RICE UNIVERSITY

METHOD OF INDEPENDENT MULTIPLIERS
FOR MINIMIZING UNCONSTRAINED FUNCTIONS

by

J.W. CANTRELL

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

Thesis Director's signature

Houston, Texas

March, 1969
Abstract

METHOD OF INDEPENDENT MULTIPLIERS
FOR MINIMIZING UNCONSTRAINED FUNCTIONS

by

J.W. CANTRELL

A new accelerated gradient method for finding the minimum of a function
\( f(x) \) whose variables are unconstrained is presented. The new algorithm can be
stated as follows:

\[
\bar{x} = x + \delta x, \quad \delta x = - \alpha g(x) + \beta \delta \bar{x}
\]

where \( \delta x \) is the change in the position vector \( x \), \( g(x) \) is the gradient of the function
\( f(x) \), and \( \alpha \) and \( \beta \) are scalars chosen at each step so as to yield the greatest
decrease in the function. The symbol \( \delta \bar{x} \) denotes the change in the position vector
for the iteration preceding that under consideration.

It is shown that, for a quadratic function, the present algorithm reduces to
the Fletcher-Reeves algorithm; thus, quadratic convergence is assured. However,
for a nonquadratic function, initial convergence of the present method is much
faster than that of the Fletcher-Reeves method because of the extra degree of
freedom available. For a test problem, the number of iterations was about 40-50% 
that of the Fletcher-Reeves method and the computing time about 60-75% that of the
Fletcher-Reeves method, using comparable search techniques.
ACKNOWLEDGEMENTS

The author expresses his appreciation to Dr. Angelo Miele for suggesting the topic and stimulating discussion. Also he acknowledges the support of the Office of Scientific Research, Office of Aerospace Research, United States Air Force, Grant No. AF-AFOSR-828-67.
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2. Definitions</td>
<td>2</td>
</tr>
<tr>
<td>3. Statement of the Problem</td>
<td>4</td>
</tr>
<tr>
<td>4. Review of the Gradient Method</td>
<td>6</td>
</tr>
<tr>
<td>5. Review of the Fletcher-Reeves Method</td>
<td>11</td>
</tr>
<tr>
<td>6. Method of Independent Multipliers</td>
<td>16</td>
</tr>
<tr>
<td>7. Relation of the Independent Multiplier Method and the Fletcher-Reeves Method</td>
<td>23</td>
</tr>
<tr>
<td>8. Numerical Example</td>
<td>27</td>
</tr>
<tr>
<td>9. Discussion and Conclusions</td>
<td>33</td>
</tr>
<tr>
<td>References</td>
<td>35</td>
</tr>
<tr>
<td>List of Captions</td>
<td>36</td>
</tr>
</tbody>
</table>
1. INTRODUCTION

In Ref. 1, the Fletcher-Reeves conjugate gradient algorithm for finding the minimum of a function \( f(x) \) whose variables are unconstrained was presented. Compared with the ordinary gradient method, this algorithm has the advantage of high speed since it produces quadratic convergence. Compared with other conjugate gradient methods (such as the Davidon variable-metric algorithm), it has the advantage of simplicity of concept and small storage requirement while yielding comparable computing time.

In this thesis, a generalization of the Fletcher-Reeves algorithm is investigated. This generalization retains the property of quadratic convergence, simplicity of concept, and small storage requirement, while yielding shorter computing time. The only added complication is the need for a two-dimensional search at each iteration as opposed to the one-dimensional search required by the Fletcher-Reeves algorithm.
2. **DEFINITIONS**

The following definitions are used throughout the paper:

(a) The symbol \( x \) denotes the position vector

\[
\begin{pmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_k \\
\end{pmatrix}
\]

whose scalar components are \( x_1, x_2, \ldots, x_k \).

(b) The symbol \( f \) denotes a scalar function of the vector \( x \), that is,

\[
f = f(x) \tag{2}
\]

(c) The symbol \( g \) denotes the column vector

\[
\begin{pmatrix}
  f \\
  x_1 \\
  f \\
  x_2 \\
  \vdots \\
  \vdots \\
  f \\
  x_k \\
\end{pmatrix}
\]

whose components are the first partial derivatives of \( f \) with respect to the scalar variables \( x_1, x_2, \ldots, x_k \). This is the gradient of the function \( f \).

(d) The symbol \( h \) denotes the square matrix
whose components are the second partial derivatives of the function $f$ with respect to the scalar variables $x_1, x_2, \ldots, x_k$.

(e) The symbol $x$ denotes the nominal point. The symbol $\tilde{x}$ denotes the point following $x$. The symbol $\hat{x}$ denote the point preceding $x$. The symbol $\check{x}$ denotes the point preceeding $\hat{x}$ (see Fig. 1).

(f) The symbol $\delta(\ldots)$ denotes the displacement leading from a point to the next point. Therefore, the following relations hold (see Fig. 1):

\[
\begin{align*}
\tilde{x} &= x + \delta x \\
x &= \hat{x} + \delta \hat{x} \\
\check{x} &= \hat{\check{x}} + \delta \hat{\check{x}}
\end{align*}
\]

(g) The prime denotes the transpose of a matrix.
3. **STATEMENT OF THE PROBLEM**

The purpose of this paper is to find the minimum of a function

$$ f = f(x) $$

whose variables are unconstrained. The basic idea is to construct corrections $\delta x$ leading from a nominal point $x$ to a varied point $\bar{x}$ such that

$$ f(\bar{x}) < f(x) $$

Therefore, by an iterative procedure (that is, through successive decreases in the value of the function), it is hoped that the minimum of $f$ is approached to any desired degree of accuracy.

