71-26,277

CRAGG, Edward E, 1946-
ON A CONJUGATE GRADIENT-RESTORATION ALGORITHM FOR MATHEMATICAL PROGRAMMING PROBLEMS.

Rice University, Ph.D., 1971
Mathematics

RICE UNIVERSITY

ON A CONJUGATE GRADIENT-RESTORATION ALGORITHM FOR MATHEMATICAL PROGRAMMING PROBLEMS

by

EDWARD E CRAGG

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

Thesis Director's signature:

Angelo Miele

Houston, Texas
May 1971

Abstract

On a Conjugate Gradient-Restoration Algorithm

for Mathematical Programming Problems

by

EDWARD E CRAGG

In this thesis, 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 is a q-vector, with $q < n$. The use of the augmented penalty function is explored in connection with the conjugate gradient-restoration algorithm. The augmented penalty function $W(x, \lambda, k)$ is defined to be the linear combination of the augmented function $F(x, \lambda)$ and the constraint error $P(x)$, where the q-vector λ is the Lagrange multiplier and the scalar k is the penalty constant.

The conjugate gradient-restoration algorithm includes a conjugate-gradient phase involving n-q iterations and a restoration phase involving one iteration. In the conjugate-gradient phase, one tries to improve the value of the function, while avoiding excessive constraint violation. In the restoration phase, one reduces the constraint error, while avoiding excessive change in the value of the function.

Concerning the conjugate-gradient phase, two classes of algorithms are considered: for algorithms of Class I, the Lagrange multiplier λ is

determined so that the error in the optimum condition is minimized for given x; for algorithms of Class II, the Lagrange multiplier λ is determined so that the constraint is satisfied to first order. For each class, two versions are studied. In version (α), the penalty constant is held unchanged throughout the entire algorithm. In version (β), the penalty constant is updated at the beginning of each conjugate-gradient phase so as to achieve certain desirable properties.

Concerning the restoration phase, the minimum distance algorithm is employed. Since the use of the augmented penalty function automatically prevents excessive constraint violation, single-step restoration is considered.

If the function f(x) is quadratic and the constraint $\varphi(x)$ is linear, all the previous algorithms are identical, that is, they produce the same sequence of points and converge to the solution in the same number of iterations. This number of iterations is at most $N_* = n-q$ if the starting point x_s is such that $\varphi(x_s) = 0$ and at most $N_* = 1+n-q$ if the starting point x_s is such that $\varphi(x_s) \neq 0$.

In order to illustrate the theory, five numerical examples are developed. The first example refers to a quadratic function and a linear constraint. The remaining examples refer to nonquadratic functions and nonlinear constraints. For the linear-quadratic example, all the algorithms behave identically, as predicted by the theory. For the nonlinear-nonquadratic examples, algorithms of Class II generally exhibit faster convergence than algorithms of Class I and algorithms of type (β) generally exhibit faster convergence than algorithms of type (α).

ACKNOWLEDGEMENT

The author is indebted to Dr. Angelo Miele for helpful discussions. He would also like to thank Mr. Alejandro Velasco Levy for checking the numerical accuracy of the figures included. This research was supported by the National Science Foundation Grant No. GP-27271.

TABLE OF CONTENTS

1. Introduction

In Ref. 1, the problem of minimizing a function f(x) subject to a constraint $\varphi(x) = 0$ was considered, where f is a scalar, x an n-vector, and φ a q-vector, with $q < n$. The use of the augmented penalty function (Ref. 2) was explored in connection with the ordinary gradient algorithm. The augmented penalty function $W(x, \lambda, k)$ combines linearly the augmented function and the constraint error and is defined as

$$W(x, \lambda, k) = f(x) + \lambda^T \varphi(x) + k\varphi^T(x)\varphi(x) \qquad (1)$$

where the q-vector λ is the Lagrange multiplier and the scalar $k > 0$ is the penalty constant. The superscript T denotes the transpose of a matrix.

The ordinary gradient algorithm of Ref. 1 is constructed in such a way that the following properties are satisfied in toto or in part: (a) descent property on the augmented penalty function, (b) descent property on the augmented function, (c) descent property on the constraint error, and either (d) constraint satisfaction on the average or (e) individual constraint satisfaction.

With the above considerations in mind, two classes of algorithms were developed in Ref. 1. For algorithms of Class I, the multiplier is determined so that the error in the optimum condition is minimized for given x; for algorithms of Class II, the multiplier is determined so that the constraint is satisfied to first order.

For each class, two versions were presented. In version (α), the penalty constant is held unchanged for all iterations. In version (β), the penalty constant is updated at each iteration so as to ensure satisfaction of property (d) for algorithms of Class I and property (b) for algorithms of Class II.

In this thesis, the use of the augmented penalty function (1) is explored in connection with the conjugate gradient-restoration algorithm (Refs. 3-4). Specifically, algorithms whose basic cycle involves a conjugate-gradient phase and a restoration phase are considered. In the conjugate-gradient phase, one tries to improve the value of the function while avoiding excessive constraint violation. In the restoration phase, one reduces the constraint error, while avoiding excessive change in the value of the function.

Concerning the conjugate-gradient phase, two classes of algorithms are considered: for algorithms of Class I, the multiplier λ is determined so that the error in the optimum condition is minimized for given x; for algorithms of Class II, the multiplier λ is determined so that the constraint is satisfied to first order. As in Ref. 1, two versions are given for each class. In version (α), the penalty constant is held unchanged throughout the entire algorithm. In version (β), the penalty constant is updated at the beginning of each conjugate-gradient phase so as to achieve certain desirable properties.

