data is provided by the use of zero-shear rate viscosity. The agreement is very satisfactory and provides considerable support for the use of Equation (1) to estimate stabilizing action of surfactants on pseudoplastic falling films.

NOTATION

g = gravitational acceleration, cm/s^2
K = consistency factor, (dyne s^n)/cm^2
n = flow index, dimensionless
Q = liquid mass flow rate per unit width of plate, g/cm s
u = liquid local velocity in the film, cm/s

Greek Letters

Γ = surfactant surface concentration, mole/cm^2
$\dot{\gamma}$ = rate of shear, s^{-1}
δ = film thickness, cm
μ = viscosity, g/cm s
ν = kinematic viscosity, cm^2/s
ρ = liquid density, g/cm^3
σ = surface tension, dyne/cm
τ_{12} = shear stress, dyne/cm^2

Subscripts

p = pseudoplastic
w = water
a = apparent
ef = effective

LITERATURE CITED

Davies, J. T., and R. W. Vose, "On the Damping of Capillary Waves by Surface Films," *Proc. Roy. Soc.*, **A286**, 218 (1965).
Fulford, G. D., "The Flow of Liquids in Thin Films", in *Advances in Chemical Engineering*, Vol. 5, Academic Press, New York (1964).
Levich, V. G., *Physicochemical Hydrodynamics*, p. 380, p. 610, Prentice-Hall, Englewood Cliffs, N. J. (1962).
Skelland, A. H. P., *Non-Newtonian Flow and Heat Transfer*, p. 74, Wiley, New York (1967).
———., and C. L. Caenepeel, "Effects of Surface Active Agents on Mass Transfer During Droplet Formation, Fall, and Coalescence," *AIChE J.*, **18**, 1154 (1972).
Strobel, W. J., and S. Whitaker, "The Effect of Surfactants on the Flow Characteristics of Falling Liquid Films," *AIChE J.*, **15**, 527 (1969).
Tailby, S. R., and S. Portalski, "The Optimum Concentration of Surface Active Agents for the Suppression of Ripples," *Trans. Inst. Chem. Engrs.*, **39**, 328 (1961).
Whitaker, S., "Effect of Surface Active Agents on the Stability of Falling Liquid Films," *Ind. Eng. Chem. Fundamentals*, **3**, 132 (1964).

Manuscript received December 17, 1973; revision received and accepted February 27, 1974.

Quadratically Convergent Techniques in Linearly Constrained Optimization

GREGORY J. HUSEN
and
JAMES M. EAKMAN

Department of Chemical Engineering
University of Nebraska
Lincoln, Nebraska 68508

Methods for the solution of linearly constrained optimization of a nonlinear objective function are presented and compared. The methods of Fletcher-Reeves, Fletcher, and Powell are used to generate search directions in the decision variable space. Generalized Kuhn-Tucker conditions are presented and used to check for a local minimum.

SCOPE

The objective of this work was to develop an approach to the minimization of nonlinear objective functions subject to linear equality and inequality constraints such that any effective unconstrained minimization technique could be employed to determine the search directions. Such a general approach would allow the immediate application of new unconstrained methods to linearly constrained problems. A number of significant chemical engineering problems fit this mathematical pattern, for example, determination of chemical equilibrium with competing reactions by minimization of the free energy subject to conservation of mass constraints.

Previous approaches (Goldfarb and Lapidus, 1968; Wolfe, 1963, 1967a, 1967b) to this problem were specific to certain search techniques and did not make use of generalized Kuhn-Tucker conditions which remove the requirement that independent variables be bounded to the positive region. Abadie (1970) considered the generalized Kuhn-Tucker conditions in his treatment of the general nonlinear programming problem but did not take advantage of the special treatment which can be given linear constraints.

CONCLUSIONS AND SIGNIFICANCE

The objective was accomplished as illustrated by the successful application of the unconstrained methods of Fletcher-Reeves (1964), Fletcher (1970), and Powell (1964) to a series of linearly constrained problems. Application of Powell's method is particularly significant since the user is not required to provide expressions for the derivatives of the objective function as in the other methods. Additionally, the results presented here may be used to improve generalized nonlinear programming techniques which are based upon repetitive linear programming in that the nonlinear nature of the objective

function can be explicitly accounted for. Also, since the generalized Kuhn-Tucker conditions allow complete freedom in the specification of bounds on independent variables, the proposed technique lends itself quite well to the linearization bounds associated with repetitive approaches.

The general nonlinear programming problem is one of minimization (maximization) of an arbitrary nonlinear objective function subject to a set of nonlinear inequality and equality constraints. The chemical engineer is often faced with an important subclass of the general nonlinear programming problem, that is, the minimization of an arbitrary nonlinear objective function subject to linear constraints. Many economic problems take this form. The economic problems facing the chemical and petroleum industry range from determining the optimal research policy for a company, through finding the best solution to a gasoline blending problem, to scheduling problems in underground oil production or design of a refinery.

Goldfarb and Lapidus (1968) developed an algorithm for this subclass of nonlinear programming problems using projection matrices. Wolfe (1963, 1967a, 1967b) introduced the reduced gradient Jacobian technique and applied it to linearly constrained problems. Abadie and Carpentier (1965, 1966, 1969) have extended this approach to the general nonlinear programming problem. Their algorithm is considered by many workers to be the most powerful presently available for solution of the general nonlinear programming problem. However, some of the algorithm's efficiency is lost when applied to linear constraints due to the special treatment that could be given.

This study is an extension of the reduced gradient technique. The unconstrained approaches of Fletcher and Reeves (1964) Fletcher (1970), and Powell (1964) are coupled with the reduced gradient approach to give algorithms combining some of the most powerful logic in unconstrained and constrained optimization. The generalized Kuhn-Tucker conditions are introduced much as Abadie (1970) has presented them and applied here to linearly constrained optimization. They are also found to be applicable in reducing the matrix size in the linear programming (LP) problem where upper bounds are considered.

GENERALIZED KUHN-TUCKER CONDITIONS

In unconstrained optimization a stationary point is defined as the point where the gradient of the objective is zero; if $\mathbf{x}^*$ is a stationary point then $\mathbf{g}(\mathbf{x}^*) = \mathbf{0}$. A necessary condition for an unconstrained minimum is that it be at a stationary point.

