent algorithms for unconstrained function minimization: the (n+q)-vector $y = [x, \lambda]^T$ is the independent variable associated with the performance index $R(y)$.

Since the performance index $R(y)$ involves the first derivatives f_x and φ_x, the gradient $G(y) = R_y(y)$, which is employed in quadratically convergent algorithms, involves the second derivatives f_{xx} and φ_{xx}. To avoid the explicit use of these second derivatives, a two-point determination of the gradient $G(y)$ is developed: the (n+q)-vector $G(y)$ is computed numerically through only two evaluations of the function $R(y)$.

Concerning the one-dimensional determination of the stepsize α, a two-point quasilinearization search is developed. This method requires only two evaluations of the function R(y), but preserves the eventual quadratic convergence of the quasilinearization method. Two terminating conditions are investigated: exact search and one-cycle search.

Thus, the method presented here is a first-order method. For the ideal case of a quadratic function subject to a linear constraint, it converges to the solution in n+q iterations, at most. The total computational effort involved is equivalent to, at most, 3(n+q) + 1 evaluations of the function R(y).

Three numerical examples are given using both the exact search and the one-cycle search. The results are presented in terms of number of iterations and number of function evaluations for convergence.

ACKNOWLEDGEMENTS

The author is indebted to Dr. H.Y. Huang for suggesting the topic and helpful discussions. This research was supported by the National Science Foundation, Grant No. GP-27271.

To my wife

Patricia E. Naqvi

TABLE OF CONTENTS

1. Introduction

In recent years, considerable success has been achieved in the area of Unconstrained function minimization using first-order methods having quadratic convergence properties (Refs. 1-10). In this thesis, a technique for transforming a constrained minimization problem into an equivalent unconstrained minimization problem is presented. In this way, the methods already developed for unconstrained function minimization can be employed without any change. Here, we restrict ourselves to using the first derivatives of the function and the constraint, at most. Therefore, the method given here is a first-order method.

2. Statement of the Problem

We consider the problem of extremizing the function

$$f = f(x) \tag{1}$$

subject to the constraint

$$\varphi(x) = 0 \tag{2}$$

In the above equations, f is a scalar, x an n-vector, and φ a q-vector, where $0 \leq q < n$. For the limiting case where $q = 0$, the problem becomes that of extremizing the unconstrained function (1).

---

[1] All vectors are column vectors.

3. Exact First-Order Conditions

From theory of maxima and minima, it is known that the previous problem can be recast as that of extremizing the augmented function

$$F(x, \lambda) = f(x) + \lambda^T \varphi(x) \tag{3}$$

subject to the constraint (2). Here, λ is a q-vector Lagrange multiplier and the superscript T denotes the transpose of a matrix. If

$$F_x(x, \lambda) = f_x(x) + \varphi_x(x)\lambda \tag{4}$$

denotes the gradient of the augmented function with respect to the vector x, the optimal solution x, λ must satisfy the simultaneous equations

$$F_x(x, \lambda) = 0 \quad , \quad \varphi(x) = 0 \tag{5}$$

In Eqs. (4)-(5), the gradient f_x and F_x are n-vectors and the matrix φ_x is n x q.

4. Performance Index

In general, the system (5) is so complicated that an analytical solution is not available. This being the case, an iterative method must be employed to obtain an approximate solution. In this connection, we introduce the performance index

$$R(x, \lambda) = Q(x, \lambda) + P(x) \tag{6}$$

where

$$Q(x, \lambda) = F_x^T(x, \lambda) F_x(x, \lambda) \quad , \quad P(x) = \varphi^T(x)\varphi(x) \tag{7}$$

Here, the functions $Q(x, \lambda)$ and $P(x)$ measure the errors in the optimum condition and the constraint, respectively. From Eq. (6), it is seen that the performance index $R(x, \lambda)$ represents the cumulative error in both the optimum condition and the constraint.

We observe that, since $R(x, \lambda)$ is the sum of n+q squares of functions dependent on x and λ, it is positive for any combination of x and λ which does not satisfy Eqs. (5) and it is zero for those combinations of x and λ which satisfy Eqs. (5). This observation leads to the conclusion that the solution x, λ of Eqs. (5) is an unconstrained minimal point for the performance index $R(x, \lambda)$, x and λ being considered as the independent variables.