We consider corrections $\delta x$ having the following form:

$$ \delta x = - \alpha g(x) + \beta \delta \bar{x} $$

If

$$ R = 0 $$

the algorithm (8) reduces to the ordinary gradient method. If

$$ \beta = \frac{ag'(x)g(x)}{\delta g'(\bar{x})g(\bar{x})} $$

the algorithm (8) reduces to the Fletcher-Reeves method.
Generally speaking, the convergence properties of the Fletcher-Reeves method are better than those of the ordinary gradient method. In particular, for a quadratic function of \( k \) scalar variables, (10) converges in no more than \( k \) steps to the exact solution while (9) does not converge in a predetermined number of steps. Since the algorithm (10) is based on the consideration of quadratic functions, it is not immediately clear that its properties should be excellent for nonquadratic functions as well. For these functions, it is felt that the addition of one extra degree of freedom to the system of corrections \( \delta x \) should substantially improve convergence, at least in the initial stage of a descent process. Therefore, it is the purpose of this research to employ the algorithm (8) with \( \alpha \) and \( \beta \) optimized independently. This is why this method is called the method of independent multipliers.

In the following sections, we first review the ordinary gradient method and the Fletcher-Reeves method. Then, we derive the properties of the method of independent multipliers.
4. REVIEW OF THE GRADIENT METHOD

In the ordinary gradient method, one considers corrections having the form

$$\delta x = - \alpha g(x)$$  \hspace{1cm} (11)

so that

$$\bar{x} = x - \alpha g(x)$$  \hspace{1cm} (12)

For each point $x$, Eq. (12) defines a one-parameter family of points $\bar{x}$ for which the function $f$ takes the form

$$f(\bar{x}) = f(x - \alpha g(x)) = F(\alpha)$$  \hspace{1cm} (13)

4.1. Properties of the Algorithm. The greatest decrease in the function $F$ occurs if the parameter $\alpha$ satisfies the necessary condition

$$\frac{F}{\alpha} = 0$$  \hspace{1cm} (14)

On account of (13), Eq. (14) becomes

$$g'(\bar{x})g(x) = 0$$  \hspace{1cm} (15)

and shows that the gradient $g(\bar{x})$ is orthogonal to the gradient $g(x)$. Note that $g(\bar{x})$ is also orthogonal to the correction $\delta x$, since $\delta x$ and $g(x)$ are parallel.

If Eq. (11) is premultiplied by $g'(x)$, the following result is obtained:

$$g'(x)\delta x = - \alpha g'(x)g(x)$$  \hspace{1cm} (16)
Since the first variation of the function $f$ is given by

$$\delta f = g'(x)\delta x$$  \hspace{1cm} (17)$$

Eq. (16) can be rewritten as

$$\delta f = -\alpha g'(x)g(x)$$  \hspace{1cm} (18)$$

Since $g'(x)g(x) > 0$, Eq. (18) shows that the first variation is negative for $\alpha > 0$. Therefore, the function $f$ decreases for $\alpha$ sufficiently small.

The complete algorithm can be summarized as follows: (a) for a given nominal point $x$, the gradient $g(x)$ is known; (b) the optimum value of the multiplier $\alpha$ must be determined by solving Eq. (14), as in Section 4.2; (c) the correction $\delta x$ to the position vector $x$ is determined using Eq. (11); and (d) the new position vector $\bar{x}$ is computed through Eq. (5-1).

4.2. Search Technique. The next step is to solve Eq. (14) for the optimum value of the parameter $\alpha$. Let $\delta\alpha = \alpha - \alpha_0$ denote the correction to $\alpha$ starting from an arbitrary nominal value $\alpha_0$. If quasilinearization is applied to Eq. (14), one obtains the linear algebraic equation

$$F_{\alpha\alpha} \delta\alpha + F_\alpha = 0$$  \hspace{1cm} (19)$$

where $F_\alpha$ and $F_{\alpha\alpha}$ are computed at $\alpha_0$. Upon introducing the scaling factor $u$, $0 \leq u \leq 1$, and imbedding Eq. (19) in the more general equation

$$F_{\alpha\alpha} \delta\alpha + F_\alpha' u = 0$$  \hspace{1cm} (20)$$
we obtain the solution

\[ \delta_\alpha = - \mu F_\alpha / F_{\alpha \alpha} \quad (21) \]

where \( F_\alpha \) and \( F_{\alpha \alpha} \) are computed at \( \alpha_0 \) in the manner indicated below.

For the one-parameter family of points (12), the gradient \( g \) takes the form

\[ g(\bar{x}) = g(x - \alpha g(x)) = G(\alpha) \quad (22) \]

and the second-derivative matrix \( h \) becomes

\[ h(\bar{x}) = h(x - \alpha g(x)) = H(\alpha) \quad (23) \]

As a consequence, the partial derivatives appearing in Eq. (21) can be computed from the expressions

\[ F_\alpha (\alpha_0) = - G'(\alpha_0) g(x) \quad (24) \]

and

\[ F_{\alpha \alpha} (\alpha_0) = g'(x) H(\alpha_0) g(x) \quad (25) \]

