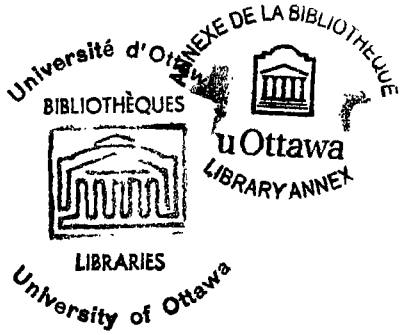
CUADRO: AN INTERACTIVE PACKAGE TO SOLVE THE QUADRATIC PROGRAMMING PROBLEM

by

Oscar Manrique Andrade

A thesis submitted to the School of Graduate Studies and Research
of the University of Ottawa
in partial fulfillment of the requirements
of Master of Science in Systems Science

Ottawa, Ontario, 1982



I hereby declare that I am the sole author of this thesis.

I authorize University of Ottawa to lend this thesis to other institutions or individuals for the purpose of scholarly research.

Oscar Manrique Andrade

I further authorize University of Ottawa to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Oscar Manrique Andrade

University of Ottawa requires the signatures of all persons using or photocopying this thesis. Please sign below, and give address and date.

ABSTRACT

The quadratic programming problem (QPP) deals with the optimization of a quadratic objective function, subject to linear constraints. In general, these constraints can be either in equality or inequality form and the nonnegativity conditions on the variables can apply for all or part of them. However, this problem has to be standardized as a preliminary step before trying to get a solution.

Some methods of solution are briefly presented and their computational behaviour analyzed. The method of Beale is widely discussed in both theoretical and practical ways and is adopted as the procedure to find the optimum. An interactive software package written in FORTRAN is presented.

ACKNOWLEDGEMENTS

I would like to express sincere gratitude to my supervisor, Professor Louis G. Birta, for suggesting the topic of this thesis, for his constant advice, valuable suggestions and guidance. He always found time for productive discussions and encouragement to my work.

INTRODUCTION

Mathematical programming is the name given to the analysis of problems where the optimum of a function is to be found, and the variables are subject to inequality or equality constraints. The term "linear programming", corresponds to the case where the function to be optimized, i.e. the objective function, is linear and where the constraint set is the intersection of a finite number of half spaces. Nonlinear programming is, then, the case where the objective function is nonlinear.

An important case of nonlinear programming is the quadratic programming problem (QPP), where the objective function is a quadratic function and the constraints set is as in the linear case, a finite set of half spaces.

Research on nonlinear programming has gone almost parallel to the development of linear programming, but the nonlinear case has found some mathematical difficulties notably more important that have hinded its development. Thus, we still cannot speak today of a complete theory of nonlinear programming. In this development, the period of the last two decades has, thus far, appeared to be a time of appraisal and consolidation. Generally speaking, the major achievements in this time have been related to evaluating and improving the efficiency of existing solution methods,

The rates at which many of the most commonly used algorithms converge to optimal solutions have been subjected to both theoretical and empirical examination [8,25,29]

Recently, particular attention has been paid to "Large-scale Mathematical Programming", that is, to the difficulties encountered in attempting to solve the very large nonlinear problems that sometimes arise in Operations Research. In this field, important works have been done [16] in the developing of methods based on solving a sequence of QPP as a route towards the solution of a general nonlinear programming problem.

This thesis begins with a brief presentation of some of the most commonly used algorithms for solving the QPP and analyses their computational behaviour. The main intent of this thesis, however, is to present an interactive software package for the QPP, and discuss the most important aspects and problems encountered during its development.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter I

THE QUADRATIC PROGRAMMING PROBLEM

1.1 HISTORICAL OUTLINE

The subject of mathematical programming has formally existed as a well defined subject area, since the late 1940's when George Dantzig developed the simplex method for linear programming. Later, in 1951 H.W.Kuhn and A.W.Tucker, in the Second Berkeley Symposium on Mathematical Statistics and Probability, published the paper "Nonlinear Programming" [18] which established necessary conditions for optimal solutions to nonlinear programming problems. Since that time, the history of Mathematical Programming can be divided, roughly speaking, into two parallel and occasionally convergent streams of development : one inspired by the Kuhn-Tucker optimality criterion and the other based on the principles of Dantzig's Simplex Method.

Initially, the development of Dantzig's Simplex Method was rather narrowly confined to the identification and study of various important subclasses of linear programming problems (Transportation, Network flows, Game theory, etc). Then in the late 1950's, a confluence of the Simplex

computational approach and the Kuhn-Tucker theory produced a number of different algorithms for solving quadratic programming problems. Basically, these methods rely on the fact that the first partial derivative of a quadratic function is linear : this is precisely what makes quadratic programs the most easily handled of all nonlinear problems.

Perhaps the most important period in the development of the solution methods of quadratic problems and, in general, of nonlinear ones was between 1958-1967. Some important works appeared presenting useful and efficient algorithms that can be applied to many nonlinear programs [1,11,30]. Some were applicable to problems having linear constraints only and still others were designed specifically for certain categories of the nonlinear programming problem... the quadratic programming problem is in this class.

1.2 STATEMENT OF THE PROBLEM

Generally speaking, the problem of optimizing a function of one or more variables is called a mathematical programming problem. In considering the nonlinear programming problem one always prefers to think of a minimization problem. This implies no loss of generality

since if the objective is to maximize $F(x)$, this can be converted to an equivalent minimization problem by letting $f(x) = -F(x)$ and then minimizing $f(x)$.

Consider a convex quadratic function of n variables which is to be minimized :

$$G(x) = (1/2)\, x'Ax + b'x + c$$

where :

x is an n-vector of the unknowns

A is a symmetric positive semidefinite matrix with n rows and n columns

b is an n-vector

c is a scalar

The m constraints ($m \leq n$) associated with the problem, are

$$Px - q \leq 0$$

$$x \geq 0$$

where :

P is a matrix with m rows and n columns

q is an m-vector with strictly positive components

Thus the QPP is characterized by linear constraints, nonnegativity requirement on the variables and an objective function $G(x)$ which is a positive semidefinite quadratic form.

The reason why the matrix A must be positive semidefinite arises from one of the most serious problems in nonlinear programming: the problem of local and global minima. If

is a positive semidefinite matrix, any local minimum is also a global minimum. When A is not positive semidefinite, then there is a possibility that several local minima exist.

The QPP is often written as

$$\min \{ G(x) = (1/2)x'Ax + b'x + c \mid Px \leq q , x \geq 0 \} \quad (P1)$$

Note that the constraints and nonnegativity conditions are the same as in the case of linear programming. Constraints which are not in this form must be first transformed to this standard form by using any of techniques of linear programming; i.e. the phase I-phase II procedure or the big M technique (by adding artificial variables) [7].

In the statement of the QPP there are no restrictions on the choice of the elements of P, q or b; but the restriction that A be positive semidefinite is crucial. So far, there is no method for solving the QPP where A is an arbitrary symmetric matrix. As noted in the subsequent discussions, some methods even require that A be positive definite.

1.3 SOME APPLICATIONS

Although linear functions are the most widely used type in the formulation of mathematical optimization problems, quadratic functions are quite firmly established in second

place. This is due to the fact that a large number of the functional relationships occurring in the real world are either truly quadratic or may be so approximated.

For example, the area of a disk, cube or other regular figures is proportional to the square of their characteristic linear dimensions. The kinetic energy carried by a rocket or atomic particle is proportional to the square of its velocity, while the potential energy of a rigid standing wall or dam is a quadratic function of its height. The revenue of a monopolistic firm that sells x_1 units of some product at a unit price of x_2 is $x_1 x_2$, which is also quadratic. In statistics, the variance of a given sample of observations is a quadratic function of the values that constitute the sample.

One very common example of a QPP arises when data are to be fitted to a mathematical model by the least squares method. Suppose an economist theorizes that the fraction B_j of the American public's total consumption expenditure that is allocated to goods and services in the j^{th} class ($j = 1, 2, \ldots, n$) is constant from year to year, regardless of the overall level of consumption (assume all types of goods and services have been partitioned into n mutually exclusive classes). He has available as data a detailed breakdown of all consumption expeditures over an m-year period and wishes to use these statistics to form least squares estimates of

the unknown parameters B_j. Let C_i be the total consumption expenditure and let X_{ij} be the amount allocated for goods and services in the j^{th} class ($j = 1, 2, \ldots, n$) during the i^{th} year ($i = 1, 2, \ldots, m$)

$$C_i = \sum_{j=1}^{n} X_{ij}$$

$$i = 1, 2, \ldots, m$$

$$j = 1, 2, \ldots, n$$

Then the problem can be formulated as

minimize Z

where

$$Z = \sum_{i=1}^{m} \sum_{j=1}^{n} (X_{ij} - B_j C_i)^2$$

$$i = 1, 2, \ldots, m$$

$$j = 1, 2, \ldots, n$$

subject to $\sum_{j=1}^{n} B_j = 1$ and $0 \leq B_j < 1$

Another quadratic program known as the Portfolio problem arises in the planning of investments. Suppose one wants to invest a total of \$B in n different stocks and bonds x_j ($j = 1, 2, \ldots, n$). The annual return on a \$1-investment in the j^{th} security is a random variable whose expected value and variance (based on historical data) are P_j and S_{jj}; the covariance of the i^{th} and j^{th} returns is $S_{ij} = S_{ji}$.

If one wants only to maximize the expected overall return, regardless of risk, it would only be necessary to invest in the security offering the greatest P_j. However, one might prefer to invest in such a way as to minimize the

overall variance of the portfolio subject to the condition of an expected total return of at least $P per year. If we define S to be a symmetric N by N matrix with elements s the problem can be formulated as follows :

find the values of $x_1, x_2, \ldots, x_n$ that minimize Z where

$$Z = x'Sx$$

subject to

$$\sum_{j=1}^{n} R_j x_j \geq P$$

$$\sum_{j=1}^{n} x_j = B \quad \text{and} \quad x_j \geq 0$$

$$j = 1, 2, \ldots, n$$

Chapter II

MATHEMATICAL BACKGROUND

2.1 GLOBAL EXTREMA

A function f(x) takes its absolute minimum at a point x* if f(x) ≥ f(x*) for all x over which the function f(x) is defined. The absolute minimum is also called global minimum.

A function f(x) has a relative minimum at some point x° if there exists an interval including x°, no matter how small, such that for all x in this interval, f(x) ≥ f(x°). The relative minimum is also referred to as local minimum. Similarly, there exist global maxima and local maxima, whose definitions follow logically from the previous ones.

For a concave function, linear interpolation between any two points never overestimates the value of the function. Obviously, if f(x) is concave, then -f(x) is convex.

Let f(x) be a convex function over a closed interval a≤x≤b, then any local minimum of f(x) in this interval is also the global minimum of f(x) over the interval. The global maximum of a convex function f(x) over a closed

interval $a \leq x \leq b$ will be taken on at either $x = a$ or $x = b$ or both. The two previous statements are often stated as theorems and their proofs can be found in the optimization literature [9,10]. We may well be interested in minima that occur when one or more of the variables are at their upper or lower bound. It is readily seen that there are 2^n possible combinations of the a_i and b_i which the x_i may take. If it is to minimize $z = f(x)$ and there is no characteristic known of $f(x)$ (even if it is known that it is continuous and differentiable), it can only be hoped to find a local minimum. However, considerably more can be said if the function being maximized or minimized is concave or convex.

If $f(x)$ is a convex function, it has a unique minimum over a feasible convex region and the same minimum value is found for a convex subset of the feasible region. Furthermore, the minimum of a concave function over a convex region occurs at an extreme point of that region, or otherwise points which are not extreme points can be written as linear combinations of extremal points.

The problem (P1) is said to be a convex quadratic programming problem if A is positive semidefinite, so that $G(x)$ is convex. If A is negative semidefinite, so that $G(x)$ is concave, then problem (P1) is called a concave quadratic programming problem.

2.2 THE SIMPLEX METHOD

2.2.1 The canonical form

When dealing with linear programming problems, the simplex method provides the most efficient procedure for solution. To apply the simplex method, the problem is assumed in its canonical form

$$\begin{aligned} \text{Max } z &= b'x \\ Px &= q \qquad (P2) \\ x &\geq 0 \end{aligned}$$

with n unknowns and m constraints, then :

b is an n-vector,

P is an mxn matrix,

q is an m-vector and

x is the n-vector of the unknowns.

This formulation is clearly very restrictive; however linear programming applies equally to situations where the constraints are inequalities or a mixture of equalities and inequalities , where some of the variables can have negative values and where the objective function is to be maximized.

In general, the case may be

$$\text{maximize } Z$$

where

$$Z = d'y$$

Subject to

$$Fy \leq r$$

$$Gy \geq s$$

$$Hy = t$$

$$y_i \geq 0 \quad i = 1, 2, \ldots k \; ; \; k \leq n$$

which need to be transformed into canonical form.