Kuhn and Tucker (1951) were the first to define the necessary conditions for a local optimum of a constrained problem. The following is a development of a generalization of the conditions they proposed. Consider the problem:

$$\text{minimize} \quad \{y(\mathbf{x})\} \tag{1}$$

where

$$\mathbf{x}^T = [x_1, x_2, \ldots, x_n]$$

subject to

$$\begin{aligned} f_j(\mathbf{x}) &\leq 0, \quad j = 1, \ldots, m_1 \\ f_j(\mathbf{x}) &\geq 0, \quad j = m_1 + 1, \ldots, m_2 \\ f_j(\mathbf{x}) &= 0, \quad j = m_2 + 1, \ldots, m \end{aligned} \tag{2}$$

and

$$\mathbf{p} \leq \mathbf{x} \leq \mathbf{q}$$

The problem is first transformed to an equality constrained one, thereby increasing the dimensionality of the problem to $nn(= n + m_2)$. The inequality constraints are transformed to equality constraints by the addition of slack variables to constraints 1 to m_1 and subtraction of slack variables from constraints $m_1 + 1$ to m_2. The constraints then take the forms

$$\begin{aligned} f_j(\mathbf{x}) + x_{n+j} &= 0, \quad j = 1, \ldots, m_1 \\ f_j(\mathbf{x}) - x_{n+j} &= 0, \quad j = m_1 + 1, \ldots, m_2 \\ f_j(\mathbf{x}) &= 0, \quad j = m_2 + 1, \ldots, m \end{aligned} \tag{2}$$

where x_j, $j = n + 1, \ldots, nn$ are slack variables which are also subject to bounds in the form of nonnegativity requirements

$$\mathbf{x}_j \geq 0 \quad \text{for} \quad j = n + 1, \ldots, nn$$

Since there are m constraints there must be m state or dependent variables. The state variables must be free to move in either a positive or negative direction in order to keep the solution on the constraints at least over a limited range of the independent or decision variables. From a practical standpoint this eliminates from the state set any variable which is at one of its bounds. Hence, there must be at least as many variables not at a bound as there are constraints. If there are fewer variables not at a bound than there are constraints then the situation is said to be degenerate. A nondegenerate situation will be assumed unless otherwise stated.

After the state variables have been chosen the remaining set of variables is termed the *decision* set. There are $r = (nn - m)$ decision or independent variables. The dimensionality of the problem is now nn. The nn variables are divided into the state and decision sets, $\mathbf{x}^T = [\mathbf{s}^T, \mathbf{d}^T]$. The problem may then be written

$$\text{minimize} \quad \{y\ (\mathbf{s}, \mathbf{d})\}$$

subject to

$$f_j(\mathbf{s}, \mathbf{d}) = 0, \quad j = 1, \ldots, m$$

where

$$p_i \leq x_i \leq q_i, \quad i = 1, \ldots, nn$$

The slack variables do not appear in the objective function; hence, they may be treated as having zero coefficients in the objective function. Any partial derivative of the objective function with respect to a slack variable is zero. The slack variables have zero lower bounds and in practice the upper bounds are set to large numbers.

An expression to determine the manner in which differential changes in the decision variables effect the objective function is obtained using differentials in a manner similar to Wilde and Beightler (1967)

$$\partial y = [\partial y/\partial s_1, \partial y/\partial s_2, \ldots, \partial y/\partial s_m]\ \partial \mathbf{s} + [\partial y/\partial d_1, \partial y/\partial d_2, \ldots, \partial y/\partial d_r]\partial \mathbf{d} \tag{4}$$

or $\partial y = \mathbf{g}_s{}^T \partial \mathbf{s} + \mathbf{g}_d{}^T \partial \mathbf{d}$

Differential representation of the constraint equations provides

$$\mathbf{0} = \begin{bmatrix} \partial f_1/\partial s_1 & \partial f_1/\partial s_2 & \ldots & \partial f_1/\partial s_m \\ \partial f_2/\partial s_1 & \partial f_2/\partial s_2 & \ldots & \partial f_2/\partial s_m \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \partial f_m/\partial s_1 & \partial f_m/\partial s_2 & \ldots & \partial f_m/\partial s_m \end{bmatrix} \begin{bmatrix} \partial s_1 \\ \partial s_2 \\ \cdot \\ \cdot \\ \cdot \\ \partial s_m \end{bmatrix} \tag{5}$$

(*Continued on next page*)

$$
+ \begin{bmatrix} \partial f_1/\partial d_1 & \partial f_1/\partial d_2 & \ldots & \partial f_1/\partial d_r \\ \partial f_2/\partial d_1 & \partial f_2/\partial d_2 & \ldots & \partial f_2/\partial d_r \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \partial f_m/\partial d_1 & \partial f_m/\partial d_2 & \ldots & \partial f_m/\partial d_r \end{bmatrix} \begin{bmatrix} \partial d_1 \\ \partial d_2 \\ \cdot \\ \cdot \\ \cdot \\ \partial d_r \end{bmatrix}
$$

or $\mathbf{0} = \mathbf{J}\partial \mathbf{s} + \mathbf{C}\partial \mathbf{d}$

where $\mathbf{J}$, the Jacobian, is the matrix of partial derivatives of the constraints with respect to the state variables and $\mathbf{C}$, the control matrix, is the matrix of partial derivatives of the constraints with respect to the decision variables. Rearranging Equation (5)

$$\mathbf{J}\partial \mathbf{s} = -\mathbf{C}\partial \mathbf{d}$$

and solving for $\partial \mathbf{s}$ provides

$$\partial \mathbf{s} = -\mathbf{J}^{-1}\mathbf{C}\partial \mathbf{d} \tag{6}$$

Equation (6) is now substituted into Equation (4)

$$\partial y = \mathbf{g}_s^T[-\mathbf{J}^{-1}\mathbf{C}\partial \mathbf{d}] + \mathbf{g}_d^T\partial \mathbf{d} \tag{7}$$

Regrouping terms

$$\partial y = [\mathbf{g}_d^T - \mathbf{g}_s^T\mathbf{J}^{-1}\mathbf{C}]\partial \mathbf{d} \tag{8}$$

The decision derivative vector, the projection of the gradient of the objective function on the constraint surfaces, is the relation between changes in the decision variables and changes in the objective function on the constraint surfaces

$$\boldsymbol{\delta}^T y \equiv [\delta y/\delta d_1, \delta y/\delta d_2, \ldots, \delta y/\delta d_r] = \mathbf{g}_d^T - \mathbf{g}_s^T\mathbf{J}^{-1}\mathbf{C} \tag{9}$$

If $\mathbf{x}^*$ is a local minimum, then small changes in the decision variables will only increase the constrained objective.

$$\partial y^* = [\delta y^*/\delta d_1, \delta y^*/\delta d_2, \ldots, \delta y^*/\delta d_r]\partial \mathbf{d}$$

If a decision variable is at its lower bound ($d_j = p_j$), then only nonnegative changes are allowed ($\partial d_j \geqq 0$). This means that the corresponding decision derivative must be nonnegative:

$$\delta y^*/\delta d_j \geqq 0$$

In the case where an upper bound is active ($d_j = q_j$), only negative or zero changes are allowed ($\partial d_j \leqq 0$). The corresponding decision derivative must be negative or zero.

$$\delta y^*/\delta d_j \leqq 0$$

If a decision variable is not at a bound ($p_j < d_j < g_j$), the differential change may be either positive or negative; hence, the corresponding decision derivative must be zero

$$\delta y^*/\delta d_j = 0$$

A local optimum may be defined by the following necessary conditions:

$$\text{if} \quad d_i^* = p_i, \quad \text{then} \quad \delta y^*/\delta d_i \geqq 0$$
$$\text{if} \quad d_i^* = q_i, \quad \text{then} \quad \delta y^*/\delta d_i \leqq 0$$
$$\text{if} \quad p_i < d_i^* < q_i, \quad \text{then} \quad \delta y^*/\delta d_i = 0$$

These are generalizations of those originally given by Kuhn and Tucker (11).