If the matrix \( H \) is not explicitly available, the second derivative cannot be computed with Eq. (25). In this case, one can use the following difference scheme:

\[ F_{\alpha \alpha} (\alpha_0) = (1/2\varepsilon)[ F_\alpha (\alpha_0 + \varepsilon) - F_\alpha (\alpha_0 - \varepsilon) ] \quad (26) \]

where \( \varepsilon \) is a small number.
4.3. **Remark.** The first variation of the function $F(\alpha)$ is given by

$$\delta F = F\frac{\delta \alpha}{\alpha} \quad (27)$$

and, for $\delta \alpha$ given by Eq. (21), becomes

$$\delta F = -\mu F\frac{2}{F_{\alpha \alpha}} \quad (28)$$

If $F_{\alpha \alpha} > 0$, the first variation is negative, and the function $F$ decreases for $\mu$ sufficiently small. If $F_{\alpha \alpha} < 0$, the first variation is positive, and the function $F$ increases for $\mu$ sufficiently small. In order to prevent the latter situation and force the function $F$ to decrease in every case, it is convenient to replace (21) with the expression

$$\delta \alpha = -\mu (F_{\alpha} / F_{\alpha \alpha}) \text{ sign } (F_{\alpha \alpha}) \quad (29)$$

so that (27) becomes

$$\delta F = -\mu (F_{\alpha}^2 / F_{\alpha \alpha}) \text{ sign } (F_{\alpha \alpha}) \quad (30)$$

Clearly, the first variation (30) is now negative regardless of the sign of $F_{\alpha \alpha}$.

To perform the search, a nominal value must be given to $\alpha_0$. Then, one sets $\mu = 1$, computes $\delta \alpha$ from (29), and $\alpha$ from $\alpha = \alpha_0 + \delta \alpha$. If $F(\alpha) < F(\alpha_0)$, the scaling factor $\mu = 1$ is acceptable. If $F(\alpha) > F(\alpha_0)$, the previous value of $\mu$ must be replaced by some smaller value in the range $0 \leq \mu \leq 1$ until the condition $F(\alpha) < F(\alpha_0)$ is met. At this point, the search step is completed. The value obtained for $\alpha$ becomes (*)

These smaller values of $\mu$ can be obtained by successively dividing $\mu$ by 2.
the nominal value $\alpha_0$ for the next search step, and the procedure is repeated until a desired degree of accuracy on $\alpha$ is obtained. In the absence of better information, the first step of the search procedure can be made with $\alpha_0 = 0$. 
5. **REVIEW OF THE FLETCHER-REEVES METHOD**

In the Fletcher-Reeves method, one considers corrections having the form

\[ \delta x = - \alpha p(x) \]  

(31)

where

\[ p(x) = g(x) + \frac{g'(x)g(x)}{g'(\hat{x})g(\hat{x})} p(\hat{x}) \]  

(32)

so that

\[ \hat{x} = x - \alpha p(x) \]  

(33)

For each point \( x \), Eq. (33) defines a one-parameter family of points \( \hat{x} \) for which the function \( f \) takes the form

\[ f(\hat{x}) = f(x - \alpha p(x)) = F(\alpha) \]  

(34)

If one observes that

\[ \delta \hat{x} = - \hat{\alpha} p(\hat{x}) \]  

(35)

Eq. (32) becomes

\[ p(x) = g(x) - \frac{g'(x)g(x)}{g'(\hat{x})g(\hat{x})} \frac{\delta \hat{x}}{\delta} \]  

(36)

and Eq. (31) can be written as

\[ \delta x = - \alpha g(x) + \frac{\alpha g'(x)g(x)}{\hat{\alpha} g'(\hat{x})g(\hat{x})} \delta \hat{x} \]  

(37)
This is precisely (8) with $\beta$ defined as in (10).

5.1. Properties of the Algorithm. The greatest decrease in the function $F$ occurs if the parameter $\alpha$ satisfies the necessary condition

$$F_{\alpha} = 0$$  \hspace{1cm} (38)

On account of (34), Eq. (38) becomes

$$g'(x)p(x) = 0$$  \hspace{1cm} (39)

and shows that the gradient $g(x)$ is orthogonal to the search direction $p(x)$. Note that $g(x)$ is also orthogonal to the correction $\delta x$, since $\delta x$ and $p(x)$ are parallel. That is,

$$g'(x)\delta x = 0$$  \hspace{1cm} (40)

If Eq. (37) is premultiplied by $g'(x)$, the following result is obtained:

$$g'(x)\delta x = -\alpha g'(x)g(x) + \frac{\alpha g'(x)g(x)}{\hat{g}'(x)g(x)} g'(x)\delta x$$  \hspace{1cm} (41)

Because of Eqs. (17) and (40), Eq. (41) becomes

$$\delta f = -\alpha g'(x)g(x)$$  \hspace{1cm} (42)

Since $g'(x)g(x) > 0$, Eq. (42) shows that the first variation is negative for $\alpha > 0$. Therefore, the function $f$ decreases for $\alpha$ sufficiently small.

For any iteration except the first, the complete algorithm can be summarized as follows: (a) for a given nominal point $x$, the gradient $g(x)$ is known, the vector
p(\hat{x}) is known from the previous iteration, and p(x) can be computed through (32); 
(b) the optimum value of the multiplier \( \alpha \) must be determined by solving Eq. (38), 
as in Section 5.3; (c) the correction \( \delta x \) to the position vector \( x \) is determined 
using Eq. (31); and (d) the new position vector \( \tilde{x} \) is computed through Eq. (5-1).

Of course, operations (a) through (d) imply that p(\hat{x}) is known from the 
previous iteration. Since this is not the case for the first iteration, an arbitrary 
assumption concerning p(\hat{x}) must be made in order to start the algorithm: the 
simplest assumption is p(\hat{x}) = 0, equivalent to stating that the first step is a 
gradient step. For this algorithm, numerical experience has shown the desirability 
of restarting the process every \( k \) or \( k + 1 \) iterations (Ref. 1).