5. Quadratically Convergent Algorithms

From the previous discussion, we see that the solution of the system (5) can be obtained by minimizing the performance index $R(x, \lambda)$, where x and λ are regarded to be independent variables. This being the case, any of the quadratically convergent algorithms for unconstrained function minimization summarized in Refs. 9-10 can be employed. However, in the discussion that follows, we present only three algorithms in detail.

Before presenting the three algorithms, we introduce the variable

$$y = \begin{bmatrix} x \\ \lambda \end{bmatrix} \tag{8}$$

and the function

$$r(y) = \begin{bmatrix} F_x(x, \lambda) \\ \varphi(x) \end{bmatrix} \tag{9}$$

so that the performance index (6) becomes

$$R(y) = r^T(y)r(y) \tag{10}$$

Here, y and r are (n+q)-vectors. If G(y), an (n+q)-vector, denotes the gradient of the performance index R(y) with respect to the vector y, that is,

$$G(y) = R_y(y) \tag{11}$$

a minimizing algorithm is represented by the relations

$$p_i = H_iG_i \quad , \quad \Delta y_i = -\alpha_i p_i \quad , \quad y_{i+1} = y_i + \Delta y_i \tag{12}$$

where the subscript i denotes the present point and the subscript i+1 denotes the next point. In Eqs.(12), the (n+q)-vector p_i is the search direction; the (n+q) x (n+q) matrix H_i characterizes a particular algorithm; the (n+q)-vector Δy_i is the displacement leading from the present point y_i to the next point y_{i+1}; and α_i is the stepsize. The sequence of operations is as follows: (a) at a given point y_i, the gradient G_i is computed; (b) the matrix H_i is defined according to different algorithms; (c) the search direction p_i is computed according to Eq. (12-1); (d) the stepsize α_i is determined by a one-dimensional search on the function $R(y_{i+1}) = R(y_i - \alpha_i p_i)$; in this search, the function $R(y_i - \alpha_i p_i)$ is minimized with respect to α_i; (e) the displacement Δy_i is computed according to Eq. (12-2); and (f) the next point y_{i+1} is obtained from Eq. (12-3).

Operations (a)-(f) form a complete iteration leading from the present point y_i to the next point y_{i+1}, providing the matrix H_i is specified. This is given below for three particular algorithms.

Algorithm I. At the initial point x_o, the initial matrix H_o is defined as an (n+q) x (n+q) identity matrix, that is

$$H_o = I \tag{13}$$

For any other point except a restarting point, the matrix H_i is updated by the formula

$$H_i = H_{i-1} + \frac{(\Delta y_{i-1} - H_{i-1}\Delta G_{i-1})(\Delta y_{i-1} - H_{i-1}\Delta G_{i-1})^T}{(\Delta y_{i-1} - H_{i-1}\Delta G_{i-1})^T \Delta G_{i-1}} \tag{14}$$

where

$$\Delta G_{i-1} = G_i - G_{i-1} \tag{15}$$

denotes the difference in gradients. According to the analyses and experiments of Refs. 9-10, it is advantageous to restart this algorithm only when the inequality

$$|G_i^T p_i| \le \varepsilon_1 \tag{16}$$

is satisfied at a given point x_i, where ε_1 is a small preselected number. In this case, Eq. (14) is replaced by

$$H_i = H_o \tag{17}$$

<u>Algorithm II.</u> At the initial point x_o, the initial matrix H_o is defined by Eq. (13). For any other point except a restarting point, the matrix H_i is updated by

$$H_i = H_{i-1} - \frac{H_{i-1}\Delta G_{i-1}\Delta G_{i-1}^T H_{i-1}}{\Delta G_{i-1}^T H_{i-1}\Delta G_{i-1}} \tag{18}$$

This algorithm must be restarted every n + q iterations.

<u>Algorithm III.</u> At the initial point x_o, the initial matrix H_o is defined by Eq. (13). For any other point except a restarting point, the matrix H_i is defined by

$$H_i = I + \frac{p_{i-1}G_i^T}{G_{i-1}^T G_{i-1}} \tag{19}$$

This algorithm is restarted every n+q + 1 iterations.