Concerning the restoration phase, the minimum distance algorithm of Refs. 3-4 is employed. From Ref. 4, we know that incomplete restoration is to be preferred to complete restoration and infrequent restoration is to be preferred to frequent restoration if fast convergence is desired. Since the use of the augmented penalty function (1) automatically prevents excessive constraint violation, the algorithm with incomplete and infrequent restoration is investigated here. Therefore, the restoration phase involves only one iteration and precedes every n-q conjugate-gradient iterations.

2. Statement of the Problem

We consider the problem of minimizing the function

$$f = f(x) \tag{2}$$

subject to the constraint

$$\varphi(x) = 0 \tag{3}$$

In the above equations, 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 functions f and φ with respect to x exist and are continuous; it is also assumed that the constrained minimum exists.

2.1. Exact First-Order Conditions. From theory of maxima and minima, it is known that the previous problem can be recast as that of minimizing the augmented function

$$F(x, \lambda) = f(x) + \lambda^T \varphi(x) \tag{4}$$

subject to the constraint (3). Here, λ is a q-vector Lagrange multiplier, and the superscript T denotes the transpose of a matrix. If

$$F_x(x, \lambda) = f_x(x) + \varphi_x(x)\lambda \tag{5}$$

denotes the gradient of the augmented function[2], the optimum solution x, λ must satisfy the simultaneous equations

$$\varphi(x) = 0 \quad , \quad F_x(x, \lambda) = 0 \tag{6}$$

---

[1] All vectors are column vectors.

[2] In Eq. (5), the gradients f_x and F_x denote n-vectors and the matrix φ_x is $n \times q$.

2.2. Approximate Solutions. In general, the system (6) is nonlinear; consequently, approximate methods must be employed. These are of two kinds: first-order methods (such as the one discussed in subsequent sections of this thesis) and second-order methods. Here, we introduce the scalar quantities

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

measuring the error in the constraint and the optimum condition, respectively. We observe that $P = 0$ and $Q = 0$ for the optimum solution, while $P > 0$ and/or $Q > 0$ for any approximation to the solution. When approximate methods are used, they must ultimately lead to values of x, λ such that

$$P(x) \leq \varepsilon_1 \quad , \quad Q(x,\lambda) \leq \varepsilon_2 \tag{8}$$

Alternatively, (8) can be replaced by

$$R(x,\lambda) \leq \varepsilon_3 \tag{9}$$

where

$$R(x,\lambda) = P(x) + Q(x,\lambda) \tag{10}$$

denotes the cumulative error in the constraint and the optimum condition. Here, ε_1, ε_2, ε_3 are small, preselected numbers. Note that satisfaction of Ineq. (9) implies satisfaction of Ineqs. (8), if one chooses $\varepsilon_1 = \varepsilon_2 = \varepsilon_3$.

3. Conjugate-Gradient Phase

In this section, we construct a conjugate-gradient algorithm based on the consideration of the augmented penalty function

$$W(x, \lambda, k) = F(x, \lambda) + kP(x) \tag{11}$$

where

$$F(x, \lambda) = f(x) + \lambda^T \varphi(x) \quad , \quad P(x) = \varphi^T(x)\varphi(x) \tag{12}$$

In this algorithm, the Lagrange multiplier and the penalty constant are determined so as to ensure satisfaction of the following properties, in toto or in part: (a) descent property on the augmented penalty function, (b) descent property on the augmented function, (c) descent property on the constraint error, and either (d) constraint satisfaction on the average or (e) individual constraint satisfaction. Properties (d) and (e) are employed to first order only.

3.1. Basic Algorithm. Let x denote the nominal point, $\tilde{x}$ the varied point, and Δx the displacement leading from the nominal point to the varied point. Let λ denote the Lagrange multiplier, k the penalty constant, p the present search direction, $\hat{p}$ the previous search direction, γ the directional coefficient, and α the gradient stepsize. Both p and $\hat{p}$ are n-vectors, while γ and α are scalars. With these definitions in mind, we consider the conjugate-gradient algorithm represented by

$$\begin{aligned} F_x(x, \lambda) &= f_x(x) + \varphi_x(x)\lambda \\ P_x(x) &= 2\varphi_x(x)\varphi(x) \\ W_x(x, \lambda, k) &= F_x(x, \lambda) + kP_x(x) \end{aligned} \tag{13}$$

$$
\begin{aligned}
p &= W_x(x, \lambda, k) + \gamma\hat{p} \\
\Delta x &= -\alpha p \qquad\qquad (13) \\
\tilde{x} &= x + \Delta x
\end{aligned}
$$

whose form is suggested by the results of Refs. 3-4. For given nominal point x, Lagrange multiplier λ, directional coefficient γ, and penalty constant k, Eqs. (13) constitute a complete iteration leading to the varied point $\tilde{x}$, providing one specifies the gradient stepsize α.

3.2. Lagrange multiplier. In accordance with the discussion of Section 1, two possible determinations of the multiplier are presented here.

Algorithms of Class I. In these algorithms, the Lagrange multiplier is determined so that the error in the optimum condition (7-2) is minimized with respect to λ for given x. Owing to the fact that

$$Q(x, \lambda) = [f_x(x) + \varphi_x(x)\lambda]^T [f_x(x) + \varphi_x(x)\lambda] \qquad (14)$$

the multiplier is determined by the relation

$$Q_\lambda(x, \lambda) = 0 \qquad (15)$$

which implies that

$$\varphi_x^T(x)\varphi_x(x)\lambda + \varphi_x^T(x)f_x(x) = 0 \qquad (16)$$

This linear vector equation is equivalent to q linear scalar relations in which the only unknown is the Lagrange multiplier λ. The unique multiplier solving

