70-23,521

HEIDEMAN, John Clark, 1944-
SEQUENTIAL CONJUGATE GRADIENT-RESTORATION ALGORITHM FOR THE MINIMIZATION OF CONSTRAINED FUNCTIONS.

Rice University, Ph.D., 1970
Mathematics

University Microfilms, A XEROX Company, Ann Arbor, Michigan

RICE UNIVERSITY

SEQUENTIAL CONJUGATE GRADIENT-RESTORATION ALGORITHM FOR THE MINIMIZATION OF CONSTRAINED FUNCTIONS

by

JOHN C. HEIDEMAN

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Thesis Director's signature

Houston, Texas

April, 1970

Abstract

SEQUENTIAL CONJUGATE GRADIENT-RESTORATION ALGORITHM FOR THE MINIMIZATION OF CONSTRAINED FUNCTIONS

by

JOHN C. HEIDEMAN

The problem of minimizing a function $f(x)$ subject to a constraint $\varphi(x) = 0$ is considered. Here, f is a scalar, x an n-vector, and φ a q-vector. A sequential algorithm is presented, made up of the alternate succession of gradient phases and restoration phases. In the gradient phase, a nominal point x satisfying the constraint is assumed; a displacement Δx leading from point x to a varied point y is determined such that the value of the function is reduced. The determination of the displacement Δx incorporates information at point x as well as information at the previous point $\hat{x}$. In the restoration phase, a nominal point y not satisfying the constraint is assumed; a displacement Δy leading from point y to a varied point $\tilde{x}$ is determined such that the constraint is restored to a prescribed degree of accuracy. The restoration is done by requiring the least-square change of the coordinates.

If the stepsize α of the gradient phase is of $O(\varepsilon)$, then $\Delta x = O(\varepsilon)$ and $\Delta y = O(\varepsilon^2)$. For ε sufficiently small, the restoration phase preserves the descent property of the gradient phase: the function f decreases between any two successive restoration phases. The sequential algorithm possesses quadratic convergence in the neighborhood of the constrained minimum. In particular, for a quadratic function subject to a linear constraint, the algorithm yields the minimum point in no more than n-q iterations.

ACKNOWLEDGMENT

The author is indebted to Professors Angelo Miele and Ho-Yi Huang for their assistance in the preparation of this thesis. Also, he wishes to acknowledge the support of the Office of Scientific Research, Office of Aerospace Research, United States Air Force, Grant No. AF-AFOSR-828-67.

TABLE OF CONTENTS

1. Introduction

In recent years, considerable attention has been given to the iterative algorithms for minimizing a function $f(x)$ subject to the constraint $\varphi(x) = 0$, where f is a scalar, x an n-vector, and φ a q-vector. With reference to the gradient method, one possible approach is that of the penalty functions. The advantage of this approach is that the constrained minimal problem is replaced by a mathematically simpler, unconstrained minimal problem. The disadvantages are these: no clear-cut method exists for choosing the penalty constants; the algorithm must be repeated several times for increasing values of the penalty constants; the values of the function between iterations are not comparable, since the constraints are not satisfied; and, even when the algorithm is terminated, the constraints are generally not satisfied.

In this thesis, we present a sequential algorithm constructed in such a way that the values of the function $f(x)$ between iterations are comparable. The algorithm is composed of the alternate succession of gradient phases and restoration phases (Fig. 1). In the gradient phase, a nominal point x satisfying the constraint is assumed; then, a displacement Δx leading from point x to a varied point y is determined by minimizing the first variation $\delta f(x)$ subject to the linearized constraint $\delta\varphi(x) = 0$ and a quadratic constraint on Δx. The determination of Δx incorporates information at point x as well as information at the previous point $\hat{x}$. Due to the fact that the constraint is accounted for only to first order, the varied point y may be such that $\varphi(y) \neq 0$. This being the case, a restoration phase is needed prior to starting the next gradient phase. In this <u>restoration phase</u>, a nominal point y not satisfying the constraint is assumed; then, a displacement Δy leading from point y to a varied point $\tilde{x}$ is determined by minimizing the distance $\Delta y^T \Delta y$ subject to the linearized constraint $k\varphi(y) + \delta\varphi(y) = 0$, where $0 \leq k \leq 1$. If a single restoration cycle fails to produce the required degree of accuracy in the constraint, several cycles must be employed, as shown in Fig. 2. After the restoration phase is finished, the iteration is completed, and the next iteration is started using $\tilde{x}$ as the nominal point x.

2. Statement of the Problem

We consider the problem of minimizing the function

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

subject to the constraint

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

In the above equation, f is a scalar, x an n-vector, and φ a q-vector[1], where $q < n$. It is assumed that the first and second partial derivatives of the function f and the constraint φ with respect to x exist and are continuous; it is also assumed that the constrained minimum exists.

[1] All the vectors in this thesis are column vectors.