6. Gradient of the Performance Index

As described in the previous section, the application of quadratically convergent algorithms to the minimization of the performance index R(y) requires the explicit use of its gradient G(y). From Eqs. (10)-(11), this gradient can be written as

$$G(y) = 2r_y(y)r(y) \tag{20}$$

where r_y is an (n+q) x (n+q) matrix. In the light of definitions (4), (8), (9), the explicit form of the matrix $r_y(y)$ is given by

$$r_y(y) = \begin{bmatrix} f_{xx}(x) + \varphi_{xx}(x)\lambda & \varphi_x(x) \\ \varphi_x^T(x) & 0 \end{bmatrix} \tag{21}$$

and the gradient G(y) becomes

$$G(y) = 2 \begin{bmatrix} f_{xx}(x) + \varphi_{xx}(x)\lambda & \varphi_x(x) \\ \varphi_x^T(x) & 0 \end{bmatrix} \begin{bmatrix} f_x(x) + \varphi_x(x)\lambda \\ \varphi(x) \end{bmatrix} \tag{22}$$

Since the calculation of the gradient G(y) through Eq. (22) requires the second derivatives f_{xx} and φ_{xx}, it follows that use of Eq. (22) should be avoided if a first-order method is desired.

An elementary way to avoid the explicit use of second derivatives is to use a standard finite-difference scheme to determine the gradient G(y) numerically. Since G(y) is an (n+q)-vector, this scheme requires n+q+1 evaluations of the performance index R(y). Because of this large number of evaluations, the resulting algorithm would be computationally inefficient.

6.1. Two-Point Determination of the Gradient G(y). To reduce the computing time for the numerical determination of the gradient G(y), an efficient method is needed and is presented here. First, consider the given point y and compute the vector r(y) according to Eq. (9). Next, consider the particular varied point defined by

$$\tilde{y} = y + \beta r(y) \tag{23}$$

and compute the vector $r(\tilde{y})$ according to Eq. (9). If the prescribed number β is sufficiently small, the following first-order expansion is valid:

$$r(\tilde{y}) = r(y) + \beta r_y^T(y) r(y) \tag{24}$$

From Eq. (21), it is seen that the matrix r_y is symmetric. This being the case, Eq. (24) can be rewritten as

$$r(\tilde{y}) = r(y) + \beta r_y(y) r(y) \tag{25}$$

From Eqs. (20) and (25), it becomes clear that the gradient G(y) can be approximated by

$$G(y) = (2/\beta)[r(\tilde{y}) - r(y)] \tag{26}$$

Thus, the numerical determination of the gradient G(y) needs only the calculation of r(y) at two points y and $\tilde{y}$ related by (23). Since the performance index R(y) is related to the vector r(y) by Eq. (10), the effort involved here is equivalent to only two evaluations of the performance index R(y).

In closing, the following remarks are pertinent: (a) if the factor β in Eq. (26) is taken as

$$\beta = \epsilon_2/\sqrt{R(y)} \tag{27}$$

where ϵ_2 is a preselected small number, the order of magnitude of the displacement $\tilde{y} - y$ defined by (23) is the same as that of ϵ_2; and (b) if the function f(x) is quadratic and the constraint $\varphi(x)$ is linear, the approximation (26) for the gradient G(y) becomes exact and independent of the factor β.

7. One-Dimensional Search

If Eqs. (23), (26), (27) are used, the gradient G(y) can be computed using first derivatives, at most. Therefore, the search direction p can be computed in terms of first derivatives, at most. In addition, if the stepsize α can be determined employing first derivatives at most, the algorithms described in Section 5 can be regarded to be first-order algorithms. In this section, we compute the stepsize α by a version of the quasilinearization method which uses first derivatives, at most.