LINEARLY CONSTRAINED OPTIMIZATION

General Approach

The work reported here deals with a special case of the generalized nonlinear programming problem where all constraints are linear. The treatment will be different from the method of Rosen (1960, 1961) or Goldfarb and Lapidus (1968). The technique reported here resembles the GRG algorithm with, however, special considerations being made due to the advantage of having only linear constraints.

The objective of this study was to use some of the most powerful methods available in unconstrained optimization —Fletcher-Reeves (1964) and Fletcher's (1970) (gradient required) and Powell's (1964) (pattern search)—with the reduced gradient approach of Wolfe (1963, 1967a, 1967b). The convergence criteria established makes use of the generalized Kuhn-Tucker conditions allowing independent variables to move in both their positive and negative regions. This eliminates the need to add constraint equations for upper bounds.

The equality constrained problem, represented by Equation's (3), has all constraints active. The nn variables are separated into m state variables and $r(= nn - m)$ decision variables. There are two restrictions on choices of variables for the state and decision sets. First, the state variables must be free to move in a positive or negative direction (that is, state variables can not be at a bound). Second the Jacobian $\mathbf{J}$ must not be singular. Explicit expressions for the state variables (s) may be obtained by simplexing the constraint equations.

For the jth state variable,

$$s_j = \beta_j - \sum_{i=1}^{r} \alpha_{ji}d_i \tag{11}$$

The original a_{ji}'s and b_j's have been transformed to α_{ji}'s and β_j's during the simplexing operations.

A search direction $\boldsymbol{\xi}$ must be generated in the decision space. Various methods for generating $\boldsymbol{\xi}$ will be discussed later when specific applications of various unconstrained methods are considered. Assume here that a direction $\boldsymbol{\xi}$ has been generated. The elements of $\boldsymbol{\xi}$ would need to be modified if the following combinations exist:

$$d_i = p_i \quad \xi_i < 0$$
$$d_i = q_i \quad \xi_i > 0$$

In these cases any positive step along $\boldsymbol{\xi}$ would violate the bounds on the d_i. In these cases it appears appropriate to set the element, $\xi_i = 0$. With the $\boldsymbol{\xi}$ vector modified it is then appropriate to normalize the direction vector.

$$\boldsymbol{\xi} = \boldsymbol{\xi}/\sqrt{\boldsymbol{\xi}^T\boldsymbol{\xi}} \tag{12}$$

The line of search is thus defined by

$$\mathbf{d} = \mathbf{d}^0 + \boldsymbol{\xi}t \tag{13}$$

where $\mathbf{d}^0$ is the base point and t is the step taken along $\boldsymbol{\xi}$. For any step t taken along $\boldsymbol{\xi}$, the state variables can also be calculated at the new point. By substituting Equation (13) into Equation (11)

$$s_j = \beta_j - \sum_{i=1}^{r} \alpha_{ji}(d_i + \xi_i t) \quad j = i, \ldots, m \tag{14}$$

rearranging

$$s_j = s_j^0 - \sum_{i=1}^{r} \alpha_{ji}\xi_i t = s_j^0 + \zeta_j t \tag{15}$$

where

$$s_j^0 = \beta_j - \sum_{i=1}^{r} \alpha_{ji}d_i^0 \tag{16}$$

and

$$\zeta_j = -\sum_{i-1}^{r} \alpha_{ji}\xi_i \tag{17}$$

This gives a relationship for $\mathbf{s}$ and $\mathbf{d}$ along $\boldsymbol{\xi}$; hence, the maximum possible step before driving a state or decision variable to one of its bounds is determined as follows:

$$\text{if} \quad \zeta_j > 0 \quad \text{define} \quad s_j^+ = q_j$$

$$\text{if} \quad \zeta_j < 0 \quad \text{define} \quad s_j^+ = p_j$$

and

$$\text{if} \quad \zeta_i > 0 \quad \text{define} \quad d_i^+ = q_i$$

$$\text{if} \quad \zeta_i < 0 \quad \text{define} \quad d_i^+ = p_i$$

Recall in the above that the slack variables have no upper bound defined. The maximum allowable step then is given by

$$t_{\max} = \text{minimum positive value} \begin{cases} (s_j^+ - s_j)/\zeta_j, & j = 1, 2, \ldots, m \\ (d_i^+ - d_i^0)/\xi_i, & i = 1, 2, \ldots, r \end{cases}$$

A linear search now can be performed along $\boldsymbol{\xi}$ to determine t^*, the step to the minimum. There now are two possibilities: (1) if t^* is greater than or equal $t_{\max}$ then the minimum along $\boldsymbol{\xi}$ is infeasible and the step $t_{\max}$ is taken or (2) if t^* is less than $t_{\max}$ the step to the minimum t^*, is taken. If the step taken along $\boldsymbol{\xi}$ is $t_{\max}$ there are again two possibilities: (1) if $t_{\max}$ is a step that drives a decision variable to its bound, then a new direction can be determined and the algorithm may proceed, or (2) if $t_{\max}$ is a step that drives a state variable to its bound, then there must be a simplexing or exchanging of a state variable because state variables must be free to move positively or negatively.

Whenever a simplexing is to be performed to exchange the ith state variable with some decision variable, this decision variable is chosen by a method suggested by Abadie (1970). The decision variable d_i is chosen as the one corresponding to the maximum on i of the minimum of

$$\{|\alpha_{ji}|\,(q_i - d_i), \quad |\alpha_{ji}|\,(d_i - p_i)\}$$

where α_{ji} is the element in the jth row and ith column of the constraints coefficient matrix. The use of this test to choose the new state variable from the decision set prevents the Jacobian $\mathbf{J}$ from becoming singular.

The convergence criteria checked in all algorithms developed in this study were the generalized Kuhn-Tucker conditions.