Eq. (16) is denoted by

$$\lambda = \lambda_o \tag{17}$$

Algorithms of Class II. In these algorithms, the Lagrange multiplier is determined so that, at the end of any iteration, the constraint is satisfied to first order. Let $\varphi(x) \neq 0$ and $\varphi(\tilde{x}) = 0$ to first order. If quasilinearization is employed, we obtain the relation

$$\varphi(x) + \varphi_x^T(x)\Delta x = 0 \tag{18}$$

which, for convenience, is imbedded in the more general relation

$$\mu\varphi(x) + \varphi_x^T(x)\Delta x = 0 \tag{19}$$

where $\mu \geq 0$ denotes a scaling factor to be specified. If Eqs. (13-5) and (19) are combined, we see that

$$\mu\varphi(x) - \alpha\varphi_x^T(x)p = 0 \tag{20}$$

Let μ be proportional to α throughout the algorithm, that is, let

$$\mu = C\alpha \tag{21}$$

where C is a constant to be specified[3]. Then, Eq. (20) becomes

$$C\varphi(x) - \varphi_x^T(x)p = 0 \tag{22}$$

which, in the light of Eqs. (13-1) through (13-4), becomes

$$\varphi_x^T(x)\varphi_x(x)\lambda + \varphi_x^T(x)[f_x(x) + kP_x(x) + \gamma\hat{p}] - C\varphi(x) = 0 \tag{23}$$

[3] If the function f(x), the constraint $\varphi(x)$, and the vector x are scaled in such a way that the gradient stepsize α is O(1), then the choice C = 1 is appropriate.

For given directional coefficient γ, penalty constant k, and constant C, this linear vector equation is equivalent to q linear scalar equations, in which the only unknown is the Lagrange multiplier λ. The unique multiplier solving Eq. (23) is denoted by

$$\lambda = \lambda_* \tag{24}$$

3.3. Directional Coefficient. For both algorithms of Class I and Class II, the directional coefficient γ is determined by the relation

$$\gamma = 0 \tag{25}$$

or

$$\gamma = W_x^T(x, \lambda_o, k) W_x(x, \lambda_o, k) / W_x^T(\hat{x}, \hat{\lambda}_o, k) W_x(\hat{x}, \hat{\lambda}_o, k) \tag{26}$$

Equation (25) is to be employed for the first iteration of the conjugate-gradient phase and means that the search direction p is identical with the gradient of the augmented penalty function $W_x(x, \lambda, k)$. Equation (26) is to be employed for the remaining iterations of the conjugate-gradient phase; since $\gamma \neq 0$, the search direction p is not identical with the gradient of the augmented penalty function. In Eq. (26), x denotes the present point, $\hat{x}$ the previous point, λ_o the solution of Eq. (16) at the present point, and $\hat{\lambda}_o$ the solution of Eq. (16) at the previous point.

3.4. Penalty Constant. In accordance with the discussion of Section 1, two possible determinations of the penalty constant are presented.

Version (α). The penalty constant is held at a preselected value throughout the algorithm. It can be anticipated that, if k is either very small or very large, a large number of iterations may be required for convergence; hence, k must be in a proper range.

Version (β). To circumvent the difficulties of version (α), the penalty constant is held at apreselected value only throughout each conjugate-gradient phase, and not throughout the entire algorithm. At the beginning of each conjugate-gradient phase, k is updated in such a way that certain desirable properties hold. Let x_o denote the position vector at the beginning of a conjugate-gradient phase and let the penalty constant k be selected from

$$k = 2CP(x_o)/P_x^T(x_o)P_x(x_o) \tag{27}$$

Then, for the first iteration of the conjugate-gradient phase, property (d) holds for algorithms of Class I and property (b) holds for algorithms of Class II (see Ref. 1). We recall that property (d) means constraint satisfaction on the average and property (c) means descent property on the augmented function $F(x, \lambda)$.

3.5. Descent Properties. In the previous sections, we discussed the determination of the Lagrange multiplier λ, the directional coefficient γ, and the penalty constant k for both algorithms of Class I and Class II. Prior to determining the gradient stepsize α for given values of λ, γ, k, we establish whether certain descent properties are satisfied. When the displacement (13-5) is employed, the first variations of the functions $W(x, \lambda, k)$, $F(x, \lambda)$, $P(x)$ are given by[4]

$$\begin{aligned} \delta W(x, \lambda, k) &= W_x^T(x, \lambda, k)\Delta x = -\alpha W_x^T(x, \lambda, k)p \\ \delta F(x, \lambda) &= F_x^T(x, \lambda)\Delta x = -\alpha F_x^T(x, \lambda)p \\ \delta P(x) &= P_x^T(x)\Delta x = -\alpha P_x^T(x)p \end{aligned} \tag{28}$$

[4] In the computation of the first variations of the functions $W(x, \lambda, k)$ and $F(x, \lambda)$, the multiplier λ is held constant.

and, in the light of (13-1) through (13-4), can be rewritten as

$$\delta W(x, \lambda, k) = -\alpha W_x^T(x, \lambda, k)[W_x(x, \lambda, k) + \gamma\hat{p}]$$
$$\delta F(x, \lambda) = -\alpha F_x^T(x, \lambda)[F_x(x, \lambda) + kP_x(x) + \gamma\hat{p}] \tag{29}$$
$$\delta P(x) = -\alpha P_x^T(x)[F_x(x, \lambda) + kP_x(x) + \gamma\hat{p}]$$

Augmented Penalty Function. For both algorithms of Class I and algorithms of Class II, the first variation of the augmented penalty function is negative providing

$$W_x^T(x, \lambda, k)[W_x(x, \lambda, k) + \gamma\hat{p}] > 0 \tag{30}$$

For the first iteration of the conjugate-gradient phase ($\gamma = 0$), Ineq. (30) is satisfied, and the descent property $\delta W(x, \lambda, k) < 0$ holds. Therefore, for α sufficiently small, the decrease of the augmented penalty function is guaranteed. For subsequent iterations ($\gamma \neq 0$), Ineq. (30) may or may not be satisfied, and the descent property $\delta W(x, \lambda, k) < 0$ may or may not hold. Whenever Ineq. (30) is violated, the conjugate-gradient phase must be interrupted, and the restoration phase must be started.