If a constraint is a less-than condition, then an equation can be made out of this constraint by adding a nonnegative variable u to the left-hand side and writing :

$$Fy + u = r$$

u is called a slack variable and is an additional unknown that has to be determined. Similarly, if a constraint is a greather-than condition, it can be written as

$$Gy - v = s$$

where v again is a nonnegative variable (usually called a surplus variable) that constitutes an additional unknown of the problem. If some of the variables are not constrained to be nonnegative, then these variables can be expressed as

$$y_j = y_j^1 - y_j^2$$

$$y_j^1 \geq 0 \quad y_j^2 \geq 0$$

$$j = 1, 2, \ldots n-k$$

because any number can always be expressed as the difference of two nonnegative numbers. Finally, instead of maximixing Z, -Z will be minimized .

The problem becomes

minimize Z

where

$$Z = \begin{bmatrix} d & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ u \\ v \end{bmatrix}$$

subject to

$$\begin{bmatrix} F & I & 0 \\ G & 0 & -I \\ H & 0 & 0 \end{bmatrix} \begin{bmatrix} y \\ u \\ v \end{bmatrix} = \begin{bmatrix} r \\ s \\ t \end{bmatrix}$$

By putting

$$x = \begin{bmatrix} y \\ u \\ v \end{bmatrix} \qquad P = \begin{bmatrix} F & I & 0 \\ G & 0 & -I \\ H & 0 & 0 \end{bmatrix} \qquad q = \begin{bmatrix} r \\ s \\ t \end{bmatrix} \qquad b = \begin{bmatrix} -d \\ 0 \\ 0 \end{bmatrix}$$

problem (P2) is gotten, which is in canonical form.

It should be noted that the number of unknowns have been increased by u + v + (n-k). Thus any method that solves the canonical form will also handle more general situations involving mixtures of equalities and inequalities, partial nonnegativity constraints, and minimization or maximization. The canonical form of the problem will be referred to hereafter.

2.2.2 Basic solutions

It will be convenient to define some concepts that can be useful in discussing future topics.

1. Feasible solution : Any solution to $Px = q$, $x \geq 0$
2. Basic solution : A solution of $Px = q$, obtained by setting n-m variables equal to zero and solving the remaining m variables, provided that the determinant of the coefficients of these m variables does not vanish (so that the values of these m variables are uniquely determined).
3. Basis : The collection of m variables which are not equal to zero in the construction of a basic solution.
4. Basic feasible solution : A basic solution of $Px = q$ which also satisfies $x \geq 0$.
5. Optimal solution : A feasible solution which satisfies $b'x = \min$.
6. Optimal basic solution : A basic feasible solution which satisfies $b'x = \min$.

Before starting the simplex procedure, a basic feasible solution must be found. For this, the vector q in the constraints $Px = q$ must be strictly positive. Whenever necessary, a new transformation has to be done to the system such that one decision variable be isolated in each constraint with a +1 coefficient, that that variable not

appear in any other constraint and that it have a zero coefficient in the objective function.

The variables added at this point are called artificial variables because they don't belong to the system and must eventually be suppressed. They are introduced only because they constitute inmediately a basic feasible solution. After creating the artificial variables, they have to vanish. For this, the infeasibility form w defined by

$$x_{n+1} + x_{n+2} + \ldots + x_{n+m} = w$$

has to be driven to zero [7,13]. Only then have we a basic feasible solution to start the simplex procedure.

2.2.3 Computational criteria

1. Optimality criterion : Suppose that in a minimization problem every nonbasic variable has a nonnegative coefficient in the objective function of a canonical form. Then the basic feasible solution given by that canonical form minimizes the objective function over the feasible region.
2. Unboundedness criterion : Suppose that in a minimization problem some nonbasic variable has a negative coefficient in the objective function of a canonical form. If that variable has positive or zero coefficients in all constraints, then the objective

function is unbounded from below over the feasible region.

3. Improvement criterion : Suppose that in a minimization problem some nonbasic variable has a negative coefficient in the objective function of a canonical form. If that variable has a negative coefficient in some constraint, then a new basic feasible solution may be obtained by pivoting.

4. Ratio and pivoting criterion : When improving a given canonical form by introducing the variable x_s into the basis, pivot in a constraint that gives the minimum ratio of the right-hand-side coefficient to the corresponding x_s coefficient. Compute these ratios only for constraints that have a positive coefficient of x_s.

When introducing the variable x_s into the basis, another basic variable must leave its place to x_s. This substitution (pivoting)[1] is merely the familiar variable elimination technique from high-school algebra, known more formally as Gauss-Jordan elimination. Consequently, after pivoting, the form of the problem has been altered, but the modified equations still represent the original problem and have the same feasible solutions and the same objective value when

---

[1] The commonly known procedure for pivoting is not treated in this paper. However, it can be found in any book dealing with linear programming [7,13].

evaluated at any given feasible point.

2.2.4 Geometrical interpretation

Consider the following problem stated in its canonical form :

$$\text{minimize } Z$$

where

$$Z = 0x_1 + 0x_2 + 3x_3 + x_4 - 20$$

$$\text{subject to} \quad x_1 - 3x_3 + 3x_4 = 6 \qquad (1)$$

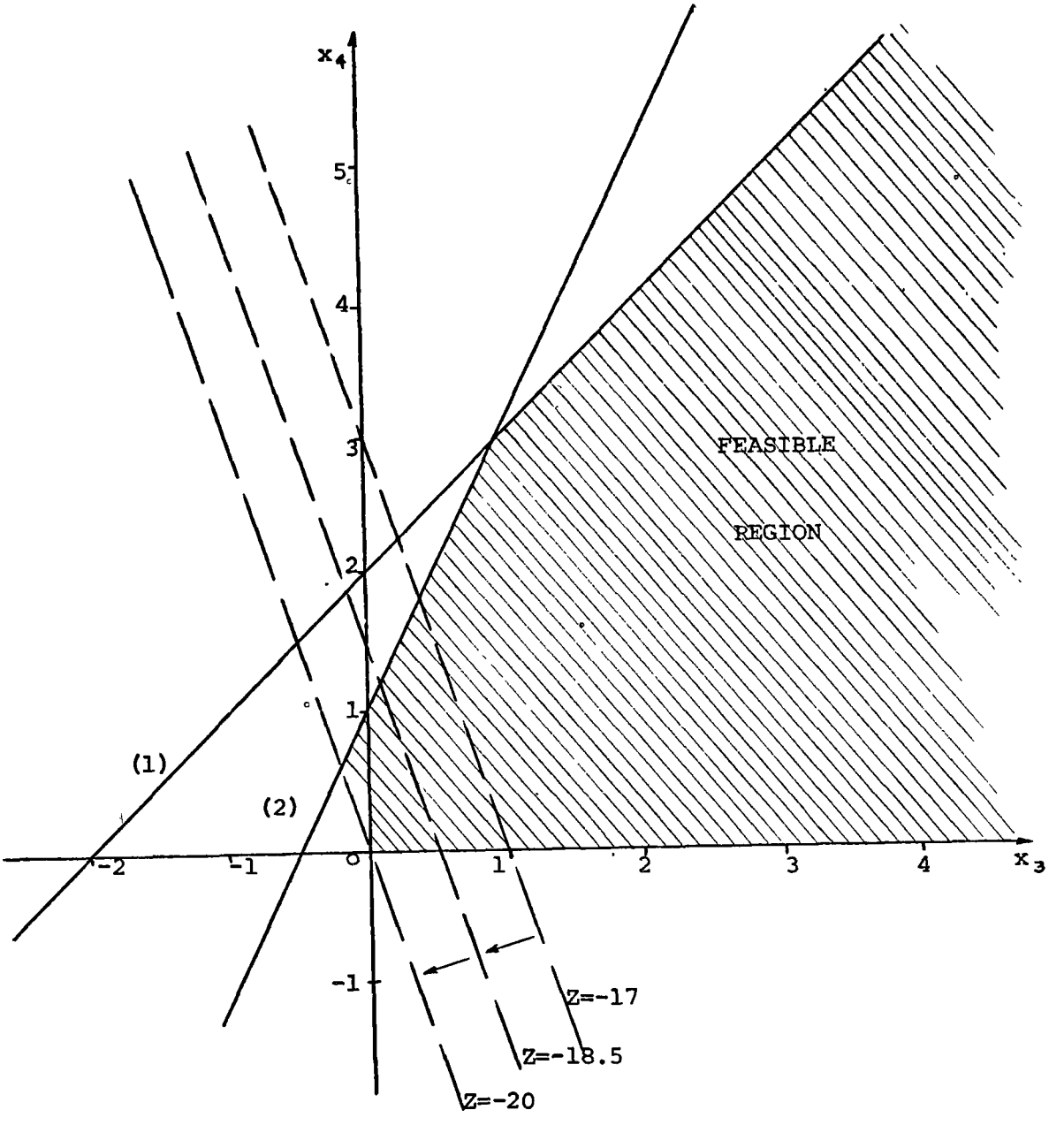
$$x_2 - 8x_3 + 4x_4 = 4 \qquad (2)$$

$$x_j \geq 0 \quad (j = 1,2,3,4)$$

Figure 1 illustrates the situation. For any value of z, say z = -17, the objective function is represented by a straight line. As z decreases to -20, the line corresponding to the objective function moves parallel to itself across the feasible region. At z = -20 it meets the feasible region only at the point $x_3 = x_3 = 0$ and, for z < -20 it no longer touches the feasible region. Consequently, z = -20 is optimal.

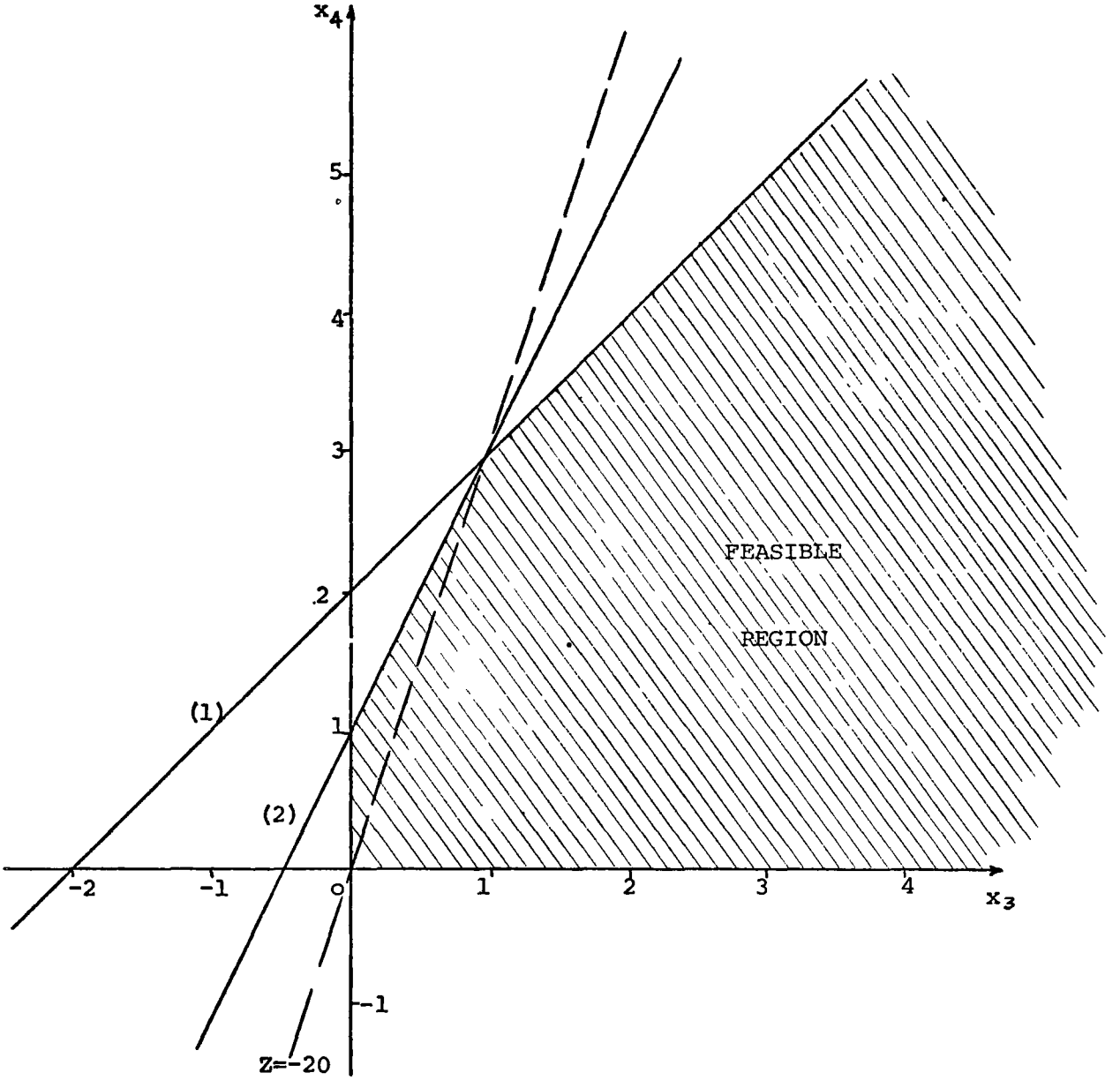
If we change our objective function to $z = -3x_3 + x_4 - 20$ (which is shown in figure 2) decreasing x_3, while holding $x_4 = 0$, corresponds to moving outwards from the origin along the x_3-axis. As we move along the axis, we never meet either constraint (1) or (2). Also, as we move along the x_3-axis, the value of the objective function is increasing

Figure 1: GEOMETRICAL INTERPRETATION OF EXAMPLE 1



to $+\infty$. Hence the objective function is unbounded from

Figure 2: GEOMETRICAL INTERPRETATION OF EXAMPLE 2



above.