3. Gradient Phase: General Discussion

Consider a displacement Δx leading from the nominal point x to the varied point y such that

$$y = x + \Delta x \tag{3}$$

Assume that the nominal point x satisfies (2) exactly and that the varied point y satisfies (2) to first order. The first-order change of the function (1) is given by

$$\delta f(x) = f_x^T(x)\Delta x \tag{4}$$

where $f_x(x)$ is the gradient of the scalar function f with respect to the vector x and the symbol T denotes the transpose of a matrix. In turn, the first-order change of the constraint (2) is represented by

$$\delta\varphi(x) = \varphi_x^T(x)\Delta x = 0 \tag{5}$$

where $\varphi_x(x)$, an n x q matrix, denotes the gradient of the vectorial function φ with respect to the vector x.

Next, consider the following quadratic constraint on the displacement Δx:

$$K = (\Delta x - \beta\Delta\hat{x})^T(\Delta x - \beta\Delta\hat{x}) \tag{6}$$

where K and β are constants and $\Delta\hat{x}$ is the displacement of the previous gradient phase, that is, the displacement leading from the nominal point $\hat{x}$ to the varied point $\hat{y}$ (see Figs. 1-2). With this understanding, we formulate the following problem: Find the displacement Δx which minimizes (4) subject to (5)-(6).

3.1. Displacement Δx. Standard methods of the theory of maxima and minima show that the fundamental function of this problem is the scalar function

$$\Omega = f_x^T(x)\Delta x + \lambda^T \varphi_x^T(x)\Delta x + (1/2\alpha)(\Delta x - \beta\Delta\hat{x})^T(\Delta x - \beta\Delta\hat{x}) \tag{7}$$

where $1/2\alpha$ is a scalar Lagrange multiplier and λ a q-vector Lagrange multiplier. If one introduces the augmented function

$$F(x,\lambda) = f(x) + \lambda^T\varphi(x) \tag{8}$$

and observes that

$$F_x(x,\lambda) = f_x(x) + \varphi_x(x)\lambda \tag{9}$$

the fundamental function (7) becomes

$$\Omega = F_x^T(x,\lambda)\Delta x + (1/2\alpha)(\Delta x - \beta\Delta\hat{x})^T(\Delta x - \beta\Delta\hat{x}) \tag{10}$$

In Eqs. (9)-(10), the symbol $F_x(x,\lambda)$ denotes the gradient of the augmented function F with respect to the vector x. The optimum displacement Δx satisfies the relation

$$\Omega_{\Delta x} = 0 \tag{11}$$

where $\Omega_{\Delta x}$ denotes the gradient of the fundamental function Ω with respect to the vector Δx. The explicit form of (11) is the following:

$$\Delta x = -\alpha F_x(x,\lambda) + \beta\Delta\hat{x} \tag{12}$$

If one defines the search direction p from

$$\Delta x = -\alpha p \tag{13}$$

and observes that, for the previous iteration,

$$\Delta\hat{x} = -\hat{\alpha}\hat{p} \tag{14}$$

the following relation ensues from (12)-(14):

$$p = F_x(x, \lambda) + \gamma\hat{p} \tag{15}$$

where

$$\gamma = \beta\hat{\alpha}/\alpha \tag{16}$$

In conclusion, the displacement Δx during the gradient phase is given by (13), with p governed by Eq. (15), where $\hat{p}$ is known from the previous iteration. Note that (13) and (15) determine Δx providing λ, α, γ are specified.

3.2. Relation between K and α. As Eq. (13) shows, the displacement Δx is proportional to α, the stepsize of the gradient phase. Upon substituting (12) into (6), we see that

$$K = \alpha^2 F_x^T(x, \lambda) F_x(x, \lambda) \tag{17}$$

Therefore, a correspondence exists between the values of the constant K and the values of the stepsize α. This being the case, one can bypass prescribing K and reason directly on α, as in the considerations which follow.

3.3. Determination of λ, α, γ. If Eqs. (5), (9), (13), (15) are combined, we obtain the relation

$$\varphi_x^T(x)f_x(x) + \varphi_x^T(x)\varphi_x(x)\lambda + \gamma\varphi_x^T(x)\hat{p} = 0 \tag{18}$$

which ensures satisfaction of the constraint (2) to first order. Equation (18) is a linear relation between λ and γ and admits the solution

$$\lambda = -[\varphi_x^T(x)\varphi_x(x)]^{-1}[\varphi_x^T(x)f_x(x) + \gamma\varphi_x^T(x)\hat{p}] \tag{19}$$

Therefore, the rate of change of λ with respect to γ is given by the vector equation

$$\lambda_\gamma = -[\varphi_x^T(x)\varphi_x(x)]^{-1}\varphi_x^T(x)\hat{p} \tag{20}$$

The next step is to assign values to α and γ. If Eqs. (3), (13), (15) are combined, the position vector at the end of the gradient phase becomes

$$y = x - \alpha F_x(x, \lambda) - \alpha\gamma\hat{p} \tag{21}$$

Since λ depends on γ through Eq. (19), Eq. (21) defines a two-parameter family of points y for which the augmented function F takes the form

$$F(y, \lambda) = F(x - \alpha F_x(x, \lambda) - \alpha\gamma\hat{p}, \lambda) = Z(\alpha, \gamma) \tag{22}$$

The greatest decrease of the function $Z(\alpha, \gamma)$ occurs if the parameters α, γ satisfy the following necessary conditions:

$$Z_\alpha(\alpha, \gamma) = 0 \quad , \quad Z_\gamma(\alpha, \gamma) = 0 \tag{23}$$