Consider now a quadratic function, that is, a function of the type

\[ f(x) = a + b'x + \frac{1}{2} x'c x \]

(43)

where \( a \) is a scalar, \( b \) is a \( k \)-vector, and \( c \) is a \( k \times k \) symmetric matrix. For 
this function, the gradient \( g(x) \) is the linear function

\[ g(x) = b + cx \]

(44)

and the matrix of the second derivatives is the constant matrix

\[ h(x) = c \]

(45)

Then, if \( i \) and \( j \) are subscripts denoting two different iterations, the following 
relations hold (Ref. 2):

\[ [g'(x)]_i [g(x)]_j = 0, \quad i \neq j \]

(46)
and

\[ [p'(x)]_i h(x) [p(x)]_j = 0 \quad i \neq j \quad (47) \]

Equation (46) states that the gradient at each iteration is orthogonal to the gradient at every other iteration. Equation (47) shows that the search direction at each iteration and the search direction at every other iteration are conjugate with respect to the matrix \( h \) (in the particular case, the constant matrix \( c \)). At the end of \( k \) iterations, the gradient must be zero: this is the only vector orthogonal to \( k \) mutually orthogonal vectors in a \( k \)-dimensional space. Therefore, except for round-off errors, the minimum of \( f \) is reached in no more than \( k \) iterations.

5.2. **Search Technique.** The next step is to solve Eq. (38) for the optimum value of the parameter \( \alpha \). Let \( \delta \alpha = \alpha - \alpha_o \) denote the correction to \( \alpha \) starting from an arbitrary nominal value \( \alpha_o \). If quasilinearization is applied to Eq. (38), one obtains the linear algebraic equation (19). Upon imbedding (19) into (20), one obtains the solution (21). If the possibility exists that \( \frac{\partial f}{\partial \alpha} < 0 \), then (21) must be replaced by (29).

For the one-parameter family of points (33), the gradient \( g \) takes the form

\[ g(\bar{x}) = g(\bar{x} - \alpha p(x)) = G(\alpha) \quad (48) \]

and the matrix \( h \) becomes

\[ \bar{h}(\bar{x}) = h(\bar{x} - \alpha p(x)) = H(\alpha) \quad (49) \]
As a consequence, the partial derivatives appearing in (21) or (29) can be computed from the expressions

\[ F_{\alpha_0} = -G'(\alpha_0)p(x) \]  \hspace{1cm} (50)

and

\[ F_{\alpha\alpha_0} = p'(x)H(\alpha_0)p(x) \]  \hspace{1cm} (51)

If the matrix \( H \) is not explicitly available, the second derivative cannot be computed with Eq. (51). In this case, one can use the difference scheme (26), where \( \varepsilon \) is a small number.
6. METHOD OF INDEPENDENT MULTIPLIERS

With this method, one considers corrections having the form (8), that is,

\[ \delta x = -\alpha g(x) + \beta \delta \tilde{x} \]  

(52)

where \( \alpha \) and \( \beta \) are independent multipliers. If Eqs. (51) and (52) are combined, the position vector at the end of any iteration becomes

\[ \tilde{x} = x - \alpha g(x) + \beta \delta \tilde{x} \]  

(53)

For each point \( x \), Eq. (53) defines a two-parameter family of points \( \tilde{x} \) for which the function \( f \) takes the form

\[ f(\tilde{x}) = f(x - \alpha g(x) + \beta \delta \tilde{x}) = F(\alpha, \beta) \]  

(54)

6.1. Properties of the Algorithm. The greatest decrease in the function \( F \) occurs if the parameters \( \alpha \) and \( \beta \) satisfy the following necessary conditions:

\[ F_\alpha = 0 \ , \ F_\beta = 0 \]  

(55)

where the subscripts denote partial derivatives. On account of (54), Eqs. (55) become

\[ g'(\tilde{x})g(x) = 0 \ , \ g'(\tilde{x})\delta \tilde{x} = 0 \]  

(56)

and show that the gradient \( g(\tilde{x}) \) is orthogonal to the gradient \( g(x) \) and the correction \( \delta \tilde{x} \).
If Eq. (52) is premultiplied by $g'(\bar{x})$, the following result is obtained:

$$g'(\bar{x})\delta x = -\alpha g'(\bar{x})g(x) + \beta g'(\bar{x})\delta \bar{x}$$ (57)

which, in the light of (56), implies that

$$g'(\bar{x})\delta x = 0$$ (58)

Therefore, $g(\bar{x})$ is also orthogonal to the correction $\delta x$.

If Eq. (52) is premultiplied by $g'(x)$, the following result is obtained:

$$g'(x)\delta x = -\alpha g'(x)g(x) + \beta g'(x)\delta \bar{x}$$ (59)

Because of (17) and (58), Eq. (59) becomes

$$\delta f = -\alpha g'(x)g(x)$$ (60)

Since $g'(x)g(x) > 0$, Eq. (60) shows that the first variation is negative for $\alpha > 0$.

Therefore, the function $f$ decreases for $\alpha$ sufficiently small. Equations (56), (58), (60) summarize the general properties of the algorithm. They are valid regardless of the function $f(x)$, as long as it is continuous and has continuous first derivatives.

For any iteration except the first, the complete algorithm can be summarized as follows: (a) for a given nominal point $x$, the gradient $g(x)$ is known, and the vector $\delta \bar{x}$ is known from the previous iteration; (b) the optimum values of the multipliers $\alpha$ and $\beta$ must be determined by solving Eqs. (55), as in Section 6.2; (c) the correction $\delta x$ to the position vector $x$ is determined using Eq. (52); and (d) the new position vector $\bar{x}$ is computed through Eq. (5-1).
Of course, operations (a) through (d) imply that $\delta \alpha$ is known from the previous iteration. Since this is not the case for the first iteration, an arbitrary assumption concerning $\delta \alpha$ must be made in order to start the algorithm: the simplest assumption is $\delta \alpha = 0$, equivalent to stating that the first step is a gradient step. Since the independent multiplier algorithm is a modification of the Fletcher-Reeves algorithm, it is probable that restarting the process every $k$ or $k + 1$ iterations may prove helpful.