Augmented Function. For both algorithms of Class I and algorithms of Class II, the first variation of the augmented function is negative providing

$$F_x^T(x, \lambda)[F_x(x, \lambda) + kP_x(x) + \gamma\hat{p}] > 0 \tag{31}$$

For algorithms of Class I, Eq. (16) implies that

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

and, Ineq. (31) becomes

$$F_x^T(x, \lambda)[F_x(x, \lambda) + \gamma\hat{p}] > 0 \tag{33}$$

For the first iteration of the conjugate-gradient phase ($\gamma = 0$), Ineq. (33) is satisfied, and the descent property $\delta F(x, \lambda) < 0$ holds. Therefore, for α sufficiently small, the decrease of the augmented function is guaranteed. For subsequent iterations, Ineq. (33) may or may not be satisfied, and the descent property $\delta F(x, \lambda) < 0$ may or may not hold.

For algorithms of Class II, Eq. (23) implies that

$$P_x^T(x)[F_x(x, \lambda) + kP_x(x) + \gamma\hat{p}] - 2CP(x) = 0 \tag{34}$$

and Ineq. (31) becomes

$$F_x^T(x, \lambda)F_x(x, \lambda) + k[2CP(x) - kP_x^T(x)P_x(x)] + \gamma[F_x(x, \lambda) - kP_x(x)]^T\hat{p} > 0 \tag{35}$$

For the first iteration of the conjugate-gradient phase ($\gamma = 0$), Ineq. (35) is satisfied if k is chosen in accordance with (27), and the descent property $\delta F(x, \lambda) < 0$ holds. Therefore, for α sufficiently small, the decrease of the augmented function is guaranteed. For subsequent iterations, Ineq. (35) may or may not be satisfied, and the descent property $\delta F(x, \lambda) < 0$ may or may not hold.

Constraint Error. For both algorithms of Class I and algorithms of Class II, the first variation of the constraint error is negative providing

$$P_x^T(x)[F_x(x, \lambda) + kP_x(x) + \gamma\hat{p}] > 0 \tag{36}$$

For algorithms of Class I, Eq. (32) holds, and Ineq. (36) becomes

$$P_x^T(x)[kP_x(x) + \gamma\hat{p}] > 0 \qquad (37)$$

For the first iteration of the conjugate-gradient phase ($\gamma = 0$), Ineq. (37) is satisfied and the descent property $\delta P(x) < 0$ holds. Therefore, for α sufficiently small, the decrease of the constraint error is guaranteed. For subsequent iterations, Ineq. (37) may or may not be satisfied, and the descent property $\delta P(x) < 0$ may or may not hold. For the above considerations, the restoration phase is indispensable to the stability of algorithms of Class I.

For algorithms of Class II, Eq. (34) applies, and Ineq. (36) becomes

$$2CP(x) > 0 \qquad (38)$$

Since $P(x)$ is positive, the descent property $\delta P(x) < 0$ holds for all iterations of the conjugate-gradient phase. Therefore, for α sufficiently small, the decrease in the constraint error is guaranteed. For the above considerations, the restoration phase is not indispensable to the stability of the algorithms of Class II, but it is desirable in order to ensure quadratic terminal convergence.

3.6. Gradient Stepsize. The descent properties established in the previous section are instrumental in the determination of the optimum gradient stepsize for given nominal point x, Lagrange multiplier λ, directional coefficient γ, and penalty constant k. If Eqs. (13-5) and (13-6) are combined, the position vector at the end of the conjugate-gradient step becomes

$$\tilde{x} = x - \alpha p \qquad (39)$$

where p is known through Eq. (13-4). This is a one-parameter family of varied points $\tilde{x}$, for which the augmented penalty function, the augmented function, and the constraint error are functions of the form

$$W(\tilde{x}, \lambda, k) = W(x-\alpha p, \lambda, k) = W(\alpha)$$
$$F(\tilde{x}, \lambda) = F(x-\alpha p, \lambda) = F(\alpha) \qquad (40)$$
$$P(\tilde{x}) = P(x-\alpha p) = P(\alpha)$$

Along the straight line defined by Eq. (39), the above functions admit the derivatives

$$W_\alpha(\alpha) = -W_x^T(\tilde{x}, \lambda, k)p$$
$$F_\alpha(\alpha) = -F_x^T(\tilde{x}, \lambda)p \qquad (41)$$
$$P_\alpha(\alpha) = -P_x^T(\tilde{x})p$$

which, at $\alpha = 0$, become

$$W_\alpha(0) = -W_x^T(x, \lambda, k)p$$
$$F_\alpha(0) = -F_x^T(x, \lambda)p \qquad (42)$$
$$P_\alpha(0) = -P_x^T(x)p$$

If $W_\alpha(0) < 0$, the descent property on $W(\alpha)$ holds, and the search for the optimum gradient stepsize can be initiated. If $W_\alpha(0) \geq 0$, the descent property on $W(\alpha)$ does not hold; the search direction p must be discarded, the conjugate-gradient phase must be interrupted, and the restoration phase must be started.