After observing that

$$Z_\alpha(\alpha,\gamma) = -F_x^T(y,\lambda)p$$

$$Z_\gamma(\alpha,\gamma) = -\alpha F_x^T(y,\lambda)\hat{p} - [\alpha F_x^T(y,\lambda)\varphi_x(x) - \varphi^T(y)]\lambda_\gamma \tag{24}$$

we see that Eqs. (23) can be written as

$$F_x^T(y,\lambda)p = 0$$

$$\alpha F_x^T(y,\lambda)\hat{p} + [\alpha F_x^T(y,\lambda)\varphi_x(x) - \varphi^T(y)]\lambda_\gamma = 0 \tag{25}$$

In the light of (15) and (19)-(21), Eqs. (25) constitute a system of two scalar equations in the unknowns α and γ. Once the stepsize α and the coefficient γ are known from (25), the multiplier λ follows from (19), the search direction p from (15), the displacement Δx from (13), and the position vector y from (3). Thus, the problem of determining λ, α, γ is solved in principle; computationally, however, further simplifications are needed to make the algorithm practical.

4. Gradient Phase: Quadratic Function, Linear Constraint

Now, consider the particular case of a quadratic function and a linear constraint given in the form

$$f(x) = a + b^T x + (1/2)x^T cx \quad , \quad \varphi(x) = d + e^T x \tag{26}$$

where a is a scalar, b is an n-vector, c an n x n symmetric matrix, d a q-vector, and e an n x q matrix. Here, all the coefficients are constant. The gradients of the functions f and φ become

$$f_x(x) = b + cx \quad , \quad \varphi_x(x) = e \tag{27}$$

Because of the linearity, the constraint is never violated during the gradient phase and, consequently,

$$\Delta y = 0 \quad , \quad \tilde{x} = x + \Delta x \quad , \quad \varphi_x^T(x)\hat{p} = 0 \tag{28}$$

This being the case, Eq. (19) supplies the following expression for the multiplier:

$$\lambda = -[\varphi_x^T(x)\varphi_x(x)]^{-1}\varphi_x^T(x)f_x(x) \tag{29}$$

which is now independent of γ, that is,

$$\lambda_\gamma = 0 \tag{30}$$

The relations (25) optimizing α and γ become

$$F_x^T(y, \lambda)p = 0 \quad , \quad F_x^T(y, \lambda)\hat{p} = 0 \tag{31}$$

showing that the gradient of the augmented function at point y is orthogonal to both the present and previous search directions. A mathematical consequence of (15) and (31) is that

$$F_x^T(y,\lambda)F_x(x,\lambda) = 0 \tag{32}$$

showing that the gradients at point y and point x are orthogonal. Furthermore, after laborious manipulations, Eqs. (31)-(32) lead to

$$F_x^T(\tilde{x},\tilde{\lambda})p = 0 \quad , \quad F_x^T(\tilde{x},\tilde{\lambda})\hat{p} = 0 \quad , \quad F_x^T(\tilde{x},\tilde{\lambda})F_x(x,\lambda) = 0 \tag{33}$$

For a quadratic function subject to a linear constraint, the following relationship can be shown to hold:

$$F_x(y,\lambda) = F_x(x,\lambda) - \alpha cp \tag{34}$$

Invoking (26)-(34), we see that (31-1) yields the following solution for the optimum stepsize:

$$\alpha = F_x^T(x,\lambda)F_x(x,\lambda)/p^T cp \tag{35}$$

Furthermore, (31-2) leads to

$$p^T c\hat{p} = 0 \tag{36}$$

which states that the search directions p and $\hat{p}$ are conjugate with respect to the matrix c. In turn, (36) yields the following explicit solution for γ:

$$\gamma = F_x^T(x,\lambda)F_x(x,\lambda)/F_x^T(\hat{x},\hat{\lambda})F_x(\hat{x},\hat{\lambda}) \tag{37}$$

In conclusion, for a given nominal point x, the multiplier λ is supplied by (29), the coefficient γ by (37), the search direction p by (15), the optimum step-size α by (35), the displacement Δx by (13), and the position vector y by (3).

4.1. Convergence Properties. For a quadratic function subject to a linear constraint, the following relations can be shown to hold providing the first step of the algorithm is a gradient step:

$$F_x^T(x,\lambda)F_x(x_*,\lambda_*) = 0 \quad , \quad F_x^T(x,\lambda)p_* = 0 \quad , \quad p^T c p_* = 0 \tag{38}$$

where x_* denotes any state preceding x. Equations (38) can be derived from (33) and (36) through mathematical induction. Equation (38-1) states that the gradient at each point is orthogonal to the gradient at every previous point. Equation (38-2) states that the gradient at each point is orthogonal to the search direction at every previous point. Finally, Eq. (38-3) states that the search direction at each point and the search direction at every previous point are conjugate with respect to the constant matrix c; this is why the algorithm is called the conjugate-gradient algorithm.