6.2. Search Technique. The next step is to solve Eqs. (55) for the optimum values of the parameters $\alpha$ and $\beta$. Let $\delta \alpha = \alpha - \alpha_o$ and $\delta \beta = \beta - \beta_o$ denote the corrections to $\alpha$ and $\beta$ starting from arbitrary nominal values $\alpha_o$ and $\beta_o$. If quasilinearization is applied to Eqs. (55), one obtains the linear algebraic equations

$$F_{\alpha \alpha} \delta \alpha + F_{\alpha \beta} \delta \beta + F_\alpha = 0$$
$$F_{\beta \alpha} \delta \alpha + F_{\beta \beta} \delta \beta + F_\beta = 0$$

(61)

where $F_{\alpha \alpha}, F_{\beta \beta}, F_{\alpha \beta}, F_{\beta \alpha}$ are computed at $\alpha_o, \beta_o$. Upon introducing the scaling factor $\mu, 0 \leq \mu \leq 1$, and imbedding Eqs. (61) in the more general equations

$$F_{\alpha \alpha} \delta \alpha + F_{\alpha \beta} \delta \beta + \mu F_\alpha = 0$$
$$F_{\beta \alpha} \delta \alpha + F_{\beta \beta} \delta \beta + \mu F_\beta = 0$$

(62)

we obtain the solutions

$$\delta \alpha = -\mu D_1/D_3, \delta \beta = -\mu D_2/D_3$$

(63)
where

\[ D_1 = F_{\alpha \beta} F_{\alpha \beta} - F_{\beta} F_{\alpha \alpha} \]

\[ D_2 = F_{\beta} F_{\alpha \alpha} - F_{\alpha} F_{\alpha \beta} \]

\[ D_3 = F_{\alpha \alpha} F_{\beta \beta} - F_{\alpha \beta}^2 \]  \hfill (64)

For the two-parameter family of points (53), the gradient \( g \) takes the form

\[ g(\delta) = g(x + \alpha g(x) + \beta \delta \delta) = G(\alpha, \beta) \]  \hfill (65)

and the second-derivative matrix \( h \) becomes

\[ h(\delta) = h(x + \alpha g(x) + \beta \delta \delta) = H(\alpha, \beta) \]  \hfill (66)

As a consequence, the partial derivatives appearing in Eqs. (64) can be computed from the expressions

\[ F_{\alpha \alpha} (\alpha_0, \beta_0) = -G'(\alpha_0, \beta_0) g(x) \]  \hfill (67)

\[ F_{\beta \beta} (\alpha_0, \beta_0) = G'(\alpha_0, \beta_0) \delta \delta \]

and

\[ F_{\alpha \beta} (\alpha_0, \beta_0) = g'(x) H(\alpha_0, \beta_0) g(x) \]

\[ F_{\alpha \beta} (\alpha_0, \beta_0) = g'(x) H(\alpha_0, \beta_0) \delta \delta \]  \hfill (68)
If the matrix $H$ is not explicitly available, the second derivatives cannot be computed with Eqs. (68). In this case, one can use the following difference scheme:

$$
\delta F_{\alpha \alpha}(\alpha_0, \beta_0) = (1/2 \epsilon_1) \left[ F_{\alpha}(\alpha_0 + \epsilon_1, \beta_0) - F_{\alpha}(\alpha_0 - \epsilon_1, \beta_0) \right]
$$

$$
\delta F_{\alpha \beta}(\alpha_0, \beta_0) = (1/2 \epsilon_2) \left[ F_{\alpha}(\alpha_0, \beta_0 + \epsilon_2) - F_{\alpha}(\alpha_0, \beta_0 - \epsilon_2) \right]
$$

$$
= (1/2 \epsilon_1) \left[ F_{\beta}(\alpha_0 + \epsilon_1, \beta_0) - F_{\beta}(\alpha_0 - \epsilon_1, \beta_0) \right]
$$

$$
\delta F_{\beta \beta}(\alpha_0, \beta_0) = (1/2 \epsilon_2) \left[ F_{\beta}(\alpha_0, \beta_0 + \epsilon_2) - F_{\beta}(\alpha_0, \beta_0 - \epsilon_2) \right]
$$

where $\epsilon_1$ and $\epsilon_2$ are small numbers. In practice, one may choose $\epsilon_1 = \epsilon/|g(\alpha)|$, $\epsilon_2 = \epsilon/|\delta \beta|$, where $\epsilon$ is a small quantity.

6.3. Remark. The first variation of the function $F(\alpha, \beta)$ is given by

$$
\delta F = F_{\alpha} \delta \alpha + F_{\beta} \delta \beta
$$

and, for $\delta \alpha$ and $\delta \beta$ given by Eqs. (63), becomes

$$
\delta F = - \mu D
$$

where

$$
D = (F_{\alpha} D_1 + F_{\beta} D_2)/D_3
$$

If $D > 0$, the first variation is negative, and the function $F$ decreases for $\mu$ sufficiently small. If $D < 0$, the first variation is positive, and the function $F$ increases for $\mu$ sufficiently small. In order to prevent the latter situation and force the function $F$ to decrease in every case, it is convenient to replace (63)
with the expressions

\[ \delta \alpha = - \mu (D_1 / D_3) \text{sign}(D), \quad \delta \beta = - \mu (D_2 / D_3) \text{sign}(D) \]  

(73)

so that (70) becomes

\[ \delta F = - \mu D \text{sign}(D) \]  

(74)

Clearly, the first variation (74) is now negative regardless of the sign of D.