7.1. Standard Quasilinearization Search. The first-order condition for the minimization of the function

$$\tilde{\Phi}(\alpha) = R(y - \alpha p) \tag{28}$$

with respect to α is

$$\tilde{\Phi}_{\alpha}(\alpha) = 0 \tag{29}$$

Starting from a nominal value α such that $\tilde{\Phi}_{\alpha}(\alpha) \neq 0$, we would like to find a correction $\Delta\alpha$ such that, at

$$\tilde{\alpha} = \alpha + \Delta\alpha \tag{30}$$

the relation $\tilde{\Phi}_{\alpha}(\tilde{\alpha}) = 0$ holds. If the correction $\Delta\alpha$ is sufficiently small, the following first-order expansion is valid:

$$\tilde{\Phi}_{\alpha}(\tilde{\alpha}) = \tilde{\Phi}_{\alpha}(\alpha) + \tilde{\Phi}_{\alpha\alpha}(\alpha)\Delta\alpha = 0 \tag{31}$$

Therefore, the correction $\Delta\alpha$ is obtained from

$$\Delta\alpha = - \tilde{\Phi}_{\alpha}(\alpha)/\tilde{\Phi}_{\alpha\alpha}(\alpha) \tag{32}$$

From the definitions (10) and (28), the first and second derivatives of $\bar{\varphi}(\alpha)$ are given by

$$\bar{\varphi}_{\alpha}(\alpha) = -2r^T(y - \alpha p)r_y^T(y - \alpha p)p$$
$$\bar{\varphi}_{\alpha\alpha}(\alpha) = 2p^T[r_y(y - \alpha p)r_y^T(y - \alpha p) + r_{yy}(y - \alpha p)r(y - \alpha p)]p \tag{33}$$

The direct implementation of this method has the following drawbacks: (i) the evaluation of $\bar{\varphi}_{\alpha}$ and $\bar{\varphi}_{\alpha\alpha}$ by Eq. (33) requires the explicit use of the second and third derivatives of the function and the constraint; and (ii) the stability of the search or descent property on the function $\bar{\varphi}(\alpha)$ is not ensured. Concerning (ii), the first-order change of the function $\bar{\varphi}(\alpha)$ is given by

$$\delta\bar{\varphi} \cong \bar{\varphi}(\tilde{\alpha}) - \bar{\varphi}(\alpha) = \bar{\varphi}_{\alpha}(\alpha)\Delta\alpha \tag{34}$$

and, in the light of (32), becomes

$$\delta\bar{\varphi} = -\bar{\varphi}_{\alpha}^{2}(\alpha)/\bar{\varphi}_{\alpha\alpha}(\alpha) \tag{35}$$

If $\bar{\varphi}_{\alpha\alpha}(\alpha)$ is negative, then $\delta\bar{\varphi}$ becomes positive, meaning that the correction $\Delta\alpha$ is in the direction of increasing $\bar{\varphi}$ rather than decreasing $\bar{\varphi}$. In addition, even if $\bar{\varphi}_{\alpha\alpha}(\alpha)$ is positive, $\Delta\alpha$ may be so large that, owing to excessive overshooting of the minimal point, $\bar{\varphi}(\tilde{\alpha}) > \bar{\varphi}(\alpha)$. To overcome these difficiulities, we suggest the following modification.

7.2. Two-Point Quasilinearization Search. Since $\Delta\alpha$ may be too large, we introduce a scaling factor μ so that $\Delta\alpha$ is redefined as

$$\Delta\alpha = -\mu\bar{\varphi}_{\alpha}(\alpha)/\bar{\varphi}_{\alpha\alpha}(\alpha) \ , \quad 0 \leq \mu \leq 1 \tag{36}$$

Thus, if $\Phi_{\alpha\alpha} > 0$, the scaling factor μ is determined by a bisection process, starting from $\mu = 1$, until the inequality $\Phi(\tilde{\alpha}) < \Phi(\alpha)$ is satisfied.

In order to avoid computing the second and third derivatives of the function and the constraint in the evaluation of Φ_α and $\Phi_{\alpha\alpha}$, we approximate the second derivative $\Phi_{\alpha\alpha}$ as follows:

$$\Phi_{\alpha\alpha}(\alpha) = 2p^T r_y(y - \alpha p) r_y^T(y - \alpha p)p \tag{37}$$

that is, we neglect the second term on the right-hand side of Eq. (33-2). This approximation is justified for two reasons: (i) in the neighborhood of the solution, r is small, and the matrix $r_{yy}r$ is negligible with respect to the matrix $r_y r_y^T$; and (ii) away from the solution, the approximation (37) ensures the descent property on the function Φ, since $\Phi_{\alpha\alpha}$ defined by (37) is always positive.