On the other hand, if our objective function is changed to $z = 3x_3 - x_4 - 20$, which is also shown in figure 2, decreasing x_4 corresponds to moving from the origin along the x_4-axis. In this case, however, we encounter constraint 2 at $x_4 = 1$ and constraint 1 at $x_4 = 2$. Consequently, to mantain feasibility in accordance with the ratio criterion, we move to the intersection of the x_4-axis and constraint 2, which is the optimal solution.

2.3 THE KUHN-TUCKER CONDITIONS

In attempting to develop algorithms for solving nonlinear programming problems, it is useful to have some information concerning the characteristics of an optimal solution. Kuhn and Tucker [18] showed that if a vector x^* minimizes $f(x)$, subject to constraints $h(x) \geq 0$, then,

$$\partial f/\partial x_j + \sum_{i=1}^{m} t_i \, \partial h_i/\partial x_j = 0$$

$$h_i(x^*) \geq 0 \qquad \text{(KT1)}$$

$$t_i h_i(x^*) = 0$$

$$t_i \geq 0$$

for all $i = 1, 2, \ldots, m$

$j = 1, 2, \ldots, n$

where

t_i is the Lagrange multiplier associated with the i^{th} constraint.

These are the well-known Kuhn-Tucker conditions.

However, in the case of the QPP, the Kuhn-Tucker conditions may have a special form. The function $f(x)$ to be minimized is in this case

$$G(x) = (1/2)x'Dx + b'x + c$$

and the constraints $h(x) \geq 0$ are of two kinds

The side constraints

$$h^1 = q - Px \geq 0$$

and the nonnegativity constraints

$$h^2 = x \geq 0.$$

Then,

$$\partial G(x)/\partial x = Dx + b$$

$$\partial h^1(x)/\partial x = \partial/\partial x\ (q - Px) = -\sum_{i=1}^{n} u_i P_{ij} = -P'u$$

where u_i is the Lagrange multiplier associated with the i^{th} side constraint Clearly, the m-vector u is the Lagrange multiplier associated with the m side constraints $h^1(x)$. Similarly, for the nonnegativity constraints we have

$$\partial h^2(x)/\partial x = \sum_{j=1}^{n} v_j e_j = v$$

where e_j is the j^{th} column of the identity matrix I and v_j is the Lagrange multiplier associated with nonnegativity constraint x_j.

At the minimum point, the conditions are :

$$Ax + b = -P'u + v$$

$$u \geq 0$$

$$v \geq 0$$

$$x \geq 0$$

Moreover, $t_i h_i(x^*) = 0$ becomes

$u_i h^1(x^*) = 0$ or $u_i (q_j - \sum_{i=1}^{n} P_{ij} x_i) = 0$

$j = 1, 2, \ldots, n$

If y_i represents the nonnegative slack variable associated with the i^{th} constraint

$$y_i = q_i - \sum_{j=1}^{n} (P_{ij} x_i)$$

then, from above

$$u'y = 0$$

similarly, for the nonnegativity constraints $v'x = 0$. Then, $t_i h_i(x^*) = 0$ becomes $v'x + u'y = 0$ which is true. Since $v, u, x, y \geq 0$, the only way for it to hold is that $v'x = 0$ and $u'y = 0$.

Then, the Kuhn-Tucker conditions for the QPP can be written as

$$\begin{aligned} Ax + b + P'u &= v \\ v'x + u'y &= 0 \\ q - Px &= y \\ x, y, u, v &\geq 0 \end{aligned} \qquad \text{(KT2)}$$

Chapter III

THE METHOD OF BEALE

3.1 THEORETICAL ASPECTS OF THE METHOD

The method of Beale [1] was published as an application of the simplex method to quadratic programming and it reduces to the simplex method when the objective function is linear.

Consider the problem of minimizing the quadratic objective function $G(\bar{x})$ ($\bar{x}$ an n-vector), subject to the constraints

$$\bar{P}\bar{x} \leq q$$

$$\bar{x} \geq 0$$

where P is an mxn matrix and q is an m-vector.

We can assume without loss of generality that a basic feasible solution can be obtained from this set of constraints (see section 2.3.1). Call y the vector of the variables that give a basic feasible solution at the starting point, then the constraints become:

$$\bar{P}\bar{x} + y = q$$

$$\bar{x} \geq 0 \ , \ y \geq 0$$

or

$$[\ \bar{P} \mid I\ ] \begin{bmatrix} x \\ y \end{bmatrix} = [q]$$

Let

$$[x] = \begin{bmatrix} \bar{x} \\ y \end{bmatrix} \qquad \text{and} \qquad [P] = [\bar{P} \mid I]$$

Then x is an (n+m)-vector and P an mx(n+m). As a result the constraints can be written

$$Px = q$$

$$x \geq 0$$

It is assumed that the basic variables are the last x_{n+k} (k=1,2, ... , m) and the first x_j (j=1, 2, ... , n) are nonbasic. Define the vector $z = (z_0, z_1, \ldots, z_n)$

$$z = \begin{bmatrix} 1 \\ x \end{bmatrix}$$

Then the objective function can be expressed in terms of z as

$$G(z) = z' \Gamma z$$

where

$$\Gamma = \left[\begin{array}{c|c} c & b'/2 \\ \hline b/2 & (1/2)P \end{array}\right]$$

Note now that for $1 \leq k \leq m$ the $(k+1)^{th}$ entry of $\partial G/\partial z_k$ is

$$b_k + \sum_{j=1}^{n} a_{kj} z_j$$

The constraints can be used to express the basic variables in terms of the nonbasic ones

$$x_i = p_{io} - \sum_{j=1}^{n} p_{ij} z_j$$

where $p_{io} = q_i$ at the starting point where a basic feasible solution is immediately available. At any trial solution of the problem, the basic variable $x_i = p_{io}$ and the nonbasic variables z are all zero. Also, at any trial point, the equations of the constraints can be used to express the objective function G in terms of the nonbasic variables only.

If $\partial G/\partial z_s \geq 0$, then an increase in z_s will not reduce G; but if $\partial G/\partial z < 0$, then a small increase in z_s will reduce G. In the general case, when G is an arbitrary continuously differentiable function, the process of increasing z_s (when $\partial G/\partial z_s < 0$), must terminate when either one of the following conditions holds:

1. A basic variable, say x_t, has become zero and is about to become negative.
2. $\partial G/\partial z_s$ vanishes and is about to become positive.

In case 1., (which is the only possibility in the case of linear programming), the remedy is to change the basis by making x_t nonbasic in place of z_s. Case 2. can give rise to substantial difficulty if G is an arbitrary function; but if G is a quadratic function, then $\partial G/\partial z_s$ is a linear function of the nonbasic variables. In this situation, case 2. results in the creation of a new constraint.

If $\partial G/\partial z_s$ becomes positive as z_s is increased, (with all other nonbasic variables equal to zero) then a new variable is defined as follows:

$$u = b_s + \sum_{j=1}^{n} a_{sj} z_j$$

Now u is made the new nonbasic variable, and throughout the constraints and the expression for G, the above equation is used to substitute for z_s in terms of u and the other nonbasic variables. Note that if z_s is one of the original x-variables (not one of the introduced u-variables), then there will be additional basic variable after this step.

One important feature of the new u variable is that it is not restricted to nonnegative values. It is therefore called a 'free' variable as opposed to the original x-variables which are called restricted variables. Since u is not sign restricted, if $\partial G/\partial u > 0$ then G can be reduced by making u negative.

It is convenient to have a set of nonbasic variables that all vanish at the trial solution, since the values of the basic variables are then simply given as the constant terms in the tableau. The new variable is chosen so that it will not be profitable to change its value when we change the values of any other nonbasic variable. It has been shown [19,3] that this method terminates in a finite number of steps.

3.2 UPDATING THE TABLEAUX

When passing from one iteration to the next, the expressions for the constraints are handled exactly as in linear programming, (the constraints part is often called the simplex tableau). However, for the objective function, there is a different approach.

TABLE 1

INITIAL TABLEAU FOR BEALE

	variables	value	z
Constraints tableau	x	q	-P
objective function tableau		c	b'/2
	z	b/2	$\frac{1}{2}$

Table 1 shows the initial tableau for the method of Beale. The constraints tableau shows the basic variables x in terms of the nonbasic ones. The objective function tableau shows the symmetric matrix Γ of size (n+1) x (n+1) whose first row can be used to determine optimality in the trial point: a positive value in each one of the columns v, where z_v is a restricted variable and a zero value in each

of the columns t, where z_t is a free variable (*), denotes an optimal tableau.

The procedure suggested by Beale can be expressed algebraically by saying that, starting from the objective function that is in terms of the nonbasic variables z, the z-variable that is to become basic is replaced by the equivalent expression in terms of the basic variable x (that is to become nonbasic) and the other nonbasic variables, giving rise to a new expression for G in terms of the new nonbasic variables. This is not very convenient in a computer, since it involves operating on both the rows and columns of the matrix. It is worthwhile examining the algebraic expressions for the combined effects of these transformations.

If we denote the pivotal column by s and if the expression for the new basic variable z is, in terms of the new nonbasic variables

$$z_t = e_o + e_s z_s + \sum_{K \neq s} e_K z_K$$

where z_s denotes the new nonbasic variable replacing z_t, then the new coefficients g'_{kj} are given in terms of the old coefficients g_{kj} and the e_K as follows :

$$g'_{ss} = g_{ss}\, e_s^2$$

$$g'_{Ks} = g'_{sK} = g_{Ks}\, e_s + g_{ss}\, e_s\, e_K$$

$$g'_{kj} = g_{kj} + g_{Ks}\, e_j + g_{sj}\, e_K + g_{ss}\, e_K e_j$$

where $k, j \neq q$

If g* is defined as

$$g^* = (1/2)g_q e$$

$$g^* = g_k + (1/2)g_q e \qquad \text{for } k \neq q$$

then, the above expressions can also be written

$$g' = 2g^* e$$

$$g' = g'_q = g^*_q e + g_q e$$

$$g' = g_q + g^*_q e + g^*_q e$$

The optimum point is reached when the derivatives of the objective function with respect to each of the nonbasic variables are nonnegative and the derivative of the objective function with respect to the free variables introduced in the problem are all zero. This is so because at this point an admissible change in any of the restricted variables will increase the value of the objective function and it is impossible either to increase or to decrease any of the free variables. Hence the value of G cannot be minimized further.

3.3 A NUMERICAL EXAMPLE

To illustrate this method, an example given by Beale [2] that has the characteristic of highlighting the principal features of the algorithm, is presented.

The problem is

$$\text{minimize } G = -6x_1 + 2x_1^2 - 2x_1 x_2 + 2x_2^2$$

subject to $x_1 + x_2 \leq 2$ $\qquad x_1, x_2 \geq 0$

Introducing a slack variable x_3, we have the constraint:

$$x_1 + x_2 + x_3 = 2 \qquad x_1, x_2, x_3 \geq 0$$

A feasible solution for starting the process is

$$x_1 = 0 \qquad x_2 = 0$$

This corresponds to point A in figure 3. At this point, the following expressions for the variable x_3 and the objective function hold:

$$x_3 = 2 - x_1 - x_2$$

$$G = (\quad - 3x_1 \quad) 1$$

$$+ (-3 + 2x_1 - x_2) x_1$$

$$+ (\quad - x_1 + 2x_2) x_2$$

and so at the trial point A

$$x_1 = 0 , \quad x_2 = 0 , \quad x_3 = 2 , \quad G = 0$$

Since $\partial G/\partial x_1 = -3$, G can be decreased by increasing x_1 (x_2 held at its value of 0)