To perform the search, nominal values must be given to \( \alpha_0, \beta_0 \). Then, one sets \( \mu = 1 \), computes \( \delta \alpha, \delta \beta \) from Eqs. (73), and \( \alpha, \beta \) from \( \alpha = \alpha_0 + \delta \alpha, \beta = \beta_0 + \delta \beta \). If \( F(\alpha, \beta) < F(\alpha_0, \beta_0) \), the scaling factor \( \mu = 1 \) is acceptable. If \( F(\alpha, \beta) > F(\alpha_0, \beta_0) \), the previous value of \( \mu \) must be replaced by some smaller value in the range \( 0 < \mu < 1 \) until the condition \( F(\alpha, \beta) < F(\alpha_0, \beta_0) \) is met. At this point, the search step is completed. The values obtained for \( \alpha, \beta \) become the nominal values \( \alpha_0, \beta_0 \) for the next search step, and the procedure is repeated until a desired degree of accuracy on \( \alpha, \beta \) is obtained. In the absence of better information, the first step in the search procedure can be made with \( \alpha_0 = \beta_0 = 0 \).

6.4. **Remark.** Whenever the independent multiplier algorithm is started or restarted, the assumption \( \delta \xi = 0 \) must be made, with the implication that

\[ F_{\beta} = F_{\alpha \beta} = F_{\beta \beta} = 0 \]  

(75)

and that

\[ D_1 = D_2 = D_3 = 0 \]  

(76)
Since Eqs. (73) lead to undetermined values for $\delta \alpha$ and $\delta \beta$, one must reexamine Eqs. (62). Specifically, Eq. (62-2) becomes an identity and Eq. (62-1) is solved by

$$\delta \alpha = - \mu F_{\alpha} / F_{\alpha \alpha}$$

(77)

where $F_{\alpha}$ and $F_{\alpha \alpha}$ are supplied by Eqs. (67-1) and (68-1). Equation (77) is valid for $F_{\alpha \alpha} > 0$. If the possibility exists that $F_{\alpha \alpha} < 0$, Eq. (77) must be replaced by

$$\delta \alpha = - \mu (F_{\alpha} / F_{\alpha \alpha}) \text{sign}(F_{\alpha \alpha})$$

(78)

Note that, for the starting or restarting step, the multiplier $\beta$ remains undetermined and has no effect on the algorithm, since $\beta \delta \hat{x} = 0$. 
7. **RELATION OF THE INDEPENDENT MULTIPLIER METHOD AND THE FLETCHER-REEVES METHOD**

Consider a quadratic function $f(x)$, that is, a function of the type (43), where $a$ is a scalar, $b$ is a $k$-vector, and $c$ is a $k \times k$ symmetric matrix. For this function, the gradient $g(x)$ is the linear function (44) and the matrix of the second derivatives $h(x)$ is the constant matrix (45). At the point $\bar{x}$, Eq. (44) becomes

$$g(\bar{x}) = b + c\bar{x} = b + c(x + \delta x) = g(x) + c\delta x$$

(79)

with the implication that

$$g'(\bar{x}) = g'(x) + \delta x'c$$

(80)

Therefore, the conditions (56) optimizing $\alpha$ and $\beta$ become

$$g'(x)g(x) + \delta x'cg(x) = 0$$
$$g'(x)\delta \bar{x} + \delta x'c\delta \bar{x} = 0$$

(81)

After observing that

$$\delta x' = -\alpha g'(x) + \beta \delta \bar{x}'$$

(82)

and accounting for (58), we see that Eqs. (81) reduce to

$$g'(x)g(x) - \alpha g'(x)c g(x) + \beta g'(x)c \delta \bar{x} = 0$$
$$-\alpha g'(x)c \delta \bar{x} + \beta \delta \bar{x}'c \delta \bar{x} = 0$$

(83)
and admit the solutions

\[ \alpha = \lambda \delta \xi' \delta \xi \]
\[ \beta = \lambda g'(x) \delta \xi \]

where \( \lambda \) is a scalar given by

\[ \lambda = \frac{g'(x)g(x)}{[g'(x)cg(x)][\delta \xi' \delta \xi] - [g'(x)c \delta \xi]^2} \]

From (84), we deduce the following relation between the optimum parameters \( \alpha \) and \( \beta \):

\[ \beta = \frac{\alpha g'(x) \delta \xi}{\delta \xi' \delta \xi} \]

(86)

If Eq. (79) is rewritten for the previous iteration, one obtains the relation

\[ g(x) = g(\hat{x}) + c \delta \xi \]

(87)

which, upon premultiplication by \( g'(x) \), leads to

\[ g'(x)g(x) = g'(x)g(\hat{x}) + g'(x)c \delta \xi \]

(88)

Therefore, on account of (56-1), one concludes that

\[ g'(x)c \delta \xi = g'(x)g(x) \]

(89)

Next, if Eq. (87) is premultiplied by \( \delta \xi' \), one obtains

\[ \delta \xi''g(x) = \delta \xi' g(\hat{x}) + \delta \xi' c \delta \xi \]

(90)
On account of Eq. (58), Eq. (90) reduces to

\[ \delta \dot{\chi} \delta \dot{\chi} = - \delta \dot{\chi} g(\dot{\chi}) \]  

(91)

Owing to the fact that

\[ \delta \dot{\chi} = - \hat{\alpha} g(\dot{\chi}) + \hat{\beta} \delta \dot{\chi} \]  

(92)

and that

\[ \delta \dot{\chi} \dot{\chi} = - \hat{\alpha} g'(\dot{\chi}) + \hat{\beta} \delta \dot{\chi} \dot{\chi} \]  

(93)

Eq. (91) becomes

\[ \delta \dot{\chi} \delta \dot{\chi} = \hat{\alpha} g'(\dot{\chi}) g(\dot{\chi}) - \hat{\beta} \delta \dot{\chi} g(\dot{\chi}) \]  