We now assume that $W_\alpha(0) < 0$ and that a minimum $W(\alpha)$ exists. Then, we employ some one-dimensional search scheme (for instance, quadratic interpolation, ubic interpolation, or quasilinearization) to determine the value of α for which

$$W_\alpha(\alpha) = 0 \tag{43}$$

This procedure should be used iteratively until the modulus of the slope satisfies any of the following inequalities:

$$W_\alpha^2(\alpha) \le \varepsilon_4 \quad \text{or} \quad W_\alpha^2(\alpha) \le \varepsilon_5 \, W_\alpha^2(0) \tag{44}$$

where ε_4 and ε_5 are small, preselected numbers. Of course, the value of α satisfying Ineq. (44) must be such that

$$W(\alpha) < W(0) \tag{45}$$

3.7. Convergence Properties. If the function $f(x)$ is quadratic, if the constraint $\varphi(x)$ is linear, and if the starting point x_s is such that $\varphi(x_s) = 0$, then algorithms of Class I and algorithms of Class II become identical. They produce the same sequence of points and converge to the solution in at most $N_* = n-q$ iterations. If any of the above conditions is violated, the quadratic convergence property does not hold. However, quadratic terminal convergence can be achieved if a suitable restoration phase is inserted in the algorithm (see Section 4).

4. Restoration Phase

Let x denote the nominal point, $\tilde{x}$ the varied point, and Δx the displacement leading from the nominal point to the varied point. Let σ denote the Lagrange multiplier, p the search direction, and μ the restoration stepsize. Here, σ is a q-vector, p an n-vector, and μ a scalar. With these definitions in mind, we consider the restoration algorithm represented by

$$\begin{aligned} &\varphi_x^T(x)\varphi_x(x)\sigma - \varphi(x) = 0 \\ &p = \varphi_x(x)\sigma \\ &\Delta x = -\mu p \\ &\tilde{x} = x + \Delta x \end{aligned} \tag{46}$$

whose form is suggested by the results of Refs. 3-4. For given nominal point x, Eqs. (46) represent a complete iteration leading to the varied point $\tilde{x}$, providing one specifies the restoration stepsize μ.

4.1. Descent Property. Prior to determining the restoration stepsize μ, we establish a basic descent property. When the displacement (46-3) is employed, the first variation of the function P(x) is given by

$$\delta P(x) = P_x^T(x)\Delta x = -\mu P_x^T(x)p \tag{47}$$

with

$$P(x) = \varphi^T(x)\varphi(x) \quad , \quad P_x(x) = 2\varphi_x(x)\varphi(x) \tag{48}$$

In the light of (46-1), (46-2), (48), Eq. (47) can be rewritten as

$$\delta P(x) = -2\mu P(x) \tag{49}$$

Since μ is positive and P(x) is positive, Eq. (49) shows that $\delta P(x) < 0$. Therefore, for μ sufficiently small, the decrease of the constraint error is guaranteed.

4.2. Restoration Stepsize. The descent property established in the previous section is instrumental in determining the optimum restoration stepsize. If Eqs. (46-3) and (46-4) are combined, the position vector at the end of a restoration step becomes

$$\tilde{x} = x - \mu p \tag{50}$$

where p is known through Eq. (46-2). This is a one-parameter family of varied points $\tilde{x}$, for which the constraint error is a function of the form

$$P(\tilde{x}) = P(x - \mu p) = P(\mu) \tag{51}$$

Along the straight line defined by Eq. (50), the constraint error admits the derivative

$$P_\mu(\mu) = -P_x^T(\tilde{x})p \tag{52}$$

which, at $\mu = 0$, becomes

$$P_\mu(0) = -P_x^T(x)p = -2P(x) \tag{53}$$

a result consistent with (49). Since $P_\mu(0) < 0$, the search for the optimum restoration stepsize can be initiated.

Assuming that a minimum of $P(\mu)$ exists, we employ some one-dimensional search scheme (for instance, quadratic interpolation, cubic interpolation, or quasilinearization) to determine the value of μ for which

$$P_\mu(\mu) = 0 \tag{54}$$

Ideally, this procedure should be used iteratively until the modulus of the slope satisfies any of the following inequalities:

$$P_\mu(\mu) \le \epsilon_6 \quad \text{or} \quad P_\mu(\mu) \le \epsilon_7 \; P_\mu(0) \tag{55}$$

where ϵ_6 and ϵ_7 are small, preselected numbers. Of course, the value of μ satisfying Ineq. (55) must be such that

$$P(\mu) < P(0) \tag{56}$$

Since a rigorous search might take excessive computer time, we propose here an alternate procedure. We observe that, for a linear constraint, Eq. (54) is solved by $\mu = 1$. This result and the descent property of the previous section suggest replacing the rigorous search by a bisection process on μ starting from $\mu = 1$. Specifically, we first assign the value $\mu = 1$ to restoration stepsize and verify Ineq. (56). If Ineq. (56) is satisfied, the iteration is completed. If Ineq. (56) is violated, μ is bisected several times until satisfaction of Ineq. (56) occurs. This is guaranteed by the descent property of the previous section.

Remark. The restoration phase is important for two reasons: (i) it gives stability to algorithms of Class I; for these algorithms, the descent property on the function $P(\mu)$ is not guaranteed during the conjugate-gradient phase; and (ii) it accelerates the convergence of both algorithms of Class I and Class II; if the function $f(x)$ is quadratic, if the constraint $\varphi(x)$ is linear, and if the starting point x_s is such that $\varphi(x_s) \neq 0$, then convergence to the solution in at most $N_* = 1+n-q$ iterations is possible if the restoration phase precedes the conjugate-gradient phase.