The next step is to determine the derivatives Φ_α and $\Phi_{\alpha\alpha}$ by a numerical method which uses, at most, the first derivatives of the function and the constraint. First, consider a given value α, compute $y - \alpha p$ and evaluate the vector $r(y - \alpha p)$ according to Eq. (9). Next, increase α by the amount γ, compute

$$\tilde{\tilde{y}} = y - (\alpha + \gamma)p \tag{38}$$

and evaluate the vector $r(\tilde{\tilde{y}}) = r(y - \alpha p - \gamma p)$. In Eq. (38), γ is a prescribed number. If the factor γ is sufficiently small, the following first-order expansion is valid:

$$r(\tilde{\tilde{y}}) = r(y - \alpha p) - \gamma r_y^T(y - \alpha p)p \tag{39}$$

so that

$$r_y^T(y - \alpha p)p = (1/\gamma)[r(y - \alpha p) - r(y - \alpha p - \gamma p)] \tag{40}$$

Substitution of (40) into (33-1) yields the first derivative Φ_α, while substitution of (40) into (37) yields the approximate second derivative $\Phi_{\alpha\alpha}$. Thus, except for the scaling factor μ, the correction $\Delta\alpha$ in Eq. (36) is determined by two evaluations of the vector r at points $y - \alpha p$ and $y - (\alpha + \gamma)p$. Since the performance index R is related to the vector r by Eq. (10), the effort involved here is equivalent to only two evaluations of the performance index R.

In closing, the following remarks are pertinent: (a) if the factor γ in Eq. (40) is taken as

$$\gamma = \varepsilon_3 / \sqrt{p^T p} \tag{41}$$

where ε_3 is a preselected small number, the order of magnitude of the displacement $-\gamma p$ is the same as that of ε_3; and (b) if the function f(x) is quadratic and the constraint $\varphi(x)$ is linear, the approximations (37) and (40) become exact and independent of the factor γ.

8. Summary of Algorithms

The previous technique employs, at most, the first derivatives of the function and the constraint; it can be summarized as follows. First, we form the exact first-order conditions and define the performance index R(y) by Eq. (10). The performance index R(y) is then minimized by any of the quadratically convergent algorithms in Section 5; the gradient G(y) is determine by the two-point approximation given in Section 6.1 and the stepsize α is determined by the two-point quasilinearization search given in Section 7.2. In addition, the following considerations are in order.

8.1. Stopping Condition for the Algorithms. The minimization of the performance index R(y) is considered achieved if

$$G_i^T G_i \le \varepsilon_4 \tag{42}$$

where ε_4 is a preselected small number. Two different cases are possible, depending on whether

$$R_i \le \varepsilon_5 \quad \text{or} \quad R_i > \varepsilon_5 \tag{43}$$

where ε_5 is a preselected small number. If Ineq. (43-1) is satisfied, the converged solution represents the solution of the first-order conditions (5). On the other hand, if Ineq. (43-2) is satisfied, the converged solution is only a relative minimum of the performance index R(y) and does not represent the solution of the first-order condition (5). In this case, a new nominal point y must be employed and the algorithm must be started anew.

8.2. Stopping Condition for the Search. Two different ways of stopping the one-dimensional search are given below.

Exact Search. For a more precise determination of the minimum of the function $\tilde{\varphi}(\alpha) = R(y - \alpha p)$ with respect to α, the two-point quasilinearization search can be employed iteratively until one of the following conditions is satisfied:

$$|\tilde{\varphi}_\alpha(\alpha)| \leq \epsilon_1 \quad \text{or} \quad |\Delta\alpha| \leq \epsilon_6 \, |\alpha| \quad \text{or} \quad N_s = N_{s*} \tag{44}$$

where ϵ_1 and ϵ_6 are preselected small numbers, N_s is the number of search iterations, and N_{s*} is the preselected upper limit for N_s. Condition (44-1) defines the ideal case where the minimum $\tilde{\varphi}(\alpha)$ is reached within a desired accuracy. Conditions (44-2) and (44-3) are merely measures to prevent the search from going indefinitely.