(94)

We note that, because of (58),

\[ \delta \dot{\chi} g(\dot{\chi}) = 0 \]  

(95)

and, therefore,

\[ \delta \dot{\chi} \delta \dot{\chi} = \hat{\alpha} g'(\dot{\chi}) g(\dot{\chi}) \]  

(96)

From (86), (89), and (96), we conclude that

\[ \beta = \frac{\alpha g'(\dot{\chi}) g(\dot{\chi})}{\hat{\alpha} g'(\dot{\chi}) g(\dot{\chi})} \]  

(97)
which is identical with (10). Thus, for a quadratic function, the independent multiplier algorithm yields values of $\alpha$ and $\theta$ identical with those of the Fletcher-Reeves algorithm.
8. **NUMERICAL EXAMPLE**

In order to compare the present method with the Fletcher-Reeves method and the ordinary gradient method, a numerical example was carried out. The function to be minimized is the following (Ref. 3):

\[
f(x) = 100(x_2 - x_1)^2 + (1 - x_1)^2 + 90(x_4 - x_3)^2 + (1 - x_3)^2 \\
+ 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1)
\]

and exhibits a relative minimum as well as an absolute minimum at the point

\[
x_1 = 1, \ x_2 = 1, \ x_3 = 1, \ x_4 = 1
\]

where

\[
f = 0
\]

The nominal point chosen for starting the descent process is the point

\[
x_1 = -3, \ x_2 = -1, \ x_3 = -3, \ x_4 = -1
\]

where

\[
f = 19,192
\]

The following convergence criterion was adopted for stopping the descent process:

\[
f(x) \leq 10^{-13}
\]
approximately equivalent to an accuracy of $10^{-7}$ in each of the coordinates involved.

Concerning a particular iteration, the search for the optimum values of the multipliers $\alpha$ and $\beta$ was terminated when the changes of $\alpha$ and $\beta$ predicted through Eqs. (73) satisfied the inequalities

$$|\delta\alpha/\alpha| \leq 10^{-6}, \quad |\delta\beta/\beta| \leq 10^{-6}$$

(104)

The first derivatives of the function $F$ with respect to the multipliers $\alpha$ and $\beta$ were computed analytically through Eqs. (67). The second derivatives were computed numerically through Eqs. (69).

Computations were performed in double precision arithmetic using the Rice University Burroughs B-5500 computer. This machine uses a 48 bit word length, resulting in 11 significant figures in single precision and 23 significant figures in double precision. For each iteration, the coordinates $x_1, x_2, x_3, x_4$ of the position vector as well as the value of the function $f$ were determined. Within each iteration, the number of search steps required to converge to the optimum values of $\alpha$ and $\beta$ was recorded.

Particular attention was given to the time element, since time rather than number of iterations is the key indicator of the effectiveness of a given algorithm. The following time components must be considered: (a) the computing time $t_1$, the actual time employed doing numerical calculations in binary form, (b) the conversion time $t_2$, the time employed converting information from binary form to decimal form, and (c) the printout time $t_3$, the time employed printing the information
required. For exploratory purposes, a scientist may be interested in the initial nominal data, the final converged data, as well as all intermediate data; on the other hand, for industrial purposes, an engineer may be uninterested in intermediate data and may wish to print only the initial nominal data and the final converged data. In the first case, the time \( t_2 \) can be of the same order of magnitude as the time \( t_1 \); in the second case, the time \( t_2 \) is negligible with respect to the time \( t_1 \). It should be noted that the time components \( t_2 \) and \( t_3 \) are not uniquely defined: they depend on the desires of the individual investigator. Therefore, the only logical basis for a time comparison is the computing time \( t_1 \), which is tied to the characteristics of the algorithm and not the characteristics of the investigator.

For the present example, the time \( t_1 \) was determined through a readout of the dedicated central processor time in 60ths of a second. For accuracy, special runs were made without printout during computation. Since the B-5500 computer is a time-sharing machine, the time \( t_1 \) is affected by the simultaneous presence of other programs in the computer; to eliminate this effect, these special runs were made without time sharing. Also, multiple runs were made in order to minimize errors in the time \( t_1 \).

Computations were performed with the independent multiplier method, the Fletcher-Reeves method, and the ordinary gradient method. For the first two methods, three variations of the algorithm were considered: (a) no restart, (b) restart every \( k \) iterations, and (c) restart every \( k + 1 \) iterations. If \( m \) denotes the number of iterations performed before restarting, case (a) is characterized by
m = ∞, case (b) by m = k, and case (c) by m = k + 1. In this connection, the number of iterations n necessary to satisfy inequality (103) and the associated computing time $t_1$ are given in Tables 1 and 2. Also, the dependence of the function $f$ on the number of iterations $n$ and the computing time $t_1$ is shown in Figs. 2-7.

### Table 1. Total number of iterations

<table>
<thead>
<tr>
<th></th>
<th>$m = ∞$</th>
<th>$m = k$</th>
<th>$m = k + 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent multiplier method</td>
<td>$n = 34$</td>
<td>$n = 17$</td>
<td>$n = 15$</td>
</tr>
<tr>
<td>Fletcher-Reeves method</td>
<td>$n = 39$</td>
<td>$n = 29$</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2. Computing time

<table>
<thead>
<tr>
<th></th>
<th>$m = ∞$</th>
<th>$m = k$</th>
<th>$m = k + 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent multiplier method</td>
<td>$t_1 = 17.8$ sec</td>
<td>$t_1 = 9.2$ sec</td>
<td>$t_1 = 8.8$ sec</td>
</tr>
<tr>
<td>Fletcher-Reeves method</td>
<td>$t_1 = 14.8$ sec</td>
<td>$t_1 = 11.9$ sec</td>
<td></td>
</tr>
</tbody>
</table>

It should be noted that the ordinary gradient method did not converge to the degree of accuracy (103) even after 100 iterations. On the other hand, the independent multiplier method converged in every case, while the Fletcher-Reeves method converged only for $m = k$ and $m = k + 1$. For those cases where the methods converged, it was found that (a) the number of iterations of the independent multiplier method is 40-50% of that of the Fletcher-Reeves method and (b) the computing time of the
independent multiplier method is 60-75% of that of the Fletcher-Reeves method. The present algorithm requires more computing time for each iteration owing to the fact that a two-dimensional search, rather than a one-dimensional search, is necessary. Despite this, the overall computing time is less than in the Fletcher-Reeves method owing to the considerable decrease in the number of iterations.