If a first feasible point is not available, then before the nonlinear programming problem can be solved, it must be determined. This is done by the so-called "phase one-phase two" approach through the addition of auxiliary slack variables Δ_j to the constraint equation $j = m_1 + 1$ to m and determination of the solution of the auxiliary phase one linear programming problem:

$$\text{minimize} \left\{ \pi = \sum_{j=m_1+1}^{m} \Delta_j \right\} \tag{18}$$

subject to

$$\begin{aligned} &\sum_{i=1}^{n} a_{ji}x_i + x_{n+j} = b_j, && j = 1, \ldots, m \\ &\sum_{i=1}^{n} a_{ji}x_i - x_{n+j} + \Delta_j = b_j, && j = m_1 + 1, \ldots, m_2 \\ &\sum_{i=1}^{n} a_{ji}x_i + \Delta_j = b_j, && j = m_2 + 1, \ldots, m \end{aligned} \tag{19}$$

and

$$\mathbf{p} \leq \mathbf{x} \leq \mathbf{q}$$

The original problem will have a feasible region if the minimum value of π is zero. The linear programming solution of this auxiliary problem generates a starting point for the solution of the original nonlinear problem.

The linear programming technique used for this step is the same as that used in the nonlinear programming approach except that the maximum step along the direction $\boldsymbol{\xi}$ is always taken. The direction vector $\boldsymbol{\xi}$ is chosen as the steepest descent direction.

Nothing has yet been said concerning the choice of the direction $\boldsymbol{\xi}$ in the nonlinear optimization. Four methods of direction generation were considered. Special comments will be made about the modification of the general approach presented above for each of these methods.

If values for the gradient of the objective function $\mathbf{g}$ were readily available then three methods were considered: (1) steepest descent, (2) Fletcher-Reeves, (3) Fletcher's method. Powell's method was considered when derivatives of the objective function were not available.

APPLICATION OF THE GRADIENT METHOD

Application of the classical gradient method to the constrained problem is straightforward. The decision derivatives are used in the constrained problem in the same way as the partial derivatives in the unconstrained gradient method. Once the gradient is obtained, the decision derivatives are calculated using Equation (9). The direction vector then becomes

$$\boldsymbol{\xi} = -\begin{bmatrix} \delta y/\delta d_1 \\ \delta y/\delta d_2 \\ \cdot \\ \cdot \\ \cdot \\ \delta y/\delta d_r \end{bmatrix} \tag{20}$$

The elements of this vector are modified in the manner indicated above if any of the decision variables are at their bounds. This is the direction to be searched. As could be expected, it was found that the constrained gradient method suffers from the same oscillatory behavior as the unconstrained gradient method (see examples). However, it serves as a good point of reference in considering the other approaches.

APPLICATION OF THE FLETCHER-REEVES METHOD

The Fletcher-Reeves (1964) conjugate direction method can be adapted to this problem by replacing the derivatives of the objective function with the decision derivatives. At the ith iteration the next search direction is generated by

$$\boldsymbol{\xi}_{i+1} = -\boldsymbol{\delta y}_{i+1} + \frac{\boldsymbol{\delta y}_{i+1}^T\,\boldsymbol{\delta y}_{i+1}}{\boldsymbol{\delta y}_i^T\,\boldsymbol{\delta y}_i}\,\boldsymbol{\xi}_i \tag{21}$$

The elements of this vector are modified as indicated in the general case if any of the decision variables are at their bounds. Convergence is accelerated by reinitializing the search direction to a steepest descent step every $r + 2$ iterations; since, for nonquadratic function, this technique tends to generate nearly parallel search directions after $r + 2$ cycles.

The proof of conjugacy for the Fletcher-Reeves method depends on searching to a minimum along $\boldsymbol{\xi}$ at each iteration. When it is not possible to step to a minimum along $\boldsymbol{\xi}$ because the minimum is infeasible, the step $t_{\max}$ is taken and the best apparent action was to reinitialize the search direction to a steepest descent direction.

The Fetcher-Reeves implementation reported here provides the user with the option to specify whether or not the objective function is linear. If the function is linear, the program behaves as a gradient linear programming algorithm. This treatment of the LP problem is interesting because of the manner in which upper bounds are treated without adding rows to the matrix and could have some use in this important application.

APPLICATION OF FLETCHER'S METHOD

Application of Fletcher's (1970) algorithm to this problem calls for a somewhat different modification to the general approach just discussed. The requirement for linear search is partially removed. Previous information about the objective is used to sequentially update the approximation to the inverse of the matrix of second order decision derivatives $\tilde{\mathbf{H}}^{-1}$. As in the gradient and Fletcher-Reeves methods the directions are generated by replacing the derivatives of the objective by the decision derivatives.

The approach is analogous to the unconstrained Fletcher approach. Initially $\tilde{\mathbf{H}}_0^{-1}$ is equated to the identity matrix, making the first direction generated a steepest descent direction. A linear search was performed along this line because experience with the program indicated that its initial application improved the convergence in most cases tested. Providing the minimum along this direction is feasible, the decision derivatives are evaluated. The first update to $\tilde{\mathbf{H}}_i^{-1}$ is made using the following formula which gave the smallest correction to $\tilde{\mathbf{H}}_i^{-1}$.

$$\tilde{\mathbf{H}}_i^{-1} = \tilde{\mathbf{H}}_{i-1}^{-1} + \frac{\Delta\mathbf{d}_i\,\Delta\mathbf{d}_i^T}{\Delta\mathbf{d}_i^T\,\Delta\boldsymbol{\delta}y_i} - \frac{\tilde{\mathbf{H}}_{i-1}^{-1}\,\Delta\boldsymbol{\delta}y_i\,\Delta\boldsymbol{\delta}y_i^T\,\tilde{\mathbf{H}}_{i-1}^{-1}}{\Delta\boldsymbol{\delta}^T y_i\,\tilde{\mathbf{H}}_{i-1}^{-1}\,\Delta\boldsymbol{\delta}y_i} \tag{22}$$

or

$$\tilde{\mathbf{H}}_i^{-1} = \tilde{\mathbf{H}}_{i-1}^{-1} - \frac{\Delta\mathbf{d}_i\,\Delta\boldsymbol{\delta}y_i^T}{\Delta\mathbf{d}_i^T\,\Delta\boldsymbol{\delta}y_i}\tilde{\mathbf{H}}_{i-1}^{-1} - \tilde{\mathbf{H}}_{i-1}^{-1}\frac{\Delta\boldsymbol{\delta}y_i\,\Delta\mathbf{d}_i^T}{\Delta\mathbf{d}_i^T\,\Delta\boldsymbol{\delta}y_i} + \left(1 + \frac{\Delta\boldsymbol{\delta}y_i^T\,\tilde{\mathbf{H}}_{i-1}^{-1}\,\Delta\boldsymbol{\delta}y_i}{\Delta\mathbf{d}_i^T\,\Delta\boldsymbol{\delta}y_i}\right)\frac{\Delta\mathbf{d}_i\,\Delta\mathbf{d}_i^T}{\Delta\mathbf{d}_i^T\,\Delta\boldsymbol{\delta}y_i} \tag{23}$$

where $\tilde{\mathbf{H}}_i^{-1}$ is the ith approximation to the inverse of the Hessian in the decision variable space. After $\tilde{\mathbf{H}}_i^{-1}$ has been updated the next change to occur in the decision variables is calculated from

$$\Delta\mathbf{d}_{i+1} = -\tilde{\mathbf{H}}_i^{-1}\,\boldsymbol{\delta}y_{i+1} \tag{24}$$

This logic may produce too large a step. If $(\Delta\boldsymbol{\delta}y^T)(\Delta\mathbf{d})$ is less than 0.0001 or greater than 0.9999 then the step is assumed to be too large. The next step tried is set to one-tenth of the previous step. This shorter step is then taken provided it is feasible. The step predicted by Fletcher's logic may be infeasible; in this case, the $t_{\max}$ step is taken. This makes it necessary to reinitiate to a steepest descent directions if a state variable and decision variable are exchanged (that is, a simplexing operation has occurred). In the case when a decision variable is driven to a bound the $\tilde{\mathbf{H}}_0^{-1}$ approximation is initiated for the new space.