Furthermore, the gradient $F_x(x,\lambda)$ satisfies the equation $E = e^T F_x(x,\lambda) = 0$ at every step. Since the dimension of E is q while the dimension of $F_x(x,\lambda)$ is n, E = 0 admits only n-q linearly independent solution vectors $F_x(x,\lambda)$. Since Eq. (38-1) requires that the gradient at each point be linearly independent of the gradients at all previous points, it follows that the gradient $F_x(x,\lambda)$ at the end of n-q steps must be the null vector. That is, the minimum of f(x) subject to $\varphi(x) = 0$ is reached in n-q steps at most. If f(x) is unconstrained (q = 0), the minimum of f(x) is reached in n steps at most (see Refs. 1-2).

5. Gradient Phase: Nonquadratic Function and/or Nonlinear Constraint

If the function $f(x)$ is nonquadratic and/or the constraint $\varphi(x)$ is nonlinear, the relations (29), (35), (37) defining λ, α, γ are not simultaneously valid. We observe that (29) and (37) involve first derivatives only, while (35) involves the second derivative matrix c. This being the case, we choose to discard (35) and retain (29) and (37). Thus, we employ the algorithm

$$\lambda = -[\varphi_x^T(x)\varphi_x(x)]^{-1}\varphi_x^T(x)f_x(x)$$

$$\gamma = F_x^T(x,\lambda)F_x(x,\lambda)/F_x^T(\hat{x},\hat{\lambda})F_x(\hat{x},\hat{\lambda})$$

$$p = F_x(x,\lambda) + \gamma\hat{p} \qquad (39)$$

$$\Delta x = -\alpha p$$

$$y = x + \Delta x$$

which, for the unconstrained case, reduces to the well-known Fletcher-Reeves algorithm (Refs. 3-4). For the constrained case, the justification of the expressions for the multiplier λ and the coefficient γ is presented in later sections of this thesis. Once the nominal point x is given, the multiplier λ is determined with (39-1), the coefficient γ with (39-2), and the search direction p with (39-3); for a given stepsize α, the displacement Δx is computed with (39-4) and the varied point y with (39-5). The determination of α is discussed in the following section.

5.1. Optimum Stepsize. If Eqs. (39-4) and (39-5) are combined, the optimum vector at the end of the gradient phase becomes

$$y = x - \alpha p \tag{40}$$

For a given nominal point x, the vector p is known through Eqs. (39-1) through (39-3); therefore, Eq. (40) defines a one-parameter family of points y for which the augmented function $F(\bar{y}, \lambda)$ takes the form

$$F(y, \lambda) = F(x - \alpha p, \lambda) = Z(\alpha) \tag{41}$$

The greatest decrease in the function $Z(\alpha)$ occurs if the parameter α satisfies the following necessary condition:

$$Z_\alpha(\alpha) = 0 \tag{42}$$

After observing that

$$Z_\alpha(\alpha) = -F_x^T(y, \lambda)p \tag{43}$$

we see that Eq. (42) can be written as

$$F_x^T(y, \lambda)p = 0 \tag{44}$$

showing that the gradient of the augmented function at point y is orthogonal to the search direction p.

To obtain satisfaction of (42) or (44), some one-dimensional search method must be employed. In particular, cubic interpolation (it employs first derivatives only) and quasilinearization (it employs both first and second derivatives) are powerful methods (Ref. 4). These methods are to be employed iteratively until Eq. (42) is satisfied to a desired degree of accuracy, that is, until

$$Z_\alpha^2(\alpha) \le \theta_1 \tag{45}$$

where θ_1 is a small number. In practice, θ_1 can be fixed or one may choose

$$\theta_1 = \epsilon_1 Z_\alpha^2(0) \tag{46}$$

where ϵ_1 is a small number.

5.2. Starting the Algorithm. The algorithm (39) requires that the search direction $\hat{p}$ be known from the previous iteration. Since this is not the case for the first iteration, some assumption concerning $\gamma\hat{p}$ is needed in order to start the algorithm. In practice, one may choose $\hat{p} = 0$ or $\gamma = 0$.

5.3. Restarting the Algorithm. For a quadratic function subject to a linear constraint, the present algorithm converges to the exact minimum in no more than n-q steps. For the general case, this suggests the idea of restarting the algorithm every $\Delta N = n-q$ or $\Delta N = n-q+1$ steps.

6. Restoration Phase

At the end of the gradient phase, the point y is known. If the constraint is linear, the relation $\varphi(y) = 0$ holds. On the other hand, if the constraint is nonlinear, $\varphi(y) \neq 0$, which means that some degree of dissatisfaction exists. Therefore, a restoration phase is needed prior to starting the next gradient phase. Specifically, one has to apply a small variation Δy to y to generate a new position vector

$$\tilde{x} = y + \Delta y \tag{47}$$

such that $\varphi(\tilde{x}) = 0$. While there are infinite ways to perform the restoration, the most logical is that developed in Ref. 5: the constraint is restored to a prescribed degree of accuracy with the least-square change of the position vector.

If quasilinearization is employed, Eq. (2) is approximated by

$$\varphi(y) + \varphi_x^T(y)\Delta y = 0 \tag{48}$$

In order to prevent the variation Δy from becoming too large, we imbed Eq. (48) into the one-parameter family of equations

$$k\varphi(y) + \varphi_x^T(y)\Delta y = 0 \tag{49}$$

where

$$0 \leq k \leq 1 \tag{50}$$

denotes a scaling factor.