One-Cycle Search. Quadratically convergent algorithms exhibit rapid convergence only when the point under consideration is in the neighborhood of the minimum, that is, only when R(y) can be approximated fairly well by a quadratic function; otherwise, the algorithms give no assurance of rapid convergence. Away from the minimal point, the exact search is unnecessary, since the satisfaction of the conjugacy conditions for the search direction is not important. Near the minimal point, the two-point quasilinearization search ensures one-step convergence to the minimum of $\tilde{\varphi}(\alpha)$. Therefore, the total computational effort can be reduced, while terminal convergence properties can be preserved, if the exact search is replaced by a one-cycle search. That is, one sets

$$N_{s*} = 1 \tag{45}$$

in condition (44-3).

9. Numerical Examples

In this section, three examples[2] of constrained function extremization problems are investigated using a Burroughs B-5500 computer and double precision arithmetic. For simplicity, all the symbols used here are scalar quantities.

Example 9.1. Minimize the function

$$f = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^4 \tag{46}$$

subject to the constraint

$$\varphi = x_1(1 + x_2^2) + x_3^4 - 4 - 3\sqrt{2} = 0 \tag{47}$$

The solution is $f = 0.3256 \times 10^{-1}$ at the point defined by

$$x_1 = 1.1048 \quad , \quad x_2 = 1.1966 \quad , \quad x_3 = 1.5352 \tag{48}$$

and the related multiplier is

$$\lambda = -0.1072 \times 10^{-1} \tag{49}$$

The nominal coordinates and multiplier chosen for starting an algorithm are

$$x_1 = 2 \quad , \quad x_2 = 2 \quad , \quad x_3 = 2 \quad , \quad \lambda = 0 \tag{50}$$

Example 9.2. Minimize the function

$$f = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_3 - 1)^2 + (x_4 - 1)^4 + (x_5 - 1)^6 \tag{51}$$

---

[2] These examples are taken from Refs. 11-12.

subject to the constraints

$$\varphi_1 = x_1^2 x_4 + \sin(x_4 - x_5) - 2\sqrt{2} = 0 \quad , \quad \varphi_2 = x_2 + x_3^4 x_4^2 - 8 - \sqrt{2} = 0 \tag{52}$$

The solution is f = 0.2415 at the point defined by

$$x_1 = 1.1661 \; , \; x_2 = 1.1821 \; , \; x_3 = 1.3802 \; , \; x_4 = 1.5060 \; , \; x_5 = 0.6109 \tag{53}$$

and the related multipliers are

$$\lambda_1 = -0.8553 \times 10^{-1} \quad , \quad \lambda_2 = -0.3187 \times 10^{-1} \tag{54}$$

The nominal coordinates and multipliers chosen for starting an algorithm are

$$x_1 = 2 \; , \; x_2 = 2 \; , \; x_3 = 2 \; , \; x_4 = 2 \; , \; x_5 = 2 \; , \; \lambda_1 = 0 \; , \; \lambda_2 = 0 \tag{55}$$

Example 9.3. Minimize the function

$$f = (x_1 - 1)^2 + (x_1 - x_2)^2 + (x_2 - x_3)^2 + (x_3 - x_4)^4 + (x_4 - x_5)^4 \tag{56}$$

subject to the constraint

$$\varphi_1 = x_1 + x_2^2 + x_3^3 - 2 - 3\sqrt{2} = 0 \; , \quad \varphi_2 = x_2 - x_3^2 + x_4 + 2 - 2\sqrt{2} = 0$$

$$\varphi_3 = x_1 x_5 - 2 = 0 \tag{57}$$

The solution is $f = 0.7877 \times 10^{-1}$ at the point defined by

$$x_1 = 1.1911 \; , \; x_2 = 1.3626 \; , \; x_3 = 1.4728 \; , \; x_4 = 1.6350 \; , \; x_5 = 1.6790 \tag{58}$$

and the related multipliers are

$$\lambda_1 = -0.3882 \times 10^{-1} \ , \quad \lambda_2 = -0.1672 \times 10^{-1} \ , \quad \lambda_3 = -0.2872 \times 10^{-3} \qquad (59)$$