5. Summary of Algorithms

The conjugate gradient-restoration algorithms discussed here involve the alternate succession of conjugate-gradient phases and restoration phases. The basic functions involved in these phases are the augmented function $F(x, \lambda)$, the constraint error $P(x)$, and the augmented penalty function $W(x, \lambda, k)$. They are defined by

$$F(x, \lambda) = f(x) + \lambda^T \varphi(x)$$

$$P(x) = \varphi^T(x)\varphi(x) \qquad (57)$$

$$W(x, \lambda, k) = F(x, \lambda) + kP(x)$$

5.1. Conjugate-Gradient Phase. For algorithms of Class I, the conjugate-gradient phase involves n-q iterations, each of which is represented by the following equations:

$$\varphi_x^T(x)\varphi_x(x)\lambda_o + \varphi_x^T(x)f_x(x) = 0$$

$$F_x(x, \lambda_o) = f_x(x) + \varphi_x(x)\lambda_o$$

$$P_x(x) = 2\varphi_x(x)\varphi(x)$$

$$W_x(x, \lambda_o, k) = F_x(x, \lambda_o) + kP_x(x) \qquad (58)$$

$$\gamma = W_x^T(x, \lambda_o, k)W_x(x, \lambda_o, k)/W_x^T(\hat{x}, \hat{\lambda}_o, k)W_x(\hat{x}, \hat{\lambda}_o, k)$$

$$p = W_x(x, \lambda_o, k) + \gamma\hat{p}$$

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

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

For the first iteration, Eq. (58-5) is bypassed and is replaced by $\gamma = 0$.

Algorithm (I-α). This algorithm is represented by Eqs. (58), with the penalty constant k held unchanged throughout the entire algorithm.

Algorithm (I-β). This algorithm is represented by Eqs. (58), with the penalty constant k held unchanged only throughout each conjugate-gradient phase. At the beginning of any conjugate-gradient phase, the penalty constant is updated according to

$$k = 2CP(x_o)/P_x^T(x_o)P_x(x_o) \tag{59}$$

where x_o denotes the position vector at the beginning of the first iteration of any conjugate-gradient phase.

For algorithms of Class II, the conjugate-gradient phase involves n-q iterations, each of which is represented by the following equations:

$$\begin{aligned}
&\varphi_x^T(x)\varphi_x(x)\lambda_o + \varphi_x^T(x)f_x(x) = 0\\
&F_x(x, \lambda_o) = f_x(x) + \varphi_x(x)\lambda_o\\
&P_x(x) = 2\varphi_x(x)\varphi(x)\\
&W_x(x, \lambda_o, k) = F_x(x, \lambda_o) + kP_x(x)\\
&\gamma = W_x^T(x, \lambda_o, k)W_x(x, \lambda_o, k)/W_x^T(\hat{x}, \hat{\lambda}_o, k)W_x(\hat{x}, \hat{\lambda}_o, k)\\
&\varphi_x^T(x)\varphi_x(x)\lambda_* + \varphi_x^T(x)[f_x(x) + kP_x(x) + \gamma\hat{p}] - C\varphi(x) = 0 \qquad (60)\\
&F_x(x, \lambda_*) = f_x(x) + \varphi_x(x)\lambda_*\\
&W_x(x, \lambda_*, k) = F_x(x, \lambda_*) + kP_x(x)\\
&p = W_x(x, \lambda_*, k) + \gamma\hat{p}\\
&\Delta x = -\alpha p\\
&\tilde{x} = x + \Delta x
\end{aligned}$$

For the first iteration, Eq. (60-5) is bypassed and replaced by $\gamma = 0$.

Algorithm (II-α). This algorithm is represented by Eqs. (60), with the penalty constant k held unchanged only throughout each conjugate-gradient phase. At the beginning of any conjugate-gradient phase, the penalty constant is updated according to

$$k = 2CP(x_o)/P_x^T(x_o)P_x(x_o) \tag{61}$$

where x_o denotes the position vector at the beginning of the first iteration of any conjugate-gradient phase.

Search Scheme. The search for the optimum gradient stepsize ia made on the augmented penalty function

$$W(\tilde{x}, \lambda, k) = W(x-\alpha p, \lambda, k) = W(\alpha) \tag{62}$$

where $\lambda = \lambda_o$ for algorithms of Class I and $\lambda = \lambda_*$ for algorithms of Class II. First, one checks the sign of the derivative

$$W_\alpha(0) = -W_x^T(x, \lambda, k)p \tag{63}$$

If $W_\alpha(0) < 0$, the search for the optimum gradient stepsize is initiated. If $W_\alpha(0) \geq 0$, the conjugate-gradient phase is interrupted and the restoration phase is started. Assuming that $W_\alpha(0) < 0$, one employs any one-dimensional search scheme until the following stopping condition is satisfied:

$$W_\alpha^2(\alpha) \leq \epsilon_4 \tag{64}$$

or

$$W_\alpha^2(\alpha) \leq \epsilon_5 \; W_\alpha^2(0) \tag{65}$$

where

$$W_\alpha(\alpha) = -W_x^T(\tilde{x}, \lambda, k)p \tag{66}$$

5.2. Restoration Phase. The restoration phase is represented by the following equations:

$$\begin{aligned} &\varphi_x^T(x)\varphi_x(x)\sigma - \varphi(x) = 0 \\ &p = \varphi_x(x)\sigma \\ &\Delta x = -\mu p \\ &\tilde{x} = x + \Delta x \end{aligned} \tag{67}$$

For every iteration, the search for the restoration stepsize is made on the constraint error

$$P(\tilde{x}) = P(x - \mu p) = P(\mu) \tag{68}$$

Specifically, one employs a bisection process on μ, starting from $\mu = 1$, until the following inequality is satisfied:

$$P(\mu) < P(0) \tag{69}$$

5.3. Special Conditions. In this section, special conditions relevant to the computer implementation of conjugate gradient-restoration algorithms are presented.

Starting Condition. The algorithms can be started from any nominal point x_s, regardless of whether $\varphi(x_s) = 0$ or $\varphi(x_s) \neq 0$.

Initial Phase. The algorithms are started with a restoration phase if $P(x_s) > \epsilon_1$ and a conjugate-gradient phase if $P(x_s) \le \epsilon_1$.