$\partial G/\partial x_1 = -3 + 2x_1 - x_2$ vanishes for $x_1 = 3/2$

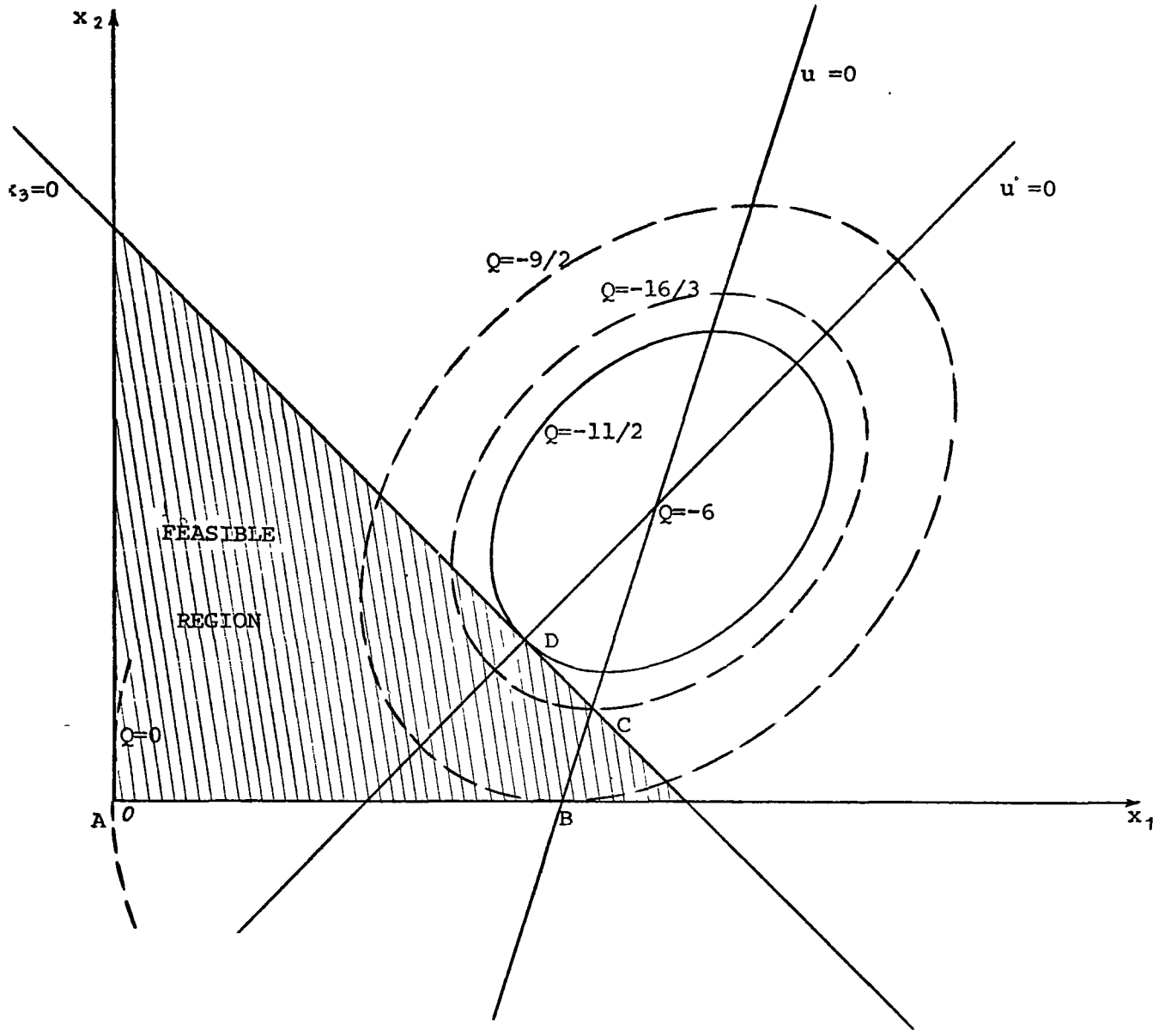
This point still belongs to the feasible region since x_3 equals 0 only when $x_1 = 2$ and then $\partial G/\partial x_1 = -3 + 2x = +1$

so, a free variable is introduced:

$$u_1 = \partial G/\partial x_1 = -3 + 2x_1 - x_2$$

where the subscript 1 on u means that it is the first free variable introduced into the problem. This u_1 variable replaces x_1 as an independent variable that vanishes.

Figure 3: THE METHOD OF BEALE ON EXAMPLE



u_1 is zero in all the points where the ellipse G =

constant has a horizontal tangent. From above : $x_1 = 3/2 + (1/2)u_1 + (1/2)x_2$

Hence

$s = 1 \quad e_o = 3/2 \quad e_1 = 1/2 \quad e_2 = 1/2$

$(1/2)q_{ss} = 1 \quad q^*_o = -3/2 \quad q^*_1 = 1/2 \quad q^*_2 = -1/2$

Then the second trial point can be set and the tableau reads

$$x_1 = 3/2 + (1/2)u_1 + (1/2)x_2$$

$$x_3 = 1/2 - (1/2)u_1 - (3/2)x_2$$

$$G = (-9/2 \qquad -(3/2)x_2)\,1 + (\quad (1/2)u_1 \quad)u_1 + (-3/2 \qquad (3/2)x_2)\,x_2$$

At the second point B

$u_1 = 0$, $x_2 = 0$, $x_1 = 3/2$, $x_3 = 1/2$, $G = -9/2$

Since $\partial G/\partial x_2 = -3/2$, x_2 must be put into the basis (u_1 stays 0) :

$\partial G/\partial x_2 = -3/2 + (3/2)x_2$ vanishes for $x_2 = 1$

It is not possible to reach this value since x_3 vanishes before x_2, that is, for $x_2 = 1/3$.

Then we make x_3 nonbasic and we express the new basic variable x_2 and the previous one x_1, as functions of x_3 and u_1, the new nonbasic set.

This gives as a third point :

$$x_2 = 1/3 - (1/3)u_1 - (2/3)x_3$$

$$x_1 = 5/3 + (1/3)u_1 - (1/3)x_3$$

Hence,

$s = 2 \quad e_o = 1/3 \quad e_1 = -1/3 \quad e_2 = -2/3$

$(1/2)\,q_{ss} = 3/4 \quad q_0^* = -5/4 \quad q_1^* = -1/4 \quad q_2^* = -1/2$

And the objective function at this point is :

$$G = (-16/3 + (1/3)u_1 + (2/3)x_3)\,1 + (1/3 + (2/3)u_1 + (1/3)x_3)\,u_1 + (2/3 + (1/3)u_1 + (2/3)x_3)\,x_3$$

The trial point corresponds to point C

$u_1 = 0$, $x_3 = 0$, $x_2 = 1/3$, $x_1 = 5/3$, $G = -16/3$

Since an x-variable has been made nonbasic, u_1 must become basic. The only way to decrease G at this point is by decreasing u_1. because $\partial G/\partial u_1 > 0$. This is possible until

$$5/3 + (1/3)u_1 - (1/3)x_3 = 0$$

that is, for $u_1 = -5$. But $\partial G/\partial u_1 = 1/3 + (2/3)(-5) = -3$ vanishes first, so the next nonbasic variable is a free variable u_2 defined by

$$u_2 = 1/3 + (2/3)u_1 + (1/3)x_3$$

i.e.

$$u_1 = -1/2 + (3/2)u_2 - (1/2)x_3$$

hence,

$s = 1 \qquad e_0 = -1/2 \qquad e_1 = 3/2 \qquad e_2 = -1/2$

$(1/2)\,q_{ss} = 1/3 \quad q_0^* = 1/6 \quad q_1^* = 1/2 \quad q_2^* = 1/6$

and a new tableau is available:

$$u_1 = -1/2 + (3/2)u_2 - (1/2)x_3$$

$$x_1 = 3/2 + (1/2)u_2 - (1/2)x_3$$

$$x_2 = 1/2 - (1/2)u_2 - (1/2)x_3$$

$$G = (-11/2 + (1/2)x_3)1 + (3/2)u_2)u_2 + (1/2 + (1/2)x_3)x_3$$

Since the derivative of G with respect to the free variable u_2 is zero and the derivative with respect to the restricted variable x_3 is positive, the minimum point has been reached. The problem solution therefore is:

$$x_1 = 3/2 \ , \quad x_2 = 1/2 \ , \quad x_3 = 0 \ , \quad G = -11/2$$

This corresponds to point D in figure 3

3.4 THE ALGORITHM

1. Optimality : Check the first row of the objective function tableau. If the tableau is optimum, stop.

2. Pivot column : Select in the first row of the tableau, among the u-column a non-zero element. If all the elements in the first row corresponding the u-columns are zero or there are no u-columns then choose among the x columns the most negative. Call

this element g . Then r is the pivot column, which defines the variable z (either u or x) that becomes basic.

3. Pivot row : Divide the absolute value of the element q by the element a in the diagonal of the objective function tableau. If this element is positive, then divide the elements p of the first column of the constraints tableau by the absolute value of the correspondant element p in the pivot column, but only if this element has the same sign as the element g . Obtain the same ratio in all the rows (except the first one) of the objective function tableau having in the first column one element with the same sign as g , (take absolute value of q , h ≠ 0). The row giving the minimum ratio is selected as pivot row. The element located in the intersection of the pivot column and the pivot row (either q or p) is called the pivot element.

4. If the pivot element is p , that is the pivot row belongs to the constraints tableau, replace the variable z (in the top of the pivot column) by the variable z (heading the pivot row).

5. If the pivot element is q , that is the pivot row belongs to the objective function tableau, replace z

by a new variable u, that goes to the constraints tableau.

6. Update the tableaux (constraints section and objective function section) and repeat from step 1.

Chapter IV

SOME METHODS FOR THE QPP AND THEIR COMPUTATIONAL PERFORMANCE

Several articles have appeared in the literature comparing the computational efficiency of three of the best known simplicial methods of quadratic programming [1,11,30], namely the methods of Wolfe, Beale and Dantzig. A large number of problems have been solved with coefficients which were random variates obtained from a pseudo-random number generator. In certain cases, problem parameters were varied to determine whether they have a significant effect on the rate of convergence of the algorithms.

The following results are taken from a study carried out by Braitsch [8], who followed an experimental approach to compare the algorithms of Wolfe, Beale and Dantzig. A fourth method is also treated which is a slight modification of Wolfe's algorithm. This modification have been introduced in an attempt to reduce the effects of some of the inefficiencies inherent in Wolfe's algorithm. The method of Beale was presented in the previous chapter. A brief discussion of the methods of Wolfe and Dantzig is presented here.

4.1 THE ALGORITHM OF WOLFE

Wolfe [30], working from an idea suggested by Makowitz [23], developed a method for the QPP that is based on solving the Kuhn-Tucker conditions. Computationally, it corresponds to a simple modification of the artificial variable technique for finding a first feasible solution to a linear programming problem [7].

For the problem (P1), (3n + 2m) supplementary variables are added to the corresponding Kuhn-Tucker conditions (KT2)

$$w_i \qquad i = 1, 2, \ldots n$$

$$z_j^1,\ z_j^2 \qquad j = 1, 2, \ldots (m+n)$$

Now note that the x-vector which satisfies the following conditions when $w = z^1 = z^2 = 0$ and $r = 1$, is the solution to the original problem:

$$Px + y + w = q$$

$$Ax - v + P'u + z^1 - z^2 + rb = 0$$

$$v'x + u'y = 0$$

$$x, v, y, u, w, z^1, z^2 \geq 0$$

The simplex method is performed with an additional rule : $v'x + u'y = 0$ must hold throughout the procedure. This assures that at every point and for all i, no v_i and x_i can have nonzero value (cannot be in the basis) at the same time.

Wolfe's method exists in two forms: The short form that finds a minimum only if b = 0 or A is positive definite and

the long form that can be applied without restrictions. The short form requires two phases in the procedure and the long form adds one more phase to the results obtained with the short form. In the first phase, r is set to 0, such that $rb = 0$ (since in the short form, this guarantees a solution) and w is minimized (in fact, driven to zero), holding r, u and v out of the basis.

The initial tableau is set:

$$w = q - Px - y$$

$$z^1 = 0 - 'x + v - P'u + z^2 - rb$$

and the simplex method is applied until $\sum_{i=1}^{n} w_i = 0$.

The second phase minimizes z^1 and z^2. At this point, there is no need to differentiate between z^1 and z^2 since both must vanish. r must not enter the basis in this phase. Again, the simplex method is used until the sum of all entries in z^1 and z^2 is zero. This concludes the short form.

In the third phase, (in the long form) r is set to 1. and the objective is to obtain two successive tableaux in which the r values are slightly below and above 1. With each of these r's there is associated a solution vector and a linear interpolation is used to obtain the solution vector corresponding to r=1, which is the problem solution.

4.2 THE METHOD OF DANTZIG

G.B. Dantzig [11] took the algorithm of Wolfe and developed a slightly more compact version of the method that works somewhat faster if A is either positive definite or semidefinite. The significant shortcoming is that it may not terminate at all if A is indefinite.

It uses the Kuhn-Tucker conditions (KT2) and the initial tableau is set as:

$$v = b + Ax + P'u$$

$$y = q - Px$$

An important distinction between two kinds of solutions must be stated: A tableau is said to be in standard form if $v'x + u'y = 0$ holds. Otherwise it is in nonstandard form. The initial tableau is in standard form: v and y are basic with nonzero values and u and x are nonbasic, hence their values are zero. Whenever a tableau is in nonstandard form, the next iteration takes it back to the standard form.

The simplex method is performed with some special rules for interchanging the variables: If a tableau is in standard form, the variable to enter the basis is the x_k whose corresponding v_k (basic) has the largest positive value. If a tableau is in nonstandard form, the variable to enter the basis is the nonbasic v_j whose corresponding x_j is also nonbasic. The variable to be deleted from the basis is

selected with the same minimum ratio criterion used in the simplex method for linear programming [7,10].

The objective is to minimize $\sum_{j=1}^{n} v_j$. A standard tableau is optimum when the values of v in the basis are all negative: that is, no v applies for a new iteration. The x vector in this optimum tableau is the solution to the problem.