APPLICATION OF POWELL'S METHOD

The time spent in problem preparation is much more lengthy when derivatives of the objective are required. The use of a pattern or direct search method would be much more satisfactory where derivatives are difficult to obtain, not to mention the elimination of human error in taking the derivatives. Powell's (1964) method is one of the best pattern or direct search techniques available at present. It is based upon the idea of gradual introduction of conjugate directions into a set of search directions. Powell's method starts with n linearly independent directions, usually the coordinate directions, $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_n$, at $\mathbf{d}_0$. A series of linear searches is performed along the search directions. Starting from $\mathbf{d}_0$ a search along $\boldsymbol{\xi}_1$ to a minimum at $\mathbf{d}_1$ is performed, then from $\mathbf{d}_1$ a search along $\boldsymbol{\xi}_2$ to a minimum at $\mathbf{d}_2$ is performed and so on for the remaining directions. After n linear searches the final point is $\mathbf{d}_n$. A new search direction is defined as $\boldsymbol{\mu} = \mathbf{d}_n - \mathbf{d}_0$, but it is not immediately added to the direction set. Powell proposed two tests designed to prevent a potentially poor search direction from being introduced:

1) Test if function increases

$$y_t \geqq y_0$$

where $y_t = y(2\mathbf{d}_n - \mathbf{d}_0)$

$$y_0 = y(\mathbf{d}_0)$$

2) Test if $\boldsymbol{\mu}$ points across a deep valley

$$(y_0 - 2y_n + y_t)(y_0 - y_n - \Delta)^2 \geqq (y_0 - y_t)^2/2$$

where $y_n = y(\mathbf{d}_n)$

and Δ = the greatest function improvement during the last $\mathbf{n}$ linear searches.

If either or both of these tests are true, then $\boldsymbol{\mu}$ is assumed to be a poor direction and not introduced.

If $\boldsymbol{\mu}$ is not introduced, set $\mathbf{d}_0 = \mathbf{d}_n$ and repeat the series of n linear searches. However, if $\boldsymbol{\mu}$ is a good direction, a linear search is performed along $\boldsymbol{\mu}$ to the minimum $\mathbf{d}_{n+1}$. Now $\boldsymbol{\mu}$ is introduced into the direction set as $\boldsymbol{\xi}_n$ and the direction that gave the greatest function improvement during the last n linear searches is removed. Since the search direction is not always added, the method cannot be rigorously proven quadratically convergent.

This method was adapted to the linearly constrained problem using the same simplexing logic to transform the directions to the different decision spaces.

Initially, the direction matrix, the ith row of which is $\boldsymbol{\xi}_i^T$, is equated to the identity matrix giving a univariate search pattern. After each of the r directions in the direction matrix have been searched, providing no variable was driven to a bound, a new search direction is introduced in the constrained space according to the unconstrained logic outlined previously. However, if a bound is encountered then there are two possibilities. First, if the bound encountered is restricting a decision variable then no simplexing is done. But because conjugacy proofs are based upon doing a search to the minimum along each direction in turn the algorithm returns to the first direction vector in the direction matrix and begins to search along all the directions again. Second, if a state variable encounters a bound, then a simplexing operation must occur to exchange the state variable with a decision variable. The simplexing operation performed on the constraint equations can be viewed as solving one of the constraint equations for a particular variable and substituting that relationship into the other constraints for the variable. This eliminates that variable from all constraint relationships but one. An explicit expression thus exists for the variable going from the decision set to the state set. This relationship was also substituted into the direction vectors eliminating the direction elements corresponding to the new state variable and introducing elements for the new decision variable. This corresponds to transforming the original directions into the new decision space.

Since the gradient is not directly available, preliminary convergence criteria must be developed. The algorithm reported here terminates calculation in either of two cases: (1) if the percent change in the objective function and coordinates are below some specified tolerance, or (2) if there have been r linear searches without function improvement. After either of these criteria have been satisfied, the program numerically evaluates the gradient of the objective function at the last point. If the Kuhn-Tucker conditions based upon those numerical derivatives are satisfied it is assumed that a local optimum has been found. If not, the search procedure may be reinitiated.

DISCUSSION OF RESULTS

The four codes which correspond to the application of the gradient, Fletcher-Reeves (1964), Fletcher (1970), and Powell (1964) methods to linearly constrained optimization were tested on the seven problems given in the Appendix. A summary of these results is shown in Table 1.

The convergence criteria were based upon the Kuhn-Tucker conditions. Five-place accuracy was required in these expressions. Analytical derivatives are not required in the application of Powell's method; hence numerical derivatives of the objective function based on central differences are used to calculate the decision derivatives. In this case only three-place accuracy was required for the Kuhn-Tucker conditions. In Powell's application the test for convergence is made only after the fractional change in the objective function and the coordinates is less than 0.00001.

When derivatives of the objective function are available the method using Fletcher's logic was the best of those tested. This method is very efficient when working in the original space because the technique is essentially the unconstrained method introduced by Fletcher. When several simplexing operations are performed, the performance appears to be slower due to an increased number of function evaluations. This is due, in part, to the linear search in the direction of steepest descent which we have required after each simpex. No adequate way was found for estimating an initial step size for the linear search; therefore, extra function evaluations are used in an acceleration procedure when the initial step size is too small. An improvement to the Fletcher's application might be to eliminate all linear searches, but in order to do this some interpolation relationship would be required. Perhaps a cubic interpolation could be used as Fletcher has suggested; however, this was not implemented.

The application of Powell's method ranks next to Fletcher's application; in fact, this method might even be ranked equal to Fletcher's application due to the convenience of not needing the derivative expressions. The time spent in machine computation may often be negligible compared to the user's time spent in problem preparation. Powell's application usually equalled the performance of the Fletcher-Reeves' approach; while, as expected, both methods surpassed the gradient technique.