8.1. Remark. In order to compute the second derivatives through Eqs. (69), a specification of the value of ε is necessary. If ε is too large, the second derivatives are not accurate because the first derivatives do not behave linearly. If ε is too small, the second derivatives are not accurate because the differences in the first derivatives cannot be computed with sufficient precision. Therefore, ε must be in a proper range. For the test problem under consideration, convergence to $f \leq 10^{-13}$ using the algorithm (52) was achieved for ε in the range $10^{-20} \leq \varepsilon \leq 10^{-2}$. In this range, the number of iterations $n$ and the computing time for convergence $t_1$ were found to be insensitive to changes in ε. At any rate, the numerical data presented here refer to $\varepsilon = 10^{-8}$.

8.2. Remark. The computations presented here were done in double precision arithmetic. Experiments were made by using various combinations of single precision and double precision. In one of the experiments, the position vector $x$ and the gradient $g(x)$ were calculated in double precision while everything else (in particular, the optimum values of $\alpha$ and $\beta$) was calculated in single precision. With this mixed approach, the independent multiplier method converged in the same
number of iterations as with the double precision approach; however, the computing time decreased. With the same mixed approach, the Fletcher-Reeves method converged in a greater number of iterations and greater computing time than with the double precision approach. Clearly, the independent multiplier algorithm has greater flexibility than the Fletcher-Reeves algorithm, since it is less sensitive to round-off errors.
9. **DISCUSSION AND CONCLUSIONS**

In this paper, a new accelerated gradient method for finding the minimum of a function \( f(x) \) is presented, \( f \) being a scalar and \( x \) a \( k \)-vector. The new algorithm, called the independent multiplier method, can be stated as follows:

\[
\tilde{x} = x + \delta x, \quad \delta x = -\alpha g(x) + \beta \hat{x}
\]  

(105)

where \( \alpha \) and \( \beta \) are scalars, chosen at each step so as to yield the greatest decrease in the function \( f \). The above algorithm is a generalization of the Fletcher-Reeves algorithm and its principal objective is to reduce the computing time required for convergence. The following comments are pertinent:

(a) With the independent multiplier algorithm, both \( \alpha \) and \( \beta \) are optimized. With the Fletcher-Reeves algorithm, only \( \alpha \) is optimized since the ratio \( \beta/\alpha \) is kept at a preselected value determined from quadratic considerations.

(b) Because of (a), the independent multiplier algorithm requires a two-dimensional search, while the Fletcher-Reeves algorithm requires a one-dimensional search. The two-dimensional search can be performed by using quasilinearization with built-in safeguards to insure the stability of the descent process.

(c) For a quadratic function, the independent multiplier algorithm yields values of \( \alpha \) and \( \beta \) identical with those of the Fletcher-Reeves algorithm. Therefore, in this case, the number of iterations required for convergence is identical for the two methods. However, owing to the two-dimensional search, the computing time for the independent multiplier method is larger than that for the Fletcher-Reeves method.
(d) For a nonquadratic function, the independent multiplier algorithm exhibits much faster initial convergence than the Fletcher-Reeves algorithm. This is due to the extra degree of freedom available.

(e) The independent multiplier algorithm has definite advantages with respect to the Fletcher-Reeves algorithm in compensating for round-off errors. Again, this is due to the extra degree of freedom available.

(f) A test case was considered, that of a quartic involving four variables. The initial coordinates were such that \( f = 19,192 \). After the first iteration, both methods reduced the function to \( f = 134.4 \). After four iterations, the independent multiplier method reduced the function to \( f = 0.0045 \) while the Fletcher-Reeves method reduced the function to \( f = 31.5 \).

(g) For the above test case, both algorithms were continued until convergence was achieved (\( f \leq 10^{-13} \)). The number of iterations with the independent multiplier method was about 40-50\% that of the Fletcher-Reeves method; the computing time with the independent multiplier method was about 60-75\% that of the Fletcher-Reeves method.
REFERENCES


LIST OF CAPTIONS

Fig. 1  Nomenclature

Fig. 2  The function $f(n)$ for $m = \infty$.

Fig. 3  The function $f(t_1)$ for $m = \infty$.

Fig. 4  The function $f(n)$ for $m = k$.

Fig. 5  The function $f(t_1)$ for $m = k$.

Fig. 6  The function $f(n)$ for $m = k + 1$.

Fig. 7  The function $f(t_1)$ for $m = k + 1$. 
Fig. 1 Nomenclature.
Fig. 2. The function \( f(n) \) for \( m = \infty \).
Fig. 3  The function $f(t_1)$ for $m = \infty$. 
Fig. 4 The function $f(n)$ for $m = k$. 
Fig. 5  The function $f(t_1)$ for $m = k$. 
Fig. 6 The function $f(n)$ for $m = k + 1$. 
Fig. 7 The function $f(t)$ for $m = k + 1$. 

![Graph showing the function $f(t)$ for $m = k + 1$.]