In the light of previous discussion, we seek the minimum of the function

$$J = (1/2)\Delta y^T \Delta y \tag{51}$$

subject to the linearized constraint (49). Standard methods of the theory of maxima and minima show that the fundamental function of this problem is given by

$$\omega = (1/2)\Delta y^T \Delta y + \sigma^T [k\varphi(y) + \varphi_x^T(y)\Delta y] \tag{52}$$

where σ, a q-vector, denotes an undetermined, constant Lagrange multiplier. The optimum change Δy satisfies the relation

$$\omega_{\Delta y} = 0 \tag{53}$$

where $\omega_{\Delta y}$ denotes the gradient of the fundamental function ω with respect to the vector Δy. The explicit form of (53) is the following:

$$\Delta y = -\varphi_x(y)\sigma \tag{54}$$

The Lagrange multiplier σ is obtained by combining (49) and (54) to eliminate Δy. This yields the relation

$$k\varphi(y) - \varphi_x^T(y)\varphi_x(y)\sigma = 0 \tag{55}$$

In conclusion, the restoration algorithm is represented by

$$\sigma = k[\varphi_x^T(y)\varphi_x(y)]^{-1}\varphi(y)$$

$$\Delta y = -\varphi_x(y)\sigma \tag{56}$$

$$\tilde{x} = y + \Delta y$$

and must be employed as follows. For any given k in the range (50), (56-1) supplies the Lagrange multiplier vector σ. Once σ is known, the correction Δy is given by (56-2) and the corrected position vector $\tilde{x}$ is given by (56-3). Of course, the restoration phase must be performed iteratively until a desired degree of accuracy is obtained, that is, until the inequality

$$P(\tilde{x}) \leq \theta_2 \tag{57}$$

is satisfied, where θ_2 is a small number and

$$P(x) = \varphi^T(x)\varphi(x) \tag{58}$$

denotes the error in the constraint or performance index.

6.1. Descent Property. The first variation of the performance index P(y) is given by

$$\delta P(y) = 2\varphi^T(y)\varphi_x^T(y)\Delta y \tag{59}$$

and, because of Eq. (49), reduces to

$$\delta P(y) = -2k\varphi^T(y)\varphi(y) \tag{60}$$

which, in turn, can be written as

$$\delta P(y) = -2kP(y) \tag{61}$$

Since $P(y) > 0$, Eq. (61) shows that the first variation of the performance index is negative for $k > 0$. Therefore, if k is sufficiently small, the decrease of the performance index is guaranteed. In practice, one can use $k = 1$. If this value of k does not result in a decrease in P, then k is successively bisected until P is decreased.

7. Order of Magnitude Analysis

The sequential conjugate gradient-restoration algorithm is represented by Eqs. (39) and (56). While Eqs. (56) are exact, Eqs. (39) are an approximation to the true optimum conditions. In this section, an estimate of the error involved in the computation of the Lagrange multiplier λ is given; furthermore, a justification of the descent properties of the algorithm is presented.

7.1. Lagrange Multiplier. Here, we assume that

$$\alpha = O(\epsilon) \tag{62}$$

for every gradient phase, where ϵ is a small quantity. Concerning the multiplier λ, we note that the exact equation (19) has been replaced by the approximate equation (39-1); therefore, the q-vector

$$R_1 = \gamma\varphi_x^T(x)\hat{p} \tag{63}$$

is representative of the order of magnitude of the error in λ. This is due to the fact that the terms $\varphi_x^T(x)f_x(x)$ and $\varphi_x^T(x)\varphi_x(x)$ in Eq. (19) can be regarded to be of $O(1)$. Laborious manipulations (see Ref. 6), omitted for the sake of brevity, show that

$$R_1 = O(\epsilon) \tag{64}$$

Therefore, the Lagrange multiplier λ computed with (39-1) is precise to $O(\epsilon)$.

7.2. Remark. To first-order, the constraint (2) at the point y is given by

$$\varphi(y) = \delta\varphi(x) = -\alpha R_1 \tag{65}$$

Therefore, because of (39-4), (56-1), (56-2), we conclude that

$$\Delta x = O(\epsilon) \quad , \quad \Delta y = O(\epsilon^2) \tag{66}$$

7.3. Descent Property of the Gradient Phase. The first-order change of the function $F(x, \lambda)$ between points x and y is given by[2]

$$\delta F(x, \lambda) = -\alpha F_x^T(x, \lambda) F_x(x, \lambda) - R_2 \tag{67}$$

where R_2 is the scalar

$$R_2 = \alpha\gamma F_x^T(x, \lambda)\hat{p} \tag{68}$$

Laborious manipulations (see Ref. 6), omitted for the sake of brevity, show that

$$R_2 = O(\epsilon^3) \tag{69}$$

Since the first term on the right-hand side of Eq. (67) is $O(\epsilon)$, R_2 can be neglected and Eq. (67) can be approximated by

$$\delta F(x, \lambda) = -\alpha F_x^T(x, \lambda) F_x(x, \lambda) \tag{70}$$

This relationship shows that, if $\alpha > 0$, $\delta F(x,\lambda) < 0$. This is the descent property of the fundamental function during the gradient phase.