The Fletcher, Fletcher-Reeves, and Powell approaches solved all seven example problems. The gradient approach terminated early in most cases due to the oscillatory behavior which resulted in an excessive number of functional evaluations. However, on example problems [6] and [7] the gradient method worked very well. In these problems the controlling factor in convergence was apparently the

TABLE 1. SUMMARY OF RESULTS

	Gradient	Fletcher-Reeves	Fletcher	Powell
Problem 1: Time*	0.62	0.54	0.47	0.84
Function eval.	166	54	25	212
Derivative eval.	32	11	8	37†
Simplexes	4	4	4	4
Problem 2: Time*	Minimum	1.04	0.97	0.85
Function eval.	not	571	108	432
Derivative eval.	found	89	79	64†
Simplexes		0	0	0
Problem 3: Time*	Minimum	0.97	0.54	0.84
Function eval.	not	441	34	455
Derivative eval.	found	67	24	66†
Simplexes		1	1	1
Problem 4: Time*	Minimum	0.89	0.84	0.89
Function eval.	not	362	168	452
Derivative eval.	found	54	33	70†
Simplexes		3	3	2
Problem 5: Time*	Minimum	0.52	0.52	0.64
Function eval.	not	158	53	162
Derivative eval.	found	21	17	20†
Simplexes		3	3	3
Problem 6: Time*	0.52	0.47	0.95	0.54
Function eval.	94	143	744	148
Derivative eval.	19	27	59	25†
Simplexes	2	2	2	2
Problem 7: Time*	1.15	1.12	1.39	1.15
Function eval.	41	65	166	96
Derivative eval.	9	13	15	18†
Simplexes	5	5	5	5

* Time refers to computer computation time in seconds on an IBM 360-65 using the FORTRAN G compiler.

† This entry for Powell's search procedure refers to the number of linear searches rather than the number of derivative evaluations.

simplexing operations rather than the nonlinear search procedure. If the problem solution is at a vertex, such as in linear programming, it would appear that little is to be gained by nonlinear interior search algorithms. However, if the solution is in the interior, these are potentially of great advantage. Hence it is useful to have algorithms such as those presented here which can efficiently perform interior search where needed.

Problem [2] was used by Goldfarb and Lapidus (1968) to test the Davidon (1959), Fletcher-Powell (1963) (D.F.P.) conjugate gradient projection method and the gradient projection method. The Fletcher application illustrated previously required 108 function evaluations compared to 142 for the D.F.P. method. The projected gradient exceeded the maximum number of steps before the solution was found. Powell's application exceeded both Fletcher's and D.F.P. approach in the number of function evaluations with 432. Comparing function evaluations on this problem shows Fletcher's application requiring fewer. It should be noted that this problem is not truly constrained since none of the bounds are active.

Problem [7] was also tested by Goldfarb and Lapidus. The nature of this problem is such that the linear terms predominate in the region searched. Hence, the behavior is similar to that expected from a linear programming type problem and the most likely controlling factor is the simplexing operation. The D.F.P. and gradient projections methods both required fewer function evaluations. Fletcher's approach requiring 166 function evaluations compared to 13 function evaluations reported for the D.F.P. method and 29 function evaluations for the gradient projection method. Powell's application performed well on this problem requiring fewer function evaluations than Fletcher's application with 96 function evaluations. A direct comparison with Goldfarb and Lapidus' results can not be made in this case, however, since it was not possible to begin the methods presented here with the same initial point as Goldfarb and Lapidus.

Fletcher's application required a large number of function evaluations (744) to solve problem [6]. This can be explained by examination of the Hessian. The Hessian for this objective function cannot be positive definite in the region of positive $\mathbf{x}$ thus causing poor search directions to be introduced.

Another side advantage of the reduced gradient approach presented above is that the number of constraint equations can be decreased by the application of the generalized Kuhn-Tucker conditions to the linear programming (LP) problem where upper bounds on the variables are considered. The number of rows in the LP matrix are usually increased due to the treatment of the upper bounds as less than constraints. However, the general Kuhn-Tucker conditions allow the bounds to be treated without this increase in number of constraints.

Since the unconstrained minimization methods employed perform within a specific decision variable space, the quadratic nature of their convergence is maintained as long as no simplexing operations are conducted. From the viewpoint of the initial variable space, the techniques demonstrate quadratic convergence on a constraint once the final required simplexing has been conducted.

The failure of the gradient application in problems [2], [3], [4], and [5] may be attributed to the failure of the gradient technique itself rather than the method for handling the constraints proposed here. It is seen that in all these cases the algorithm has progressed through the required number of simplexing steps to the stage where an interior minimum is being sought in the final decision variable space.

One problem encountered with constrained algorithms is that they often get stuck at a bound which is not a local constrained minimum. Although this has not been experienced to date with the methods proposed, there appears to be nothing inherent within the constraint handling procedure that would either cause or prevent this occurrence. If the search direction initially generated after a simplex were poor, progress could cease. However, it should be noted that the Fletcher and Fletcher-Reeves methods as applied here begin with a gradient step after each simplex. Thus, a locally good direction is provided in these cases.

The techniques introduced above for the incorporation of the three unconstrained minimization methods into the linearly constrained problem can be applied to extend almost any unconstrained search procedure to this type of problem. Thus we may now readily take advantage of new procedures as they appear in the literature.

ACKNOWLEDGMENT

This work was supported, in part, by a grant from the University of Nebraska Engineering Research Center. Computation time was provided by the University of Nebraska Computer Network.

NOTATION

a_{ji} = coefficients of x_i in the jth linear constraint equation
b_j = the constant for the jth linear constraint equation
$\mathbf{C}$ = the control matrix, matrix of partial derivatives of the constraint equation with respect to the decision variables
$\mathbf{d}$ = vector of decision variables
$f_j(\mathbf{x})$ = some arbitrary function of $\mathbf{x}$, which is the left side of the constraint equation
$\mathbf{g}_i$ = the gradient of the objective function evaluated at $\mathbf{x}_i$
$\mathbf{H}$ = the Hessian, the matrix of second-order partial derivatives of the objective function
$\tilde{\mathbf{H}}_i^{-1}$ = the ith approximation to the inverse matrix of second-order decision derivatives
$\mathbf{J}$ = the Jacobian matrix, the matrix of partial derivatives of the constraints with respect to the state variables
$\mathbf{p}$ = vector of lower bounds on $\mathbf{x}$
$\mathbf{q}$ = vector of upper bounds on $\mathbf{x}$
$\mathbf{s}$ = vector of state variables
$\mathbf{s}^0$ = the state variable base point
t = the step taken along the direction to define a new set of independent variables
t^* = the step to the minimum along the direction
$\mathbf{x}$ = vector of coordinates
$\mathbf{x}^*$ = coordinates of the minimum
$\mathbf{x}_i$ = the ith vector of coordinates in a series of points $i = 1, 2, \ldots$
y = objective function

Greek Letters

α_{ij} = the i-row and j column element of the simplexed linear constraint coefficient matrix
β_j = the simplex constant of the jth linear constraint
$\boldsymbol{\delta} y$ = the vector of decision derivatives
$\boldsymbol{\delta}^* y$ = vector of decision derivatives at the minimum
$\boldsymbol{\Delta}$ = the vector of auxiliary slack variables
π = auxiliary objective function
$\boldsymbol{\xi}$ = direction along which the decision variables change
$\boldsymbol{\zeta}$ = direction along which the state variables change
* superscript indicates local optimum
$\sim$ tilda over variables indicates it is evaluated in decision variables space

LITERATURE CITED

Abadie, J., "Application of the GRG Algorithm to Optimal Control Problems," in *Integer and Nonlinear Programming*, p. 189, North-Holland, Amsterdam (1970).