The nominal coordinates and multipliers chosen for starting an algorithm are

$$x_1 = 2 \ , \ x_2 = 2 \ , \ x_3 = 2 \ , \ x_4 = 2 \ , \ x_5 = 2 \ , \ \lambda_1 = 0 \ , \ \lambda_2 = 0 \ , \ \lambda_3 = 0 \qquad (60)$$

Experimental Conditions. The small numbers ε_1, ε_2, ε_3, ε_4, ε_5, ε_6 appearing in (16), (27), (41), (42), (43), and (44-2) are chosen to be

$$\varepsilon_1 = 10^{-16} \ , \quad \varepsilon_2 = 10^{-9} \ , \quad \varepsilon_3 = 10^{-9}$$

$$\varepsilon_4 = 10^{-12} \ , \quad \varepsilon_5 = 10^{-10} \ , \quad \varepsilon_6 = 10^{-3} \qquad (61)$$

For the exact search, the upper limit N_{s*} to the number of search iterations is chosen to be

$$N_{s*} = 10 \qquad (62)$$

Experimental Results. The computational results pertaining to Algorithms I-III of Section 5 are given in Tables 1-3, where N_* denotes the number of iterations for convergence and K_* denotes the total number of evaluations of the function R(y) or evaluations of the vector r(y). This number is determined by the formula

$$K_* = \sum_{i=0}^{N_*-1} Z_i + N_* + 1 \qquad (63)$$

where Z_i is the number of evaluations of the function R(y) or evaluations of the vector r(y) needed for the one-dimensional search associated with the ith iteration.

Equation (63) is based on an ideally written program where the function R(y) or its equivalent r(y) is calculated only once at a given point. This means that, if necessary, the function R(y) or its equivalent r(y) must be transferred from the main program to the search subroutine, and vice versa. Clearly, the number K_* gives a measure of the total computational effort needed for convergence.

10. Conclusions

In this thesis, the problem of extremizing a function $f(x)$ subject to the constraint $\varphi(x) = 0$ is considered. Here, f is a scalar, x an n-vector, and φ a q-vector, where $0 \leq q < n$. This constrained problem is transformed into that of minimizing the unconstrained function $R(x, \lambda)$, where x and λ are regarded as independent variables. The q-vector λ is the Lagrange multiplier associated with the constraint and the function $R(x, \lambda)$ is the performance index measuring the cumulative error in both the optimum condition and the constraint. The minimum $R(x, \lambda) = 0$ of the performance index is sought by applying quadratically convergent algorithms for unconstrained function minimization: the (n+q)-vector $y = [x, \lambda]^T$ is the independent variable associated with the performance index $R(y)$.

Since the performance index $R(y)$ involves the first derivatives f_x and φ_x, the gradient $G(y) = R_y(y)$, which is employed in quadratically convergent algorithms, involves the second derivatives f_{xx} and φ_{xx}. To avoid the explicit use of these second derivatives an efficient <u>two-point determination of the gradient $G(y)$</u> is developed: the (n+q)-vector $G(y)$ is computed numerically through only two evaluations of the function $R(y)$.

Concerning the one-dimensional determination of the stepsize α, a two-point quasilinearization search is developed. This method requires only two evaluations of the function $R(y)$, but preserves the eventual quadratic convergence of the quasilinearization method.

Thus, the method presented here is a first-order method. For the ideal case of a quadratic function subject to a linear constraint, it converges to the solution in n+q iterations, at most. The total computational effort involved is equivalent to, at most, 3(n+q) + 1 evaluations of the function R(y).

To illustrate the method, three numerical examples are investigated. For each example, three quadratically convergent algorithms are employed: Algorithm I is representative of the class of algorithms which, at convergence, yields the matrix $H = R_{yy}^{-1}(y)$; Algorithm II is representative of the class of algorithms which, at convergence, yields the matrix H = 0; and Algorithm III is a simplified algorithm or conjugate-gradient algorithm. Two terminating conditions for the one-dimensional search are presented: exact search and one-cycle search. The numerical results are given in terms of number of iterations and number of function evaluations for convergence (Tables 1-3).