4.3 EXPERIMENTATION

In Braitsch's study some preliminary experiments were carried out to help select the factors and levels for the primary experimentation. Listed below are the nine factors selected that may be important in determining how fast a quadratic programming algorithm will converge.

1. Algorithm : Refers to the algorithms of Wolfe, Dantzig, Beale and W.B. The following factors apply to each of the four algorithms.
2. Rule : Rule used in each algorithm for selecting the variable to enter the basis. In all cases but three, the variable entered was one that resulted in the largest rate of increase in the criterion function. The exceptions were WB, where no selection rule was necessary and the nonstandard form cases of Beale and Dantzig.

3. Size : Defined as the number of x-variables or the number of constraints in the P matrix, since squared P-matrices of size m x m were used. Sizes of 15, 20 and 25 were used.

4. Negative b : The number of negative coefficients in the cost vector b. Levels of 3, 9 and 15 were used.

5. Rank : Number of linearly independent columns in the matrix A. Ranks of 3 and 15 were used.

6. Range : Range of the random numbers used to generate the P-matrix on the computer. The ranges (r) used were : $-20 \leq r \leq 20$ and $-180 \leq r < 180$

7. b-range : Range of the random numbers used to generate the b-vector. Ranges used were $-20 \leq r \leq -1$ and $-180 \leq r \leq -1$

8. q-range : Ranges for the random numbers in the q vector. Ranges used were $1 \leq r \leq 20$ and $1 \leq r \leq 180$

9. P-range : Only the single level of $1 \leq r \leq 180$ was used for the random numbers that made up the P-matrix.

A complete factorial model was used where, for each of the four algorithms, all combination of factors (3) through (8) were run. The experiment was replicated ten times. The various factor combination were run by nesting a series of loops in the computer program, each loop corresponding to a factor. It was assumed that since the problem components

were randomly generated, further randomization would not be necessary. Each randomly generated problem was solved by all four algorithms.

4.4 RESULTS AND ANALYSIS

TABLE 2

MEAN NUMBER OF ITERATIONS FOR DIFFERENT LEVELS

FACTOR		DANTZIG	BEALE	W.B.	WOLFE
Size :	15	6.9	7.8	8.3	19.5
	20	7.5	8.7	8.9	21.6
	25	7.8	9.3	9.6	23.3
Negative p :	3	4.5	4.6	5.3	7.6
	9	7.8	8.8	9.4	20.9
	15	9.9	12.4	12.1	35.8
Rank :	3	7.3	8.6	8.9	21.6
	15	7.5	8.7	9.0	21.3
A-range :	[-20,20]	6.1	4.8	7.6	22.9
	[-180,180]	8.7	12.4	10.3	20.1
b-range :	[-20,-1]	7.9	10.1	9.5	20.6
	[-180,-1]	6.9	7.1	8.4	22.3
q-range :	[1,20]	7.2	7.7	8.6	22.5
	[1,180]	7.6	9.5	9.3	20.4
Algorithm means		7.4	8.6	8.9	21.5

Table 2 shows the average number of iterations taken at each level of all of the factors that were varied, for each of the algorithms.

The data in this table lead to the conclusions that the rank of the matrix has little effect, but the size factor is very significant. A number of the results for Dantzig's and Beale's algorithms are explainable in terms of the relative importance of the linear and nonlinear part of the objective function. It is clear that increasing the b-range or decreasing the A-range will emphasize the linear part of G(x). Also, decreasing the q-range or increasing the P-range increases the importance of the linear part. This is true because these latter two actions tend to lower x-values which reduces the contribution of the quadratic part more than that of the linear part.

Average iterations decreased in the three cases mentioned above, that is increasing the b-range, decreasing the A-range and decreasing the q-range. This is reasonable, because in the extreme case when the quadratic portion is irrelevant and the problem becomes a linear programming problem, both methods reduce to the Simplex method which moves between adjacent extreme points in search of solution. In the nonlinear case, when the quadratic part is significant, the solution often lies in one of the faces in the feasible region rather than at an extreme point. This tends to add iterations because of the necessity of optimizing on faces. These extra iterations often include increasing the number of basic primal variables.

There is a major difference in the way Beale's and Dantzig's methods perform in the Linear Programming case. While Beale's method reduces to the Simplex method in the constraints portion, Dantzig's alternates between regular Simplex iterations and Dual Simplex iterations in two parts of the tableau, thus taking twice as many iterations as Beale's. As the quadratic part becomes more important, average iterations increase at a greater rate in Beale's algorithm than in Dantzig's. The major reason for this is that when we move from one face in the feasible region to another (by making a sign restricted variable nonbasic), it is necessary to replace all of Beale's free variables because these variables were used to optimize on the previous face.

Since each unit of size augmented adds a constraint and an x-variable, the new constraint adds another face to the feasible region which could raise iterations if the algorithm had to examine it on the way to the optimal solution.

Negative b is a very important factor in Dantzig's and Beale's methods. Each added negative b-coefficient adds a variable that can profitably enter in the first iteration for both algorithms. Although not all of the x-variables having negative b-coefficients will be in the optimal solution, the number will tend to increase with negative b

and iterations will be required to pivot them into the basis. This discussion does not apply to Wolfe's algorithm because it uses a different type of objective function than the other three algorithms. Algorithm WB generally requires fewer iterations because it uses only one artificial variable. The performance of this algorithm is difficult to explain, for although it is a modification of Wolfe's algorithm, it reacts to different problem types in the same way as Dantzig's.

4.5 COMPUTER TIME AND STORAGE

It is difficult to get an accurate measure of the computer time required by the various algorithms because of the large difference that programming technique can cause. A more reliable measure might be based on the number of tableau elements that must be transformed at each iteration. In this way, we can take advantage of theoretical shortcuts for improving time efficiency that might normally be ignored by a programmer. These shortcuts include transforming only half of Beale's Objective tableau because it is symmetric, eliminating transformations in Dantzig's standard-form tableaux because of symmetry relations and reducing the

space that must be set aside due to free variables in Beale's algorithm by noting that the number of these free variables cannot exceed the rank of the A-matrix.

In the experimentation, approximately 70 percent of Dantzig's tableaux were in standard form. All tableaux tranformations were assumed to take one unit of time, except for the transformations in Beale's objective tableau, which took about 1.6 units in computer tests. The time to transform variables added in Beale's simplex tableau due to free variables will be ignored because it is difficult to estimate, and judging from data available, probably would not affect the time estimate by more than a few percent at worst. Table 5 gives time and storage estimates for several combinations of problem parameters.

Timewise, Beale's and Dantzig's algorithms are superior, with Beale's algorithm tending to be preferable when the number of x-variables does not greatly exceed the number of constraints. This is due to the fact that Beale's tableaux are in terms of the nonbasic variables. In terms of storage, Beale's algorithm takes better advantage of situations where the P-matrix is nearly square or the rank of A is not a very large fraction of the number of rows or columns in A. The latter phenomenon is due to a reduction in space necessary

TABLE 3

TIME AND STORAGE REQUIREMENTS

M	N	A-rank	Algorithm	Time	Storage
Z	Z	Z	BEALE	1.8	2.5
			DANTZIG	2.6	4
			W. B.	4	4
			WOLFE	4	4
Z	2Z	2Z	BEALE	5.2	8
			DANTZIG	6	9
			W. B.	9	9
			WOLFE	9	9
Z	4Z	4Z	BEALE	16.8	28
			DANTZIG	16.3	25
			W. B.	25	25
			WOLFE	25	25
Z	4Z	Z	BEALE	16.8	16
			DANTZIG	16.3	25
			W. B.	25	25
			WOLFE	25	25

With, N: number of variables, M: number of constraints
Time: Time per iteration (Z^2 units) and one unit = 23×10^{-6} secs. on the IBM 360-75
Storage: Requirements of storage in memory (Z^2 locations)

to hold the rows of variables driven out of the nonbasic set because of the introduction of free variables.

The results of this study suggest that in most of the cases, Beale's algorithm will generally take more iterations than Dantzig's, but the difference is not very big. But Beale's algorithm has some other advantages : It will find a local minimum even if A is not definite which may be of value in some applications; second, a redefinition of the

A-matrix, which when combined with a computer code using the product form of the inverse, can solve larger problems than Dantzig's algorithm, provided the rank of A is low.

Chapter V

THE INTERACTIVE PACKAGE

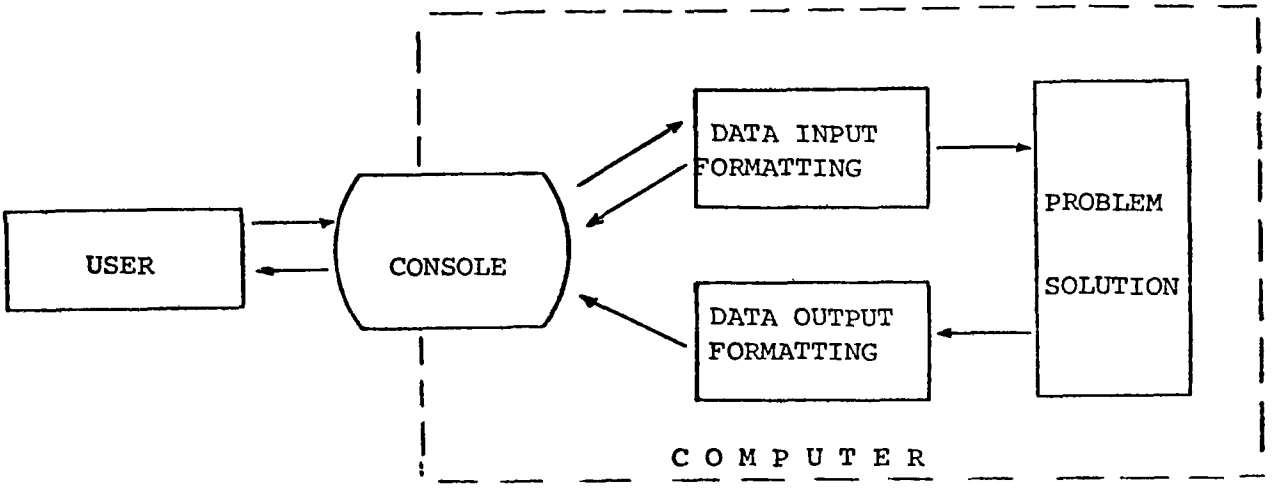
5.1 STRUCTURE OF THE PACKAGE

The method of Beale was selected among the methods available after considering several important factors and, specifically, because

1. Beale's method does not require matrix D to be positive definite : it works also with a negative definite as well as in the semi-definite cases.
2. The tableaux used are small, giving economy in storage and computing time.
3. It uses the simplex method, thus is efficient.
4. The computational properties are acceptable or superior to the other methods compared.
5. Is the geometrically most illustrative among all methods for the QPP.
6. The procedure is relatively simple, thereby facilitating the programming task.

Figure 4 provides a general view of the package format. It consists of two major parts: A first part deals with the communication with the user, interchange of information and

Figure 4: PACKAGE INTERACTION



the transformation of the data into a suitable format for both the user and the program itself. The second part is the procedure that actually solves the problem.

The execution of the package begins by giving the user a summary of its objective, the QPP is defined and some hints are given for putting the general objective function into matrix form. The user is informed about the procedure for communicating with the package.

No special format for the input data from the terminal is needed. The data entered by the user is checked and only the expected type of data is accepted. This is made with the aid of TAPE 99, a FORTRAN device that enables the reading of data with variable format.

The data received is displayed again on the terminal and the user is asked to check its accuracy. Corrections are permitted when errors are found. The tableau for the method of Beale is constructed from the data received and various parameters are initialized. The initial tableau is displayed if the user has chosen this output option.

The Beale algorithm is then initiated to solve the QPP, and again, according to the user's selected output option, intermediate tableaux and/or the final solution are displayed.

5.2 STORAGE ORGANIZATION

The procedure requires the matrix

$$\left[\begin{array}{c|c} c & b'/2 \\ \hline b/2 & \frac{1}{2}P \end{array}\right]$$

which is (n+1)x(n+1) and symmetric (because P is symmetric). In order to save storage, only the upper triangular part of this matrix is stored. The mx(n+1) matrix

$$[\ q \mid -P\ ]$$

is also created.

All 2-dimensional arrays required in the procedure are stored in a vector format which is created from the catenation of the rows of the matrix. For example, the 2x3 matrix R, where

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \end{bmatrix}$$

is stored as the 6-vector

$$v = \begin{bmatrix} r_{11} \\ r_{12} \\ r_{13} \\ r_{21} \\ r_{22} \\ r_{23} \end{bmatrix}$$

It then follows that the entry in row i, column j of R appears in v(k) where

$$k = (i-1)\alpha + j$$

and α is the column dimension of R ($\alpha = 3$ in this example). A slight modification of this process is used to store only the upper triangular part of symmetric matrices.

A means for distinguishing the variables that belong to the basis is required. For this purpose, the original nonbasic variables are always given the indices 1 through n

and the original basic variables the indices n+1 through n+m. The free variables, referred to as "u" variables in chapter 3, are treated internally as x variables and they are assigned indices between n+m+1 and 2n+m. This was done to simplify their treatment and there is no risk of confusion for the user since these variables do not appear in the solution vector.