———., and J. Carpentier, "Generalisation de la Methode du Gradient reduit de Wolfe au cas des Contraintes Nonlineaires," EDF note HR 6678 (1965).

——————., "Generalisation de la methode du Gradient de Wolfe au cas Contraintes Nonlineaires," DiB Hertz and J. Melese, (eds.), *Proc. IFORS Conf.*, p. 1041, Wiley, New York (1966).

——————., "Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints, *Optimization*, R. Fletcher, (ed.), p. 37 Academic Press, New York (1969).

Colville, A. R., "A Comparative Study of Nonlinear Programming Codes," IBM N. Y. Scientific Center Report No. 320-2949 (1968).

Davidon, W. C., "Variable Metric Method for Minimization," A. E. C. Research & Development Report ANL-5990, Rev. (1959).

Fletcher, R., "A New Approach to Variable Metric Algorithms," *Computer J.*, 13, 317 (1970).

———., and M. J. D. Powell, "A Rapidly Convergent Descent Method for Minimization," *ibid.*, **6**, 163 (1963).

Fletcher, R., and C. M. Reeves, "Function Minimization by Conjugate Gradients," *ibid.*, **7**, 149 (1964).

Goldfarb, D., and L. Lapidus, "Conjugate Gradient Method for Nonlinear Programming Problems with Linear Constraints," *Ind. Eng. Chem. Fundamentals*, 7, 142 (1968).

Kuhn, H. W., and A. W. Tucker, "Nonlinear Programming," in J. Neyman, (ed.), *Proc. Second Berkeley Symp. Math. Statistics and Probability*, Univ. California Press (1951).

Powell, M. J. D., "An Efficient Method for Finding the Minimum of a Function of Several Variables Without Calculating Derivatives," *Computer J.*, 7, 155 (1964).

Rosen, J. B., "The Gradient Projection Method for Nonlinear Programming, Part I. Linear Constraints," *J. Soc. Indust. Appl. Math.*, 8, 181 (1960). "Part II. Nonlinear Constraints," *J. Soc. Indust. Appl. Math.*, 9, 514 (1961).

Rosenbrock, H. H., "An Automatic Method for Finding the Greatest or Least Value of a Function," *Computer J.*, 3, 175 (1960).

Wilde, Douglas J., and Charles S. Beightler, *Foundations of Optimization*, Prentice-Hall, Englewood Cliffs, N. J. (1967).

Wolfe, P., "Methods of Nonlinear Programming," in R. L. Graves and P. Wolfe, (eds.), *Recent Advances in Mathematical Programming*, p. 73, McGraw-Hill, New York (1963).

———., "Methods for Linear Constraints," in J. Abadie, (ed.), *Nonlinear Programming*, North-Holland, Amsterdam p. 120 (1967a).

———., "The Product Form of the Simplex Method," in J. Abadie, (ed.), *Nonlinear Programming*," p. 301, North-Holland, Amsterdam (1967b).

APPENDIX A. EXAMPLE PROBLEMS

Problem 1

Source: This work
No. of independent variables: 3
No. of constraints: 3 linear inequality constraints
6 bounds on independent variables
Objective function:

$$\text{Minimize} \quad (y(\mathbf{x}) = x_1^2 + x_1x_2 + 2x_1x_3 + 2x_2^2 + x_2x_3 + 3x_3^2 - 15x_1 - 18x_2 - 27x_3)$$

Constraints:

$$6x_1 + 3x_2 + 4x_3 \leq 24$$
$$4x_1 + 8x_2 + 3x_3 \leq 24$$
$$2x_1 + x_2 + x_3 \geq 2$$
$$0 \leq x_i \leq 99999999, \quad i = 1, 2, 3,$$

Feasible starting point:

$$x_1^0 = 1.0, \quad x_2^0 = x_3^0 = 0$$
$$y(x^0) = -14$$

Results:

	Gradient	Fletcher-Reeves	Fletcher	Powell
$y(\mathbf{x})$	−78.0708	−78.0708	−78.0708	−78.0708
x_1	1.2189	1.2189	1.2189	1.2189
x_2	1.1494	1.1494	1.1494	1.1494
x_3	3.3095	3.3095	3.3095	3.3095
No. of function evaluations	166	54	25	172
No. of derivative evaluations	32	11	8	0
Time, s*	0.62	0.54	0.47	0.84

* Time refers to computation time.

Problem 2

Source: Colville, A. R., (1968)
No. of independent variables: 4
No. of constraints: 2 inequality constraints
8 bounds on the independent variables
Objective function:

$$\text{Minimize} \quad (y(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3^2) + 10.1((x_2 - x_1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1))$$

Constraints:

$$6x_1 + 3x_2 + 4x_3 + x_4 \leq 24$$
$$4x_1 + 8x_2 + 3x_3 + x_4 \leq 24$$
$$-10 \leq x_i \leq 10, \quad i = 1, 2, 3, 4$$

Feasible starting point:

$$x_1^0 = x_3^0 = -3; \quad x_2^0 = x_4 = -1$$
$$y(\mathbf{x}^0) = 19192$$

Results:

	Gradient	Fletcher-Reeves	Fletcher	Powell
$y(\mathbf{x})$	0.0562	0.0000	0.0000	0.0000
x_1	1.1171	1.0000	1.0000	0.9999
x_2	1.2490	1.0000	1.0000	0.9999
x_3	0.8673	1.0000	1.0000	1.0000
x_4	0.7517	1.0000	1.0000	1.0000
No. of function evaluations	1000	571	108	432
No. of derivative evaluations	188	89	79	0
Time, s	1.62	1.04	0.97	0.85

Problem 3

Source: This work
No. of independent variables: 4
No. of constraints: 3 linear inequality constraints
8 bounds on the independent variables
Objective function:

$$\text{Minimize} \quad (y(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1))$$

Constraints:

$$6x_1 + 3x_2 + 4x_3 + x_4 \leq 24$$
$$4x_1 + 8x_2 + 3x_3 + x_4 \leq 24$$
$$2x_1 + x_2 + x_3 + x_4 \geq 2$$
$$-10 \leq x_i \leq 10, \quad i = 1, 2, 2, 4$$