In closing, the following comments are in order: (i) Algorithm I is superior to both Algorithm II and Algorithm III in terms of number of iterations and function evaluations for convergence; this characteristic is also noted in Refs. 9-10; (ii) the exact search requires less iterations for convergence but more function evaluations than the one-cycle search; since the number of function evaluations is a better measure of the total computational effort, the one-cycle search should be preferred; and (iii) as evidenced by Example 9.2 in Tables 2-3, the one-cycle search may not be as stable as the exact search, in the sense that it may not converge to the extremum which is nearest to the starting nominal point.

Table 1. Number of iterations and number of function evaluations for convergence (Algorithm I).

Example	Exact Search		One-Cycle Search	
	N_*	K_*	N_*	K_*
9.1	17	138	15	46
9.2	20	141	39	118
9.3	17	105	20	61

Table 2. Number of iterations and number of function evaluations for convergence (Algorithm II).[3]

Example	Exact Search		One-Cycle Search	
	N_*	K_*	N_*	K_*
9.1	24	182	20	61
9.2	28	167	(40)	(122)
9.3	33	185	48	145

[3] For Example 9.2, one-cycle search, Algorithm II does not converge to the solution (53)-(54) but to a different solution, namely, $f = 4.6025$ at the point defined by the coordinates $x_1 = -1.0287$, $x_2 = -1.0172$, $x_3 = 1.3545$, $x_4 = 1.7603$, $x_5 = 0.4531$ and the multipliers $\lambda_1 = -1.1266$ and $\lambda_2 = -0.2301 \times 10^{-1}$.

Table 3. Number of iterations and number of function evaluations for convergence (Algorithm III).

Example	Exact Search		One-Cycle Search	
	N_*	K_*	N_*	K_*
9.1	24	165	29	89
9.2	58	336	>200	>601
9.3	46	227	39	118

References

1. HESTENES, M.R., and STIEFEL, E., Methods of Conjugate Gradients for Solving Linear Systems, Journal of Research of the National Bureau of Standards, Vol. 49, No. 6, 1952.

2. DAVIDON, W.C., Variable Metric Method for Minimization, Argonne National Laboratory, Report No. ANL-5990, 1959.

3. FLETCHER, R., and POWELL, M.J.D., A Rapidly Convergent Descent Method for Minimization, Computer Journal, Vol. 6, No. 2, 1963.

4. FLETCHER, R., and REEVES, C.M., Function Minimization by Conjugate Gradients, Computer Journal, Vol. 7, No. 2, 1964.

5. PEARSON, J.D., On Variable Metric Methods of Minimization, Research Analysis Corporation, Technical Paper No. RAC-TP-302, 1968.

6. SORENSON, H.W., Comparison of Some Conjugate Direction Procedures for Function Minimization, Journal of theFranklin Institute, Vol. 288, No. 6, 1969.

7. MURTAGH, B.A., and SARGENT, R.W.H., Computational Experience with Quadratically Convergent Minimization Methods, Computer Journal, Vol. 13, No. 2, 1970.

8. FLETCHER, R., A New Approach to Variable Metric Algorithms, Computer Journal, Vol. 13, No. 3, 1970.

9. HUANG, H.Y., Unified Approach to Quadratically Convergent Algorithms for Function Minimization, Journal of Optimization Theory and Applications, Vol. 5, No. 6, 1970.

10. HUANG, H.Y., and LEVY, A.V., Numerical Experiments on Quadratically Convergent Algorithms for Function Minimization, Journal of Optimization Theory and Applications, Vol. 6, No. 3, 1970.

11. MIELE, A., HEIDEMAN, J.C., and LEVY, A.V., Mathematical Programming for Constrained Minimal Problems, Part 3, Combined Gradient-Restoration Algorithm, Rice University, Aero-Astronautics Report No. 68, 1970.

12. HEIDEMAN, J.C., and LEVY, A.V., Mathematical Programming for Constrained Minimal Problems, Part 4, Combined Gradient-Restoration Algorithm: Further Numerical Examples, Rice University, Aero-Astronautics Report No. 69, 1970.