5.3 NOTATION AND DEFINITIONS

Throughout the programs, whenever possible, the same notation as that in chapter 3, was used.

N — The number of variables (user input); remains fixed during the execution of the procedure.

M — The number of constraints (user input); may vary during the procedure when free variables are added (each one yields a new constraint).

VA — A N*(N+1)/2 - vector containing the upper triangular part of the A matrix in the quadratic objective function (user input).

B — Coefficient vector of the linear part of the objective function (user input).

C	The constant value in the objective function (user input).
VP	A M*N - vector containing the entries of the constraint matrix P (user input).
Q	M-vector with the right-hand side values of the constraints (user input).
EPS	A small positive number corresponding to the width of the zero-band used in checks for zero at critical points during the solution (user input).
BASIC0	N-vector containing the indices of the nonbasic variables.
BASIC1	M-vector containing the indices of the basic variables.
LOS	Vector containing the variables remaining in the objective function which are different from the free variables.
LAS	Vector of the basic variables.
LIBRE	Counts the free variables added to the problem.
ALIBRE	Stores the number of free variables at a given point.
IPIV	Stores the index of the pivot row.
KPIV	Stores the index of the pivot column.
FL	Set to n+1.
PICO	Logical variable whose value is .TRUE. if the pivot is in the COnstraints part; otherwise, its value is .FALSE.

ISDONE Has a value 1 if the problem has been solved; otherwise it has a value of 0.

MAX Contains the maximum value in the first row of the Objective function tableau.

YO Contains the value of the minimum ratio a(i,0)/a(i,j) for which a(i,0) is nonnegative and a(i,j) is positive.

JC Set to N*(N+1)/2.

J1 Set to N*M.

SOL The vector which contains the problem solution.

DSPLAY The value assigned to this variable (by the user) serves to specify one of several available display options.

ITER Counts the number of iterations that have been carried out.

5.4 THE SUBPROGRAMS

The overall program is divided into twenty one FORTRAN subroutine subprograms. Each performs a specific task. These subroutines are linked and controlled by the Main program which serves as a driver routine.

5.4.1 The Main program

This program calls subroutines DIALOG, INITIA, SETMTX, SHOWIT, and RUNIT. It interchanges information among them and transfers control from one to the other as the solution process is carried out.

5.4.2 Subroutine DIALOG

Presents the package to the user, defines the problem and the variables, gives instructions for the use of the program and gets from the user the values of the parameters to be used in the problem, checks the validity of these values and if necessary accepts any correction suggested. The subroutines called from within DIALOG are READIT, READFM and MESSAG. This subprogram is called only by the Main program.

5.4.3 Subroutine READIT

Reads one integer value from the terminal (specifically for N,M and DSPLAY). Analyzes the data received and sends messages to the user when necessary. This subprogram calls subroutine MESSAG and is called only by subroutine DIALOG.

5.4.4 Subroutine READEM

This program reads a predetermined set of real values from the terminal. Transforms the data into double precision numbers and analyzes each input accepts it or sends an error message when necessary. It is used to read the values of A, B, C, P, Q and EPS. This subprogram calls subroutine MESSAG and is called only by subroutine DIALOG.

5.4.5 Subroutine INITIA

It initializes the values of the parameters to be used in the program. Sets values for BASIC0, BASIC1, LOS, LAS, JC, J1, LIBF, ALIBRF, LIBRT, RL and ITER. This subprogram is called only by the Main program.

5.4.6 Subroutine SETMTX

Transforms the data received from the user into a form suitable for use in Beale's method; i.e., it creates the matrix Γ shown in section 3.1. It also creates the matrix $[q \mid -P]$ for the constraints tableau. This subprogram is called only by the Main program.

5.4.7 Subroutine SHOWIT

Displays information relating to the tableaux according to the value of DSPLAY. It is called at the end of each iteration to inform the user about the progress of the computations. If DSPLAY = 2, subroutines PROFUN and PRICON are called. If DSPLAY = 1, the current solution is displayed. If DSPLAY = 0 control is returned to the calling routine. This subroutine is called by both, the Main program and the subroutine RUNIT.

5.4.8 Subroutine RUNIT

It is the subroutine that carries out Beale's algorithm. It calls subroutines MALIB, COLPIV, SETV, RATIO, CONACT, OFUACT, ADDONE, SHOWIT, CHANGE, DELETE and OPTIMO. It performs certain operations and transfers control to the subroutines, according to the results obtained. Controls the interaction of all the subroutines and identifies when the problem does not have a solution or when a solution has been found. This subprogram is called only by the Main program.

5.4.9 Subroutine MALIB

This subroutine finds the largest value in the first row of the objective function tableau among the columns associated with the free variables. It is called each time a free variable is added to the problem. If MALIB finds any

positive value, this position defines the column of the pivot for the next iteration. It is called only by subroutine RUNIT.

5.4.10 Subroutine COLPIV

It finds the column of the pivot for the next iteration. If MALIB has not found a positive value, COLPIV searches in the first row of the objective function tableau for the most negative number and defines that position to be the column of the pivot for the next iteration. If COLPIV doesn't find any negative value, we are at the optimum point. COLPIV is called only by subroutine RUNIT.

5.4.11 Subroutine SETV

SETV sets the value of a variable V to be either 1 or -1, according to the sign of the element in the matrix Γ located at the intersection of its main diagonal and the pivot column. It is called only by subroutine RUNIT.

5.4.12 Subroutine RATIO

This subroutine finds the Pivot row. To do this, it determines the least value of the ratio $a(i,0)/a(i,j)$ where j is the column of the pivot, as i takes on its admissible values. If there are 2 different i's which yield the same minimum value, then this operation is repeated for all j's

different from the pivot column. This second step is necessary in order to avoid degeneracy in the next iteration. It is called only by subroutine RUNIT.

5.4.13 Subroutine CONACT

It updates the tableau of the constraints. Performs the Simplex procedure to interchange the variables determined by the pivot.This subroutine is called only by subroutine RUNIT.

5.4.14 Subroutine OFUACT

It updates the tableau of the objective function. Performs the necessary computations to update the tableau at each iteration. It is called by subroutine RUNIT.

5.4.15 Subroutine ADDONE

Adds one more free variable to the problem This means also a new constraint to the system that goes to the simplex tableau. This subroutine is called only by subroutine RUNIT.

5.4.16 Subroutine CHANGE

It performs the interchange of variables after each iteration and modifies the values of the parameters affected by this change. It is called by subroutine RUNIT.

5.4.17 Subroutine DELETE

This subprogram deletes a constraint from the system, after the value of the free variable that originated it has reached a zero value in the objective function and a new constraint enters the system. It is called by the subroutine PUNIT.

5.4.18 Subroutine PROFUN

It prints the tableau of the objective function. It reconstructs the matrix Γ and shows the result. Up to five columns are displayed at one time. If the tableau has more than five columns, they are successively displayed in groups of five until all have been shown. This subprogram is called by subroutines SHOWIT and OPTIMO.

5.4.19 Subroutine PRICON

This subroutine prints the tableau of the constraints. It reconstructs the P matrix and as in PROFUN, only five columns are shown at a time. The remaining columns are displayed in subsequent tableaux. PRICON is called by subroutines SHOWIT and OPTIMO.

5.4.20 Subroutine OPTIMO

It displays the problem solution. Prints a suitable message depending on whether ISDONE equals 0 or 1. If DSPLAY = 2, it calls PROFUN and PRICON, otherwise, creates the solution vector SOL and displays it with the appropriate messages. It is called by subroutine RUNIT.

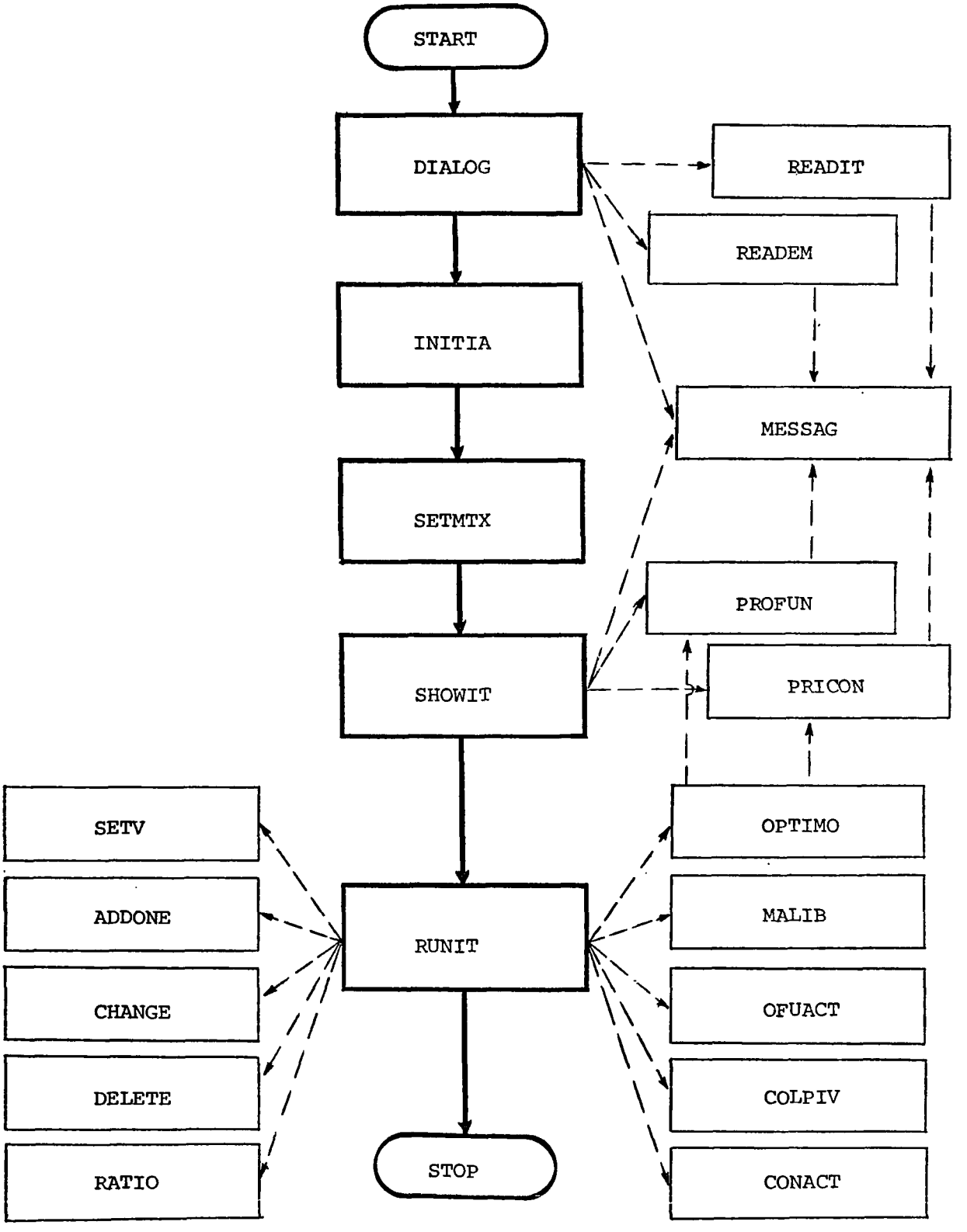
5.4.21 Subroutine MESSAG

Prints messages for the user. It is called by subroutines DIALOG, PROFUN, PRICON, READIM, READIT, SHOWIT and OPTIMO.

5.5 PROGRAM ASSEMBLY

Figure 5 shows how the routines are interconnected. The darker blocks represent the subprograms that are sequentially called within the main program. The dotted arrows represent subroutine calls.

Figure 5: THE PROGRAM AND THE SUBROUTINES



Chapter VI

USING THE PACKAGE

6.1 CONNECTING TO THE PACKAGE

The package is to be used under VM-CMS. To run it, the user must have the following files in his disk

CUADRO EXEC

SETB99 TXTLIB

CUADRO FORTRAN

The file CUADRO EXEC contains:

```
     GLOBAL TXTLIB FORTLIB
     FORTRAN &1
     FI 5 TERMINAL(RECFM F LRECL 80
     FI 6 TERMINAL(RECFM F LRECL 80
  CP SPOOL CONS START *
     LOAD &1 (START
  CP SPOOL CONS STOP CLOSE
     &TYPE ------- ------- ------- ------- ------- ---
     &TYPE TYPE 'CUADRO PROBLEM' TO ENTER FILENAME FILETYPE
     &TYPE ------- ------- ------- ------- ------- ---
     EXEC RECEIVE
     &TYPE ------- ------- ------- ------- ------- ---
```

```
     &TYPE DO YOU WANT A HARD COPY OF THE INFORMATION DISPLAYED  ?
     &TYPE -------- -------- -------- -------- -------- --------
     &TYPE (<CR> IF NO, ELSE 'YES')
     &READ VARS &ANSWER
     &IF .&ANSWER NE . &GOTO -YES
     ERASE &1 PROBLEM
     &EXIT
-YES PRINTOUT &1 PROBLEM
```

The file SETB99 TXTLIB (also known as "tape 99" or "buffer 99"), is a 132-byte array that exists in the region of core assigned to the programme. The space for this buffer is automatically created if the compiler detects the use of "tape unit 99".