Restoration Phase: Bypassing Condition. Usually, a complete cycle includes a restoration phase and a conjugate-gradient phase. However, if at the beginning of the restoration phase Ineq. (8-1) is met, the restoration phase is bypassed, and the conjugate-gradient phase is started directly.

Conjugate-Gradient Phase: Stopping Conditions. The conjugate-gradient phase must be stopped under the following conditions: (i) every n-q iterations, or (ii) if $W_\alpha(0) \ge 0$, where $W_\alpha(0)$ is given by Eq. (63).

Conjugate Gradient-Restoration Algorithm: Stopping Condition. A conjugate gradient-restoration algorithm is stopped when Ineqs. (8) are satisfied or Ineq. (9) is satisfied.

6. Numerical Examples

In order to illustrate the theory, five numerical examples were developed using a Burroughs B-5500 computer and double-precision arithmetic. The algorithms were programmed in Extended ALGOL. The constant C was specified to be C = 1.

Concerning the conjugate-gradient phase, the one-dimensional search on the function $W(\alpha)$ was done in accordance with Section 5.1; a modification of quasilinearization was employed; the stopping condition for the one-dimensional search was

$$W_\alpha^2(\alpha) \le W_\alpha^2(0) \times 10^{-6} \tag{70}$$

Concerning the restoration phase, the one-dimensional search on the function $P(\mu)$ was done in accordance with Section 5.2.

Convergence was defined as follows:

$$R(x, \lambda) \le 10^{-12} \tag{71}$$

and the number of iterations for convergence N_* was recorded[5]. Incidentally, satisfaction of Ineq. (71) implies that[6]

$$P(x) \le 10^{-12} \quad , \quad Q(x, \lambda) \le 10^{-12} \tag{72}$$

Conversely, nonconvergence was defined by means of the inequalities

$$N \ge 1000 \tag{73}$$

or

$$N_s \ge 20 \tag{74}$$

---

[5] The number N_* includes both the iterations of the conjugate-gradient phase and the iterations of the restoration phase.

[6] Inequality (72-1) constitutes the bypassing condition for the restoration phase.

Here, N is the iteration number and N_s is the number of bisections of the step-size required to satisfy the inequalities

$$W(\alpha) < W(0) \text{ or } P(\mu) < P(0) \tag{75}$$

Example 6.1. Consider the problem of minimizing the function[7]

$$f = (x - y)^2 + (y + z - 2)^2 + (u - 1)^2 + (w - 1)^2 \tag{76}$$

subject to the constraints

$$x + 3y = 0 \quad , \quad z + u - 2w = 0 \quad , \quad y - w = 0 \tag{77}$$

This function admits the relative minimum f = 4.0930 at the point defined by

$$x = -0.7674 \quad , \quad y = 0.2558 \quad , \quad z = 0.6279 \quad , \quad u = -0.1162 \quad , \quad w = 0.2558 \tag{78}$$

and

$$\lambda_1 = 2.0465 \quad , \quad \lambda_2 = 2.2325 \quad , \quad \lambda_3 = -5.9534 \tag{79}$$

The nominal point chosen to start the algorithm is the point of coordinates

$$x = y = z = u = w = 2 \tag{80}$$

not consistent with (77).

Example 6.2. Consider the problem of minimizing the function

$$f = (x - y)^2 + (y - z)^4 \tag{81}$$

[7] For simplicity, the symbols employed in the examples denote scalar quantities.

subject to the constraint

$$x(1 + y^2) + z^4 - 3 = 0 \tag{82}$$

This function admits the relative minimum f = 0 at the point defined by

$$x = y = z = 1 \tag{83}$$

and

$$\lambda_1 = 0 \tag{84}$$

The nominal point chosen to start the algorithm is the point of coordinates

$$x = y = z = 2 \tag{85}$$

not consistent with (82).

Example 6.3. Consider the problem of minimizing the function

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

subject to the constraint

$$x(1 + y^2) + z^4 - 4 - 3\sqrt{2} = 0 \tag{87}$$

This function admits the relative minimum $f = 0.3256 \times 10^{-1}$ at the point defined by

$$x = 1.1048 \ , \quad y = 1.1966 \ , \quad z = 1.5352 \tag{88}$$

and

$$\lambda_1 = -0.1072 \times 10^{-1} \tag{89}$$

The nominal point chosen to start the algorithm is the point of coordinates

$$x = y = z = 2 \tag{90}$$

not consistent with (87).

Example 6.4. Consider the problem of minimizing the function

$$f = (x - 1)^2 + (x - y)^2 + (z - 1)^2 + (u - 1)^4 + (w - 1)^6 \tag{91}$$

subject to the constraints

$$ux^2 + \sin(u - w) - 2\sqrt{2} = 0 \quad , \quad y + z^4 u^2 - 8 - \sqrt{2} = 0 \tag{92}$$

This function admits the relative minimum f = 0.2415 at the point defined by

$$x = 1.1661 \quad , \quad y = 1.1821 \quad , \quad z = 1.3802 \quad , \quad u = 1.5060 \quad , \quad w = 0.6109 \tag{93}$$

and

$$\lambda_1 = -0.8553 \times 10^{-1} \quad , \quad \lambda_2 = -0.3187 \times 10^{-1} \tag{94}$$

The nominal point chosen to start the algorithm is the point of coordinates

$$x = y = z = u = w = 2 \tag{95}$$

not consistent with (92).