---

[2] In computing $\delta F(x, \lambda)$, the multiplier λ is held constant.

Because of definition (8), the relationship

$$\delta f(x) = \delta F(x, \lambda) - \lambda^T \delta\varphi(x) \tag{71}$$

can be established. Since $\delta F(x, \lambda)$ is $O(\epsilon)$ and $\delta\varphi(x)$ is $O(\epsilon^2)$, Eq. (71) can be approximated by

$$\delta f(x) = \delta F(x, \lambda) = -\alpha F_x^T(x, \lambda) F_x(x, \lambda) \tag{72}$$

This is the descent property of the function f(x) during the gradient phase.

7.4. Descent Property of the Algorithm. Finally, we consider points x and $\tilde{x}$, both satisfying the constraint (2). To first order, the difference of the values of the function f(x) at these points is given by

$$f(\tilde{x}) - f(x) = f_x^T(x)(\Delta x + \Delta y) \tag{73}$$

Because of (66), the second term on the right-hand side of (73) can be neglected with respect to the first. Hence, Eq. (73) becomes

$$f(\tilde{x}) - f(x) = \delta f(x) = \delta F(x, \lambda) = -\alpha F_x^T(x, \lambda) F_x(x, \lambda) \tag{74}$$

Therefore, for ϵ sufficiently small, the restoration algorithm preserves the descent property of the gradient algorithm: the function f decreases between any two successive restoration phases.

7.5. Coefficient γ. In the present algorithm, expression (39-2) has been used for the coefficient γ. The main justification for (39-2) is that it produces quadratic convergence in the terminal stage of the algorithm. The additional justification is that the terms (63) and (68) containing γ are of $O(\epsilon)$ and $O(\epsilon^3)$, respectively. As a consequence, they do not seriously affect the computation of the multiplier λ and the descent property of the algorithm.

8. Sequential Conjugate Gradient-Restoration Algorithm

The algorithm presented in this thesis consists of the alternate succession of gradient phases and restoration phases. A summary of the algorithm and its properties is given below.

8.1. Gradient Phase. (a) Select a nominal point x such that $\varphi(x) = 0$; (b) at this point, determine the vector $f_x(x)$ and the matrix $\varphi_x(x)$; (c) compute the multiplier λ with (39-1), the coefficient γ with (39-2), and the search direction p with (39-3), where $\hat{p}$ is known from the previous iteration; (d) determine the optimum stepsize α by a one-dimensional search in which $F(y, \lambda) = Z(\alpha)$ is minimized along the search direction p; the search is terminated when Ineq. (45) is satisfied; (e) compute the displacement Δx with (39-4) and the varied point y with (39-5).

8.2. Restoration Phase. (a) At point y, compute the vector $\varphi(y)$ and the matrix $\varphi_x(y)$; (b) assuming $k = 1$, determine the multiplier σ with (56-1), the displacement Δy with (56-2), and the varied point $\tilde{x}$ with (56-3); (c) if $P(\tilde{x}) < P(y)$, the scaling factor $k = 1$ is acceptable; if $P(\tilde{x}) > P(y)$, the previous value of k must be replaced by some smaller value in the range (50), until the condition $P(\tilde{x}) < P(y)$ is met; this can be achieved through successive bisections of k; (d) return to step (a) and repeat the restoration algorithm using $\tilde{x}$ as the starting point y for the subsequent iteration; (e) terminate the restoration algorithm when the stopping condition (57) is satisfied; (f) once the restoration algorithm is completed, verify the inequality

$$f(\tilde{x}) < f(x) \qquad (75)$$

If Ineq. (75) is satisfied, start the next gradient phase. If Ineq. (75) is violated, return to the previous gradient phase and reduce the stepsize α until, after restoration, Ineq. (75) is satisfied.

8.3. Starting Condition. At the start of the algorithm, no information pertaining to the previous iteration is available; hence, we set $\hat{p} = 0$ or $\gamma = 0$. This means that the first step is a pure gradient step.

8.4. Restarting Condition. The algorithm must be restarted (a) when the optimum stepsize α of the gradient phase cannot be employed due to violation of Ineq. (75) and (b) at the end of every $\Delta N = n-q$ or $\Delta N = n-q+1$ iterations. The restarting is performed by setting $\hat{p} = 0$ or $\gamma = 0$.

8.5. Stopping Condition. The algorithm is terminated when

$$Q(\tilde{x}, \tilde{\lambda}) \leq \theta_3 \tag{76}$$

where θ_3 is a small number and

$$Q(x, \lambda) = F_x^T(x, \lambda) F_x(x, \lambda) \tag{77}$$

denotes the error in the optimum conditions.