The file CUADRO FORTRAN contains the source program of the package (see Appendix B).

To run the package, the user must type the command

```
cuadro cuadro
```

6.2 PREPARING THE DATA

The package is organized to handle any problem that is in the following standard form

$$\text{minimize} \quad \{ \ (1/2)x'Qx+b'x+c \mid Ax \le q \ , \ x \ge 0 \ \}$$

To illustrate how a problem can be put into standard form, the example discussed in section 3.2 is taken as model. Suppose the problem is given as minimize G

where

$$G = -6x_1 + 2x_1^2 - 2x_1x_2 + 2x_2^2$$

Subject to

$$x_1 + x_2 \le 2$$

$$x_1 \ , \ x_2 \ge 0$$

It can be easily verified that this is equivalent to minimize G

where

$$G = (1/2)\begin{bmatrix} x_1 & x_2 \end{bmatrix}\begin{bmatrix} 4 & -2 \\ -2 & 4 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} -6 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \end{bmatrix}$$

$$\text{Subject to} \quad \begin{bmatrix} 1 & 1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \le \begin{bmatrix} 2 \end{bmatrix} \quad , \quad x_1 \ , \ x_2 \ge 0$$

The values assigned by the user (see Appendix A) are as follows

$$N = 2 \qquad M = 1$$

$$VA = \begin{bmatrix} 4 \\ -2 \\ 4 \end{bmatrix} \qquad b = \begin{bmatrix} -6 \\ \\ 0 \end{bmatrix} \qquad c = [0]$$

$$VP = \begin{bmatrix} 1 \\ \\ 1 \end{bmatrix} \qquad q = [2]$$

Note that q must be strictly positive. If in a problem the standarized constraint part is not immediately available (by inspection), a basic starting solution must be searched. The technique discussed in section 2.3.1 can be used here, since the constraints part of the QPP has the same form as that of the linear programming problem.

6.3 MESSAGES

During execution of the package, various types of messages are transmitted to the user's terminal. These can be

1. Informative messages: where no answer is expected from the user; i.e. messages outlining the package, its outputs, etc.
2. Data requesting messages: where an answer (numerical data) is expected from the user; i.e. 'TYPE THE

VALUE OF N (I.E. THE NUMBER OF VARIABLES YOU HAVE)', 'TYPE THE VALUE OF YOUR m-VECTOR Q. (THE RIGHT-HAND SIDE VALUE IN YOUR CONSTRAINTS)', etc.

3. Control messages: where an answer 'yes'(hit return) or 'no'(type no) is expected from the user, example 'YOU HAVE n VARIABLES ? ', 'ARE THESE THE CORRECT VALUES OF YOUR m-VECTOR Q (THE RIGHT-HAND SIDE TERMS IN THE CONSTRAINTS) ? q q ... q ', etc.

4. Error messages: when the data is of a different kind than expected, the user is asked to input again, example

'---------> # ***** ERROR IN YOUR ENTRY : 12#456 , INPUT AGAIN OR TYPE Q TO QUIT'

'THE NUMBER OF CONSTRAINTS CANNOT BE NEGATIVE ! TRY AGAIN .. .

etc.

6.4 INTERPRETING THE RESULTS

If the value in the display option (DSPLAY) is 0, the package provides only the optimum value of the objective function together with the associated x-vector. If the choice is 1 the current value of the objective function a-

each iteration is provided, from the starting point to the final one.

When the choice of display is 2, the two main tableaux are displayed at each iteration. An example of such a display is shown in table 4

Indeed, the form of the tableaux used in Beale's method contains complete information about the problem at any step. The element in the upper left corner the objective function tableau (i.e. g(1,1)) contains the value of the objective function in that trial point. In the starting point, this corresponds to the value of c.

All the variables in the objective function tableau have a value of zero. All the variables in the constraint tableau have a nonzero value, given by the constant part in the constraint equation. However, if that variable has a subscript greater than N, then it is a slack variable (or a free variable) which has a zero coefficient in the objective function, hence it has no effect on the value of G.

The package contains a feature which permits the generation of a hard-copy of all information displayed on the screen during the computing sesion. This is achieved in the following way:

1. After the completion of the QPP solution, some information is displayed as the process of storing the solution file is progressing. The user is asked

TABLE 4

AN EXAMPLE OF TABLE OUTPUT

ITERATION NUMBER 2

OBJECTIVE FUNCTION

	1	2	3	4
	X 5	X 2	X 3	X 6
	-0.20740D 02	0.24800D 01	-0.22400D 01	0.17900D 02	0.13400D 01
X 5	0.24800D 01	0.12000D 00	-0.20000D 00	0.40000D 00	0.0
X 2	-0.22400D 01	-0.20000D 00	0.53200D 01	-0.22000D 01	0.48000D 00
X 3	0.17900D 02	0.40000D 00	-0.22000D 01	0.45000D 01	-0.30000D 00
X 6	0.13400D 01	0.0	0.48000D 00	-0.30000D 00	0.22000D 00

CONSTRAINTS

	X 5	X 2	X 3	X 6
X 1=	0.40000D 00	-0.20000D 00	0.0	-0.20000D 01	0.0
X 4=	0.60000D 00	0.0	-0.80000D 00	0.0	-0.20000D 00
X 7=	0.66667D 00	0.20000D 00	-0.20000D 00	0.10000D 01	0.20000D 00

THE CURRENT VALUE OF THE OBJECTIVE FUNCTION IS -20.74000000

THE VALUES OF THE VARIABLES ARE

X 1 = 0.40000000
X 4 = 0.60000000
X 7 = 0.66666700

ALL THE OTHERS ARE EQUAL TO ZERO

---

to type CUADFO PROGRAM as the FILENAME and FILETYPE

requested. If <CR> is made instead, no information is received.

2. The message 'DO YOU WANT A HARD-COPY OF THE INFORMATION DISPLAYED ?' (<CR> IF NOT, ELSE 'YES') is displayed and the desired answer is expected from the user.
3. If the answer is 'YES', the PRINTOUT command is sent, and the user is asked to type his batch account number and name.
4. If the answer is <CR> (no), the file CUADRO PROBLEM is erased from the user's disk.

Chapter VII

CONCLUSIONS

The interactive computing environment is acknowledged as a far superior problem solving environment than the batch mode. The intent of the work outlined in this thesis has been to provide such an environment for solving the quadratic programming problem. The package which has been developed makes possible a convenient and straight forward solution to problems of this class and in particular simplifies the process of specifying the problem parameters. Very little technical expertise is required of the user. Note also that the package is able to handle the situation where M=0; i.e. when the problem is unconstrained.

A highly modular organization has been followed in the development of the programme which should facilitate extensions or changes if these become required. The CUADRO package is a complete unit on its own . It could nevertheless be easily restructured to form part of some other package e.g. a general function minimization package. The calculations in the programme are carried out in double precision mode in order to minimize the influence of round-off errors. Inasmuch as FORTRAN IV was used as the programming language, extensive use of structured

programming concepts could not be employed; nevertheless an attempt was made to avoid excessive use of branching (GO TO) statements in order to improve the readability of the code.

At the present time, the dimensioning of various arrays in the program restricts the total number of constraints and variables to be less than 100. In order to handle larger problems, these arrays would have to be appropriately re-dimensioned.

Appendix A

AN EXAMPLE PROBLEM RUN WITH THE PACKAGE

```
    cuadro cuadro
GLOBAL TXTLIB FORTLIB
FORTRAN CUADRO
FI 5 TERMINAL (RECFM F LRECL 80
FI 6 TERMINAL (RECFM F LRECL 80
CP SPOOL CONS START *
LOAD CUADRO ( START
EXECUTION BEGINS...
```

UNIVERSITY OF OTTAWA
SYSTEMS SCIENCE
1982

CUADRO

THIS PACKAGE USES THE METHOD OF BEALE TO SOLVE THE QUADRATIC PROGRAMMING PROBLEM, HAVING N VARIABLES AND M CONSTRAINTS (M<=N), GIVEN IN ITS STANDARD FORM :

```
                MIN G = (1/2)X'AX + B'X + C
SUBJECT TO :      PX <= Q              ,   WITH  Q(J) > )
                  X(K) >= 0                  K = 1, 2, ... , N
                                             J = 1, 2, ... , M
```

WHERE : C : THE CONSTANT VALUE OF YOUR OBJECTIVE FUNCTION
         B : THE VECTOR OF THE LINEAR PART IN THE O. F.
         A : AN NXN MATRIX CONTAINING THE QUADRATIC COEFICIENTS OF THE O. F.
         P : AN NXM MATRIX WITH THE COEFICIENTS OF THE CONSTRAINTS
         Q : AN M-VECTOR WITH THE RIGHT-HAND SIDE VALUES OF THE CONSTRAINTS

NOTE THAT MATRIX A IS A SYMMETRIC MATRIX WITH A(I,J) =
 TWICE THE COEFICIENT OF X(K)**2 WHEN K=I=J (THE DIAGONAL OF THE MATRIX)
 COEF. OF X(I)X(J) WHEN I DIFFERENT FROM J (THE OFF-DIAGONAL ELEMENTS)

WHEN TYPING THE DATA REQUESTED, PLEASE RECALL :
*   AT LEAST ONE BLANK (SPACE) MUST SEPARATE SUCCESSIVE NUMBERS
*   THE DECIMAL POINT IS NOT NECESSARY, UNLESS IT IS PART OF THE NUMBER
*   IF YOU WANT TO QUIT TYPE "Q" INSTEAD OF THE INFORMATION REQUESTED
*   WHEN ANSWERING TRUE OR FALSE,
        TYPE "NO" AND <CR> IF FALSE, OR
        <CR> (HIT RETURN)   IF TRUE

```
 TYPE THE VALUE OF N (I.E. THE NUMBER OF VARIABLES YOU HAVE )
-3

 YOUR VALUE OF N = -3 IS NOT POSITIVE !
      RESTART . . .

 TYPE THE VALUE OF N (I.E. THE NUMBER OF VARIABLES YOU HAVE )
2

 TYPE THE VALUE OF M (I.E. THE NUMBER OF CONSTRAINTS YOU HAVE)
3

 THE STANDARDIZED PROBLEM REQUIRES  M <= N ... .
 IN YOUR PROBLEM, A MAXIMUM OF  2 CONSTRAINTS (M) WILL BE ACCEPTED

 TYPE THE VALUE OF M (I.E. THE NUMBER OF CONSTRAINTS YOU HAVE)
1

 THE QUADRATIC PART OF THE OBJECTIVE FUNCTION, BEING A SYMMETRIC MATRIX,
 ONLY THE UPPER-TRIANGULAR PART MUST BE TYPED

 TYPE THE  2 VALUES OF THE ROW NUMBER  1
4   -2

 TYPE THE VERY LAST ELEMENT OF YOUR MATRIX
 5

 TYPE THE  2-VECTOR B, (THE LINEAR PART OF YOUR OBJECTIVE FUNCTION)
-6  0

 TYPE THE VALUE OF C, I.E. THE CONSTANT PART IN YOUR OBJECTIVE FUNCTION
0
 ----------> 0
 ***** ERROR IN YOUR ENTRY :
0
 INPUT AGAIN OR TYPE Q TO QUIT
0

 TYPE THE  2 VALUES OF THE ROW NO.  1 IN YOUR CONSTRAINTS MATRIX B
  1     1.5

 TYPE THE VALUE OF Q IN YOUR CONSTRAINT (I.E., THE RIGHT-HAND SIDE VALUE)
2

 THE VALUES IN THE ZERO-BAND ARE TAKEN AS ZERO IN CRITICAL POINTS
 OF THE PROBLEM.
 TYPE THE VALUE OF THE ZERO-BAND FOR YOUR PROBLEM
1.d-7

 NOW, LET US CHECK THE CORRECTNESS OF THE DATA RECEIVED
     TYPE "NO" AND <CR>       IF FALSE, OR
     <CR>  (HIT RETURN)       IF TRUE
```

```
YOU HAVE  2 VARIABLES ?

YOU HAVE  1 CONSTRAINTS ?

ARE THESE THE CORRECT  2 ELEMENTS OF THE ROW NUMBER  1 OF MATRIX  A
IN YOUR OBJECTIVE FUNCTION ?
  4.0000 -2.0000

IS   5.0000 THE LAST ELEMENT OF THE MATRIX IN YOUR OBJECTIVE FUNCTION ?
NO

TYPE THE VERY LAST ELEMENT OF YOUR MATRIX
  4

IS   4.0000 THE LAST ELEMENT OF THE MATRIX IN YOUR OBJECTIVE FUNCTION ?

ARE THESE THE CORRECT VALUES OF YOUR  2-VECTOR B
(THE LINEAR PART IN YOUR OBJECTIVE FUNCTION)?
   -6.000     0.0

THE CONSTANT VALUE C IN YOUR OBJECTIVE FUNCTION IS   0.0    ?

ARE THESE THE CORRECT  2 VALUES OF ROW NUMBER  1
IN YOUR MATRIX P (CONSTRAINTS) ?
   1.0000    1.5000
NO

TYPE THE  2 VALUES OF THE ROW NO.  1 IN YOUR CONSTRAINTS MATRIX P
1    1

ARE THESE THE CORRECT  2 VALUES OF ROW NUMBER  1
IN YOUR MATRIX P (CONSTRAINTS) ?
   1.0000    1.0000

ARE THESE THE CORRECT VALUES OF YOUR  1-VECTOR Q
(THE RIGHT-HAND SIDE TERMS IN YOUR CONSTRAINTS) ?
   2.0000

THE WIDTH OF YOUR ZERO-BAND IS 0.0000001000 ?

SELECT ONE OF THE FOLLOWING OPTIONS :
0  >ONLY THE FINAL ANSWER WILL BE SHOWN
1  >THE VALUES OF THE OBJECTIVE FUNCTION AND THE BASIC VARIABLES WILL BE SHOWN
    AT EACH ITERATION
2  >THE COMPLETE TABLEAUX FOR THE METHOD OF SIMPLEX WILL BE DISPLAYED AT EACH
    ITERATION

TYPE THE NUMBER OF THE OPTION YOU CHOOSE
2
```