First feasible point:

$$x_1^0 = 0.5714, \quad x_i^0 = 0.2857, \quad i = 2, 3, 4$$
$$y(\mathbf{x}^0) = 25.0185$$

Results:

	Gradient	Fletcher-Reeves	Fletcher	Powell
$y(\mathbf{x})$	0.8936	0.0000	0.0000	0.0000
x_1	0.4838	1.0000	1.0000	1.0012
x_2	0.2464	1.0000	1.0000	1.0024
x_3	1.2473	1.0000	1.0000	0.9987
x_4	1.5593	1.0000	1.0000	0.9975
No. of function evaluations	1004	441	34	455
No. of derivative evaluations	180	67	24	
Time, s	1.62	0.97	0.54	0.84

Problem 4

Source: This work
No. of independent variables: 4
No. of constraints: 3 linear inequality constraints
8 bounds on the independent variables
Objective function:

$$\text{Minimize} \quad (y(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (- x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1))$$

Constraints:

$$6x_1 + 3x_2 + 4x_3 + x_4 \leq 14$$
$$4x_1 + 8x_2 + 3x_3 + x_4 \leq 16$$

$$2x_1 + x_2 + x_3 + x_4 \geq 2$$

$$-10 \leq x_i \leq 10, \quad i = 1, 2, 3, 4$$

Feasible starting point:

$$x_1^0 = 0.5714; \quad x_i^0 = 0.2857 \quad i = 2, 3, 4$$

$$y(\mathbf{x}^0) = 25.0185$$

Results:

	Gradient	Fletcher-Reeves	Fletcher	Powell
$y(\mathbf{x})$	0.9248	0.0000	0.0000	0.0000
x_1	0.4839	1.0000	1.0000	0.9975
x_2	0.2456	1.0000	1.0000	0.9951
x_3	1.2415	1.0000	1.0000	1.0020
x_4	1.5538	1.0000	1.0000	1.0041
No. of function evaluations	1004	362	168	452
No. of derivative evaluations	179	54	33	
Time, s	1.75	0.89	0.84	0.89

Problem 5

Source: This work
No. of independent variables: 2
No. of constraints: 3 linear inequality constraints
4 bounds on the independent variables
Objective function:

Minimize $(y(\mathbf{x}) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2)$

Constraints:

$$6x_1 + 3x_2 \leq 9$$

$$4x_1 + 8x_2 \leq 12$$

$$2x_1 + x_2 \geq 2$$

$$-10 \leq x_1 \leq 10, \quad i = 1, 2,$$

Feasible starting point:

$$x_1^0 = 8; \quad x_2^0 = 4$$

$$y(\mathbf{x}^0) = 360{,}049$$

Results:

	Gradient	Fletcher-Reeves	Fletcher	Powell
$y(\mathbf{x})$	0.0717	0.0000	0.0000	0.0000
x_1	0.7323	1.0000	1.0000	1.0000
x_2	0.5354	1.0000	1.0000	1.0000
No. of function evaluations	12	158	53	162
No. of derivative evaluations	2	21	17	
Time, s	0.47	0.52	0.52	0.64

Problem 6

Source: Rosenbrock, H. H., (1960)
No. of independent variables: 3
No. of constraints: 2 linear inequality constraints
6 bounds on independent variables
Objective function:

Minimize $y(\mathbf{x}) = -x_1x_2x_3$

Constraints:

$$0 \leq x_1 + 2x_2 + 2x_3 \leq 72$$

$$0 \leq x_i \;\; 42, \quad i = 1, 2, 3,$$

Feasible starting point:

$$x_i^0 = 10, \quad i = 1, 2, 3,$$

$$y(\mathbf{x}^0) = 1000$$

Results:

	Gradient	Fletcher-Reeves	Fletcher	Powell
$y(\mathbf{x})$	−3456.0	−3456.0	−3456.0	−3456.0
x_1	24.00	24.00	24.00	24.00
x_2	12.00	12.00	12.00	12.00
x_3	12.00	12.00	12.00	12.00
No. of function evaluations	94	143	744	148
No. of derivative evaluations	19	27	59	
Time, s	0.52	0.47	0.95	0.54

Problem 7

Source: Colville, A. R., (1968)
No. of independent variables: 5
No. of constraints: 10 linear inequality constraints
5 bounds on independent variables
Objective function:
Minimize

$$y(\mathbf{x}) = \sum_{j=1}^{5} e_jx_j + \sum_{i=1}^{5}\sum_{j=1}^{5} c_{ij}x_ix_j + \sum_{j=1}^{5} d_jx_j^3$$

Constraints:

$$\sum_{j=1}^{5} a_{ij}x_j - b_i \geq 0 \quad i = 1, \ldots, 10$$

$$x_j \geq 0, \quad j = 1, \ldots, 5$$

Feasible starting point:

$$x_1^0 = 0.0, \quad x_2^0 = 0.06538, \quad x_3^0 = 0.14359$$

$$x_4 = 0.22179, \quad x_5 = 0.71025$$

$$y(\mathbf{x}^0) = 0.000$$

Results:

	Gradient	Fletcher Reeves	Fletcher	Powell
$y(\mathbf{x})$	−32.3487	−32.3487	−32.3487	−32.3486
x_1	0.3000	0.3000	0.3000	0.3000
x_2	0.3335	0.3335	0.3335	0.3335
x_3	0.4000	0.4000	0.4000	0.4000
x_4	0.4283	0.4283	0.4283	0.4283
x_5	0.2240	0.2240	0.2240	0.2240
No. of function evaluations	41	65	166	96
No. of derivative evaluations	9	13	15	
Time, s	1.15	1.12	1.39	1.15

Data for Test Problem 7

j	1	2	3	4	5
e_j	−15	−27	−36	−18	−12
c_{1j}	30	−20	−10	32	−10
c_{2j}	−20	39	−6	−31	32
c_{3j}	−10	−6	10	−6	−10
c_{4j}	32	−32	−6	39	−20
c_{5j}	−10	32	−10	−20	30
d_j	4	8	10	6	2
a_{1j}	−16	2	0	1	0
a_{2j}	0	−2	0	4	2
a_{3j}	−3.5	0	2	0	0
a_{4j}	0	−2	0	−4	−1
a_{5j}	0	−9	−2	1	−2.8
a_{6j}	2	0	−4	0	0
a_{7j}	−1	−1	−1	−1	−1
a_{8j}	−1	−2	−3	−2	−1
a_{9j}	1	2	3	4	5
a_{10j}	1	1	1	1	1

b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}
−40	−2	−.25	−4	−4	−1	−40	−60	5	1

Manuscript received April 26, 1973; revision received February 20 and accepted February 22, 1974.