Example 6.5. Consider the problem of minimizing the function

$$f = (x - 1)^2 + (x - y)^2 + (y - z)^2 + (z - u)^4 + (u - w)^4 \tag{96}$$

subject to the constraints

$$x + y^2 + z^3 - 2 - 3\sqrt{2} = 0 \ , \quad y - z^2 + u + 2 - 2\sqrt{2} = 0 \ , \ xw - 2 = 0 \qquad (97)$$

This function admits the relative minimum $f = 0.7877 \times 10^{-1}$ at the point defined by

$$x = 1.1911 \ , \quad y = 1.3626 \ , \quad z = 1.4728 \ , \quad u = 1.6350 \ , \quad w = 1.6790 \quad (98)$$

and

$$\lambda_1 = -0.3882 \times 10^{-1} \ , \quad \lambda_2 = -0.1672 \times 10^{-1} \ , \quad \lambda_3 = -0.2879 \times 10^{-3} \qquad (99)$$

The nominal point chosen to start the algorithm is the point of coordinates

$$x = y = z = u = w = 2 \qquad (100)$$

not consistent with (97).

7. Results and Conclusions

For the previous examples and experimental conditions, the conjugate-gradient restoration algorithms of Class I and Class II were tested in versions (α) and (β). In the former, the penalty constant is held unchanged throughout the entire algorithm; in the latter, the penalty constant is held unchanged only throughout each conjugate-gradient phase. The numerical results are given in Tables 1-4, where the number of iterations at convergence N_* is shown. From the tables, the following conclusions arise:

(a) For the linear-quadratic Example 6.1, all the algorithms behave identically as predicted by the theory. For these algorithms, quadatic convergence (that is, convergence in $N_* = 1+n-q$ iterations) is verified.

(b) For the nonlinear-nonquadratic Examples 6.2 through 6.5, the algorithms do not behave identically. A detailed analysis is given below.

(c) In general, algorithms of Class II are superior to algorithms of Class I, in that they require a smaller number of iterations for convergence.

(d) In general, algorithms of type (β) are superior to algorithms of type (α) for two reasons: (i) in algorithms of type (β), the penalty constant is not arbitrary but is determined so that certain special properties are enforced for the first iteration of the conjugate-gradient phase; and (ii) the number of iterations at convergence for algorithms of type (β) is close to the minimum with respect to k of the number of iterations at convergence for algorithms of type (α).

In closing, it is of interest to compare the present algorithms with the algorithm developed by Haarhoff and Buys in Ref. 5. While all these algorithms include a restoration phase involving one iteration and a conjugate-gradient phase involving n-q iterations, two main differences must be noted: (i), here, the search for the restoration stepsize is done on the function $P(\mu)$; in Ref. 5, the search for the restoration stepsize is done on the function $W(\mu)$; (ii) here, the Lagrange multiplier λ is updated at the beginning of each iteration of the conjugate-gradient phase; in Ref. 5, the Lagrange multiplier λ has a constant value during each conjugate-gradient phase; this constant value is computed at the beginning of the previous restoration phase.

If the function $f(x)$ is quadratic, if the constraint $\varphi(x)$ is linear, and if the starting point x_s is such that $\varphi(x_s) \neq 0$, the present algorithms converge to the solution in at most $N_* = 1+n-q$ iterations, while this is not the case with the algorithm of Ref. 5. Therefore, the present algorithms should exhibit faster convergence than the algorithm of Ref. 5, even if the function $f(x)$ is nonquadratic and/or the constraint $\varphi(x)$ is nonlinear.

Table 1. Number of iterations at convergence N_* for Algorithm (I-α).

k	Example 6.1	6.2	6.3	6.4	6.5
10^{-4}	3	20	15	17	11
10^{-3}	3	20	13	14	11
10^{-2}	3	20	9	24	9
10^{-1}	3	17	18	46	12
10^{0}	3	55	18	41	15
10^{1}	3	495	38	84	29
10^{2}	3	>1000	36	117	380
10^{3}	3	>1000	68	178	>1000
10^{4}	3	>1000	120	186	>1000

Table 2. Number of iterations at convergence N_* for Algorithm (I-β).

k	Example 6.1	6.2	6.3	6.4	6.5
Eq. (59)	3	20	11	15	11

Table 3. Number of iterations at convergence N_* for Algorithm (II-α).

k	Example				
	6.1	6.2	6.3	6.4	6.5
10^{-4}	3	15	12	13	10
10^{-3}	3	23	12	13	10
10^{-2}	3	16	12	17	9
10^{-1}	3	23	15	24	11
10^{0}	3	37	15	48	11
10^{1}	3	62	61	100	14
10^{2}	3	142	>1000	118	25
10^{3}	3	>1000	>1000	120	26
10^{4}	3	>1000	>1000	>1000	39

Table 4. Number of iterations at convergence N_* for Algorithm (II-β).

k	Example				
	6.1	6.2	6.3	6.4	6.5
Eq. (61)	3	20	12	13	9

References

1. MIELE, A., CRAGG, E.E., IYER, R.R., and LEVY, A.V., Use of the Augmented Penalty Function in Mathematical Programming Problems, Part 1, Ordinary Gradient Algorithm, Rice University, Aero-Astronautics Report No. 75, 1970.

2. HESTENES, M.R., Multiplier and Gradient Methods, Journal of Optimization Theory and Applications, Vol. 4, No. 5, 1969.

3. MIELE, A., HUANG, H.Y., and HEIDEMAN, J.C., Sequential Gradient-Restoration Algorithm for the Minimization of Constrained Functions, Ordinary and Conjugate Gradient Versions, Journal of Optimization Theory and Applications, Vol. 4, No. 4, 1969.

4. MIELE, A., LEVY, A.V., and CRAGG, E.E., Modifications and Extensions of the Conjugate Gradient-Restoration Algorithm for Mathematical Programming Problems, Journal of Optimization Theory and Applications, Vol. 7, No. 6, 1971.

5. HAARHOFF, P.C., and BUYS, J.D., A New Method for the Optimization of a Nonlinear Function Subject to Nonlinear Constraints, Computer Journal, Vol. 13, No. 2, 1970.