9. Examples: General Information

In order to illustrate the theory, several numerical examples are developed using a Burroughs B-5500 computer and double-precision arithmetic. Concerning the gradient phase, the one-dimensional search is performed so as to minimize $F(y, \lambda) = Z(\alpha)$ with respect to α along the search direction p. Quasilinearization is used; the following stopping condition is employed:

$$Z_\alpha^2(\alpha)/Z_\alpha^2(0) \leq 10^{-6} \tag{78}$$

corresponding to $\epsilon_1 = 10^{-6}$ in (46). Concerning the restoration phase, the restoration algorithm is employed iteratively until the error in the constraint satisfies the inequality

$$P(\tilde{x}) \leq 10^{-20} \tag{79}$$

corresponding to $\theta_2 = 10^{-20}$ in (57). Finally, the sequential conjugate gradient-restoration algorithm is terminated when the error in the optimum conditions satisfies the inequality

$$Q(\tilde{x}, \tilde{\lambda}) \leq 10^{-12} \tag{80}$$

corresponding to $\theta_3 = 10^{-12}$ in (76).

In the following sections, two numerical examples are presented: (a) an example pertaining to a quadratic function and a linear constraint and (b) an example pertaining to a nonquadratic function and a nonlinear constraint. The following terminology is adopted: N is the iteration number (each iteration includes a gradient phase and a restoration phase), N_R is the number of restoration cycles per iteration, N_* is the number of iterations for convergence, and ΔN is the number of iterations between successive restarting points[3]. For simplicity, the symbols employed in Sections 10 and 11 are scalar.

---

[3] For $\Delta N = 1$, one obtains the ordinary gradient-restoration algorithm (Ref. 7).

10. Example: Quadratic Function, Linear Constraint

The first example deals with a function of the form (26-1) subject to a constraint of the form (26-2). For this function, the following properties hold: (a) no restoration is needed, that is, $N_R = 0$; (b) the search performed by quasi-linearization yields the optimal value of α in one step; (c) convergence to the minimum occurs in no more than n-q iterations.

Consider the function

$$f = (x + y)^2 + (y + z)^2 \tag{81}$$

subject to the constraint

$$x + 2y + 3z - 1 = 0 \tag{82}$$

Note that $n = 3$, $q = 1$, so that $n - q = 2$. This function admits the relative minimum $f = 0$ at the point defined by

$$x = 1/2 \quad , \quad y = -1/2 \quad , \quad z = 1/2 \tag{83}$$

The nominal point chosen for starting the algorithm is the point of coordinates

$$x = -4 \quad , \quad y = 1 \quad , \quad z = 1 \tag{84}$$

consistent with (82).

Convergence is achieved in $N_* = 19$ iterations for $\Delta N = 1$ and in $N_* = 2$ iterations for $\Delta N = 2$. The detailed results pertaining to $\Delta N = 2$ are shown in Table 1.

Table 1

N	N_R	x	y	z	f	Q
0	0	-4.0000	1.0000	1.0000	0.13×10^{2}	0.55×10^{2}
1	0	-1.6459	1.8759	-0.3686	0.23×10^{1}	0.20×10^{1}
2	0	0.5000	-0.5000	0.5000	0.93×10^{-42}	0.50×10^{-41}

11. Example: Nonquadratic Function, Nonlinear Constraint

The second example deals with a nonquadratic function subject to a nonlinear constraint. The algorithm is restarted every ΔN iterations.

Consider the function

$$f = (x - y)^2 + (y - z)^4 \tag{85}$$

subject to the constraint

$$x(1 + y^2) + z^4 - 3 = 0 \tag{86}$$

Note that $n = 3$, $q = 1$, so that $n-q = 2$. This function admits the relative minimum $f = 0$ at the point defined by

$$x = 1 \quad , \quad y = 1 \quad , \quad z = 1 \tag{87}$$

The nominal point chosen for starting the algorithm is the point of coordinates

$$x = -13/5 \quad , \quad y = 2 \quad , \quad z = 2 \tag{88}$$

consistent with (86).

Convergence is achieved in $N_* = 4569$ iterations for $\Delta N = 1$, $N_* = 13$ iterations for $\Delta N = 2$, $N_* = 13$ iterations for $\Delta N = 3$, and $N_* = 10$ iterations for $\Delta N = 4$. The detailed results pertaining to $\Delta N = 2$ are shown in Table 2.

Table 2

N	N_R	x	y	z	f	Q
0	0	-2.6000	2.0000	2.0000	0.21×10^{2}	0.15×10^{3}
1	5	-0.3769	0.0132	1.3556	0.33×10^{1}	0.82×10^{2}
2	5	1.5559	0.9512	0.4357	0.43×10^{0}	0.21×10^{1}
3	3	1.2026	1.1752	0.6076	0.10×10^{0}	0.86×10^{0}
4	6	0.7711	0.9447	1.1140	0.30×10^{-1}	0.22×10^{0}
5	2	0.8535	0.8659	1.1078	0.35×10^{-2}	0.34×10^{-2}
6	4	1.0766	1.0568	0.9214	0.72×10^{-3}	0.25×10^{-2}
7	2	1.0658	1.0641	0.9234	0.39×10^{-3}	0.19×10^{-3}
8	3	1.0211	1.0225	0.9769	0.62×10^{-5}	0.17×10^{-4}
9	1	1.0218	1.0217	0.9770	0.39×10^{-5}	0.17×10^{-6}
10	2	1.0055	1.0056	0.9943	0.26×10^{-7}	0.85×10^{-7}
11	1	1.0055	1.0055	0.9943	0.15×10^{-7}	0.43×10^{-10}
12	2	1.0008	1.0008	0.9991	0.12×10^{-10}	0.24×10^{-10}
13	0	1.0008	1.0008	0.9991	0.97×10^{-11}	0.65×10^{-15}

12. Discussion and Conclusions

In the previous sections, a sequential conjugate gradient-restoration algorithm is developed for minimizing a function f(x) subject to the constraint $\varphi(x) = 0$. The algorithm is composed of the alternate succession of conjugate gradient phases and restoration phases. For a quadratic function subject to a linear constraint, the minimum point is obtained in no more than n-q iterations.

In order to illustrate the theory, two numerical examples are presented. The first example is concerned with a quadratic function subject to a linear constraint: the computer results confirm the quadratic convergence property of the algorithm. The next example is concerned with a nonquadratic function subject to a nonlinear constraint: the computer results show the rapid convergence of the algorithm and its superiority with respect to the sequential ordinary gradient-restoration algorithm.

In the theory as well as in the examples, the function $F(y, \lambda) = Z(\alpha)$ is minimized along the search direction. However, the occurrence of cases where $Z(\alpha)$ does not possess a relative minimum is conceivable. For these cases, an upper limit must be imposed on the stepsize α or the performance index P(y).

In the numerical examples, the starting point was chosen so that $\varphi(x) = 0$. However, the present algorithms can be started even if $\varphi(x) \neq 0$. In this case, the first phase is a restoration phase rather than a gradient phase.

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. BECKMAN, F.S., The Solution of Linear Equations by the Conjugate Gradient Method, Mathematical Methods for Digital Computers, Edited by A. Ralston and H.S. Wilf, John Wiley and Sons, New York, 1960.

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

4. MIELE, A., HUANG, H.Y., and CANTRELL, J.W., Gradient Methods in Mathematical Programming, Part 1, Review of Previous Techniques, Rice University, Aero-Astronautics Report No. 55, 1969.

5. MIELE, A., HEIDEMAN, J.C., and DAMOULAKIS, J.N., The Restoration of Constraints in Holonomic and Nonholonomic Problems, Journal of Optimization Theory and Applications, Vol. 3, No. 5, 1969.

6. MIELE, A., HUANG, H.Y., and HEIDEMAN, J.C., Mathematical Programming for Constrained Minimal Problems, Part 2, Sequential Conjugate Gradient-Restoration Algorithm, Rice University, Aero-Astronautics Report No. 61, 1969.

7. MIELE, A., and HEIDEMAN, J.C., Mathematical Programming for Constrained Minimal Problems, Part 1, Sequential Gradient-Restoration Algorithm, Rice University, Aero-Astronautics Report No. 59, 1969.

Additional Bibliography

ROSEN, J.B., The Gradient Projection Method for Nonlinear Programming, Part 2, Nonlinear Constraints, SIAM Journal on Applied Mathematics, Vol. 9, No. 4, 1961.

BRYSON, A.E., Jr., and HO, Y.C., Applied Optimal Control, Chapter 1, Blaisdell Publishing Company, New York, 1969.

Fig. 1

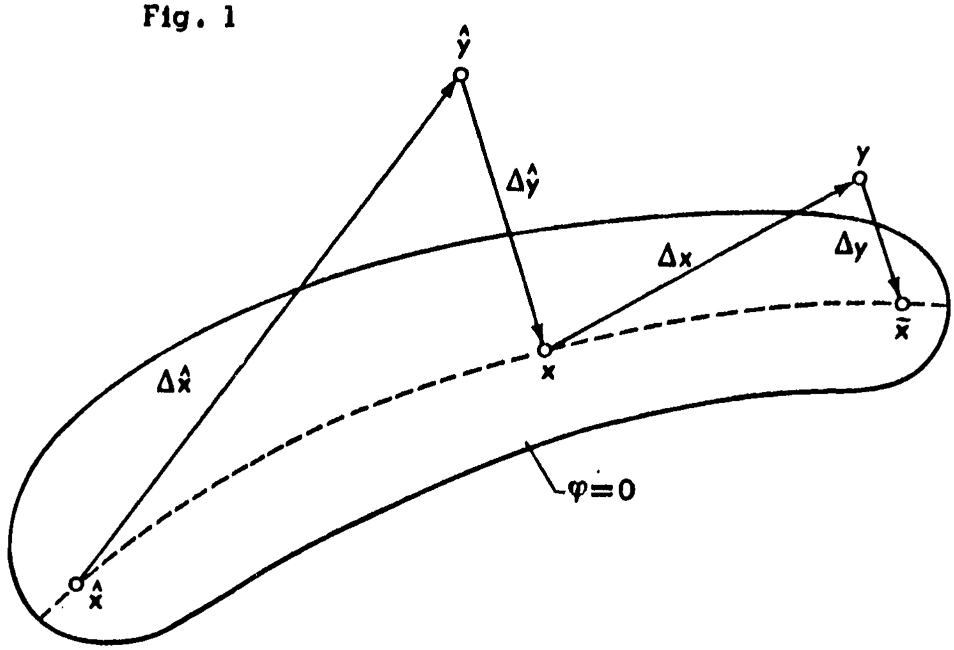

Fig. 2