ORIGINAL TABLEAU

OBJECTIVE FUNCTION

	1	2
	X 1	X 2
	0.0	-0.30000D 01	0.0
1	-0.30000D 01	0.20000D 01	-0.10000D 01
2	0.0	-0.10000D 01	0.20000D 01

CONSTRAINTS

	X 1	X 2
3=	0.20000D 01	-0.10000D 01	-0.10000D 01

THE CURRENT VALUE OF THE OBJECTIVE FUNCTION IS 0.0

THE VALUES OF THE VARIABLES ARE

X 3 = 2.00000000

ALL THE OTHERS ARE EQUAL TO ZERO

```
                ITERATION NUMBER  1

    OBJECTIVE FUNCTION
                          1                2

                        X 4              X 2

    -0.45000D 01    0.0             -0.15000D 01

X 4  0.0            0.50000D 00      0.0

X 2 -0.15000D 01    0.0              0.15000D 01

    CONSTRAINTS

                        X 4              X 2

X 3= 0.50000D 00   -0.50000D 00    -0.15000D 01

X 1= 0.15000D 01    0.50000D 00     0.50000D 00

      THE CURRENT VALUE OF THE OBJECTIVE FUNCTION IS      -4.50000000

      THE VALUES OF THE VARIABLES ARE
                                X 3 =       0.50000000
                                X 1 =       1.50000000

      ALL THE OTHERS ARE EQUAL TO ZERO
```

ITERATION NUMBER 2

OBJECTIVE FUNCTION

	1	X 4	2 X 3
	-0.53333D 01	0.33333D 00	0.66667D 00
X 4	0.33333D 00	0.66667D 00	0.33333D 00
X 3	0.66667D 00	0.33333D 00	0.66667D 00

CONSTRAINTS

	X 4	X 3
X 2=	0.33333D 00	-0.33333D 00 -0.66667D 00
X 1=	0.16667D 01	0.33333D 00 -0.33333D 00

THE CURRENT VALUE OF THE OBJECTIVE FUNCTION IS -5.33333333

THE VALUES OF THE VARIABLES ARE

X 2 = 0.33333333
X 1 = 1.66666667

ALL THE OTHERS ARE EQUAL TO ZERO

ITERATION NUMBER 3

OBJECTIVE FUNCTION

	1	2
	X 4	X 3
	-0.55000D 01	0.0	0.50000D 00
X 4	0.0	0.15000D 01	0.0
X 3	0.50000D 00	0.0	0.50000D 00

CONSTRAINTS

	X 4	X 3
X 2=	0.50000D 00	-0.50000D 00	-0.50000D 00
X 1=	0.15000D 01	0.50000D 00	-0.50000D 00

THE CURRENT VALUE OF THE OBJECTIVE FUNCTION IS -5.50000000

THE VALUES OF THE VARIABLES ARE

X 2 = 0.50000000

X 1 = 1.50000000

ALL THE OTHERS ARE EQUAL TO ZERO

SOLUTION

OBJECTIVE FUNCTION

	1	2
	X 4	X 3
	-0.55000D 01	0.0 0.50000D 00
X 4	0.0	0.15000D 01 0.0
X 3	0.50000D 00	0.0 0.50000D 00

CONSTRAINTS

	X 4	X 3
X 2=	0.50000D 00	-0.50000D 00 -0.50000D 00
X 1=	0.15000D 01	0.50000D 00 -0.50000D 00

THE OPTIMAL IS -5.50000000

THE VALUES OF THE VARIABLES ARE

X 2= 0.50000000
X 1= 1.50000000

ALL THE OTHERS ARE EQUAL TO ZERO

```
CP SPOOL CONS STOP CLOSE
CON FILE 0463  TO  VWPSL
-------- -------- -------- -------- -------- ---
TYPE 'CUADRO PROBLEM' TO ENTER FILENAME FILETYPE
-------- -------- -------- -------- -------- ---
EXEC RECEIVE
RDR file 463 unnamed
Enter FILENAME FILETYPE for READ. <cr>=no RECEIVE
cuadro problem
RECORD LENGTH IS '132' BYTES.
File CUADRO PROBLEM A RECEIVEd from VWPSL
-------- -------- -------- -------- -------- -------
DO YOU WANT A HARD COPY OF THE INFORMATION DISPAYED ?
-------- -------- -------- -------- -------- -------
(<cr> IF NOT, ELSE YES)
yes
PRINTOUT CUADRO PROBLEM
batch account number vwpslxxx
programmer name oscar
PUN FILE 0464 TO OSVS1  COPY=0001  RECDS=000318  NO HOLD
R; T=3.38/4.45 13:06:16
```

Appendix B

THE FORTRAN PROGRAMS

```
      CALL DIALOG                                                       CUA00010
      CALL INITIA                                                       CUA00020
      CALL SETMTX                                                       CUA00030
      CALL SHOWIT                                                       CUA00040
      CALL RUNIT                                                        CUA00050
      STOP                                                              CUA00060
      END                                                               CUA00070
                                                                        CUA00080
                                                                        CUA00090
      SUBROUTINE RUNIT                                                  CUA00100
      IMPLICIT REAL*8 (A-H,O-Z)                                         CUA00110
      INTEGER BASIC0,BASIC1,PL,ALTBEF,S,T,V                             CUA00120
      REAL*8 MAX                                                        CUA00130
      LOGICAL PICO                                                      CUA00140
      COMMON /NUMERO/ N,M                                               CUA00150
      COMMON /SUBJCT/ P(200)                                            CUA00160
      COMMON /LIBERS/ LIBR(50),LIBRE,ALIBER(50)                         CUA00170
      COMMON /BASICA/ BASIC0(50),BASIC1(50)                             CUA00180
      COMMON /VALUES/ EPS,MIC,TTTP,DSPLAY,KIM                           CUA00190
      COMMON /MAXIMO/ MAX                                               CUA00200
      COMMON /MINIMO/ YO                                                CUA00210
      COMMON /PARAME/ V,T,F                                             CUA00220
      COMMON /PIVOTE/ IPIV,KPIV                                         CUA00230
      COMMON /KNUMER/ KCO,KVA                                           CUA00240
      COMMON /LENGTH/ PL,JC,J1                                          CUA00250
      COMMON /LOGICO/ PICO                                              CUA00260
      COMMON /FINAL / ISDONE                                            CUA00270
      ISDONE = 1                                                        CUA00280
1110  IF(LIBRE.EQ.0)GO TO 1120                                          CUA00290
      CALL MALIB                                                        CUA00300
      IF(DABS(MAX).GT.EPS)GO TO 1200                                    CUA00310
1120  IF(KVA.LE.0)GO TO 1130                                            CUA00320
      CALL COLPIV                                                       CUA00330
      IF(YO.LT.-EPS)GO TO 1200                                          CUA00340
1130  ISDONE=1                                                          CUA00350
      GO TO 1990                                                        CUA00360
1200  CALL SETV                                                         CUA00370
      CALL RATIO                                                        CUA00380
      IF(IPIV.NE.0.OR.MAX.NE.0)GO TO 1300                               CUA00390
      ISDONE=0                                                          CUA00400
      GO TO 1990                                                        CUA00410
1300  IF(IPIV.EQ.0.OR.YO.GT.DABS(MAX))GO TO 1400                        CUA00420
      I=BASIC0(KPIV)                                                    CUA00430
      BASIC0(KPIV)=BASIC1(IPIV)                                         CUA00440
      BASIC1(IPIV)=I                                                    CUA00450
      IF=R                                                              CUA00460
```

```
      P=1                                                               CUA00470
      CALL CONACT                                                       CUA00480
      IF(ISDONE.EQ.0)GO TO 1990                                         CUA00490
      P=IP                                                              CUA00500
      PICO=.TRUE.                                                       CUA00510
      CALL OFUACT                                                       CUA00520
      GO TO 1600                                                        CUA00530
1400  ALIBPE(KPIV)=ALIBPE(KPIV)+1                                       CUA00540
      I=0                                                               CUA00550
      P=0                                                               CUA00560
1410  IF(LIBPE.EQ.0)GO TO 1500                                          CUA00570
      DO 16 I=1,LIBPE                                                   CUA00580
         IF(LIBP(I).EQ.KPIV)GO TO 1430                                  CUA00590
16    CONTINUE                                                          CUA00600
      GO TO 1500                                                        CUA00610
1430  CALL ADDONE                                                       CUA00620
      CALL CONACT                                                       CUA00630
      IF(ISDONE.EQ.0)GO TO 1990                                         CUA00640
      PICO=.FALSE.                                                      CUA00650
      CALL OFUACT                                                       CUA00660
      CALL SHOWIT                                                       CUA00670
      GO TO 1110                                                        CUA00680
1500  CALL CHANGE                                                       CUA00690
      GO TO 1410                                                        CUA00700
1600  IF(BASIC1(IPIV).LE.N+MIO)GO TO 1650                               CUA00710
      CALL DELETE                                                       CUA00720
1650  CALL SHOWIT                                                       CUA00730
      GO TO 1110                                                        CUA00740
1990  CALL OPTIMO                                                       CUA00750
      RETURN                                                            CUA00760
      END                                                               CUA00770
                                                                        CUA00780
                                                                        CUA00790
      SUBROUTINE INITIA                                                 CUA00800
      IMPLICIT REAL*8 (A-H,O-Z)                                         CUA00810
      INTEGER BASIC0,BASIC1,ALIBPE,PL                                   CUA00820
      COMMON /NUMERO/ N,M                                               CUA00830
      COMMON /BASICA/ BASIC0(50),BASIC1(50)                             CUA00840
      COMMON /LIBRIS/ LIBR(50),LIBPE,ALIBP(50)                          CUA00850
      COMMON /KNUMER/ KCO,KVA                                           CUA00860
      COMMON /LENGTH/ PL,JC,J1                                          CUA00870
      COMMON /VALUES/ EPS,MIO,ITER,DSPLAY,KTM                           CUA00880
      COMMON /OTROS / LOS(50),LAS(50)                                   CUA00890
      JC=((N+1)*(N+2))/2                                                CUA00900
      J1=(M+N+2)*(N+1)                                                  CUA00910
      EPS = DABS(EPS/2.D0)                                              CUA00920
      NI=N+1                                                            CUA00930
      DO 10 I=1,NI                                                      CUA00940
         BASIC0(I)=I                                                    CUA00950
   10 CONTINUE                                                          CUA00960
      DO 13 K=1,N                                                       CUA00970
         LOS(K)=K                                                       CUA00980
```

```
      ALIBRF(K)=0                                                       CUA00990
      LIBP(K)=K                                                         CUA01000
   13 CONTINUE                                                          CUA01010
      IF(M.EQ.0)GO TO 1102                                              CUA01020
      DO 14 I=1,M                                                       CUA01030
         LAS(I)=I                                                       CUA01040
         BASIC1(I)=N+I                                                  CUA01050
   14 CONTINUE                                                          CUA01060
 1102 KVA=N                                                             CUA01070
      KCO=M                                                             CUA01080
      IL=N+1                                                            CUA01090
      KIM=N+M                                                           CUA01100
      ITER=0                                                            CUA01110
      LIBRF=0                                                           CUA01120
      MIO=M                                                             CUA01130
      RETURN                                                            CUA01140
      END                                                               CUA01150
                                                                        CUA01160
                                                                        CUA01170
                                                                        CUA01180
      SUBROUTINE SETMTX                                                 CUA01190
      IMPLICIT REAL*8 (A-H,O-Z)                                         CUA01200
      DIMENSION G(200),F(100)                                           CUA01210
      COMMON /NUMERO/ N,M                                               CUA01220
      COMMON /SUBJCT/ P(200),Q(50)                                      CUA01230
      COMMON /CFUNCN/ A(100),B(50),C                                    CUA01240
      KM=(N+1)*(N+2)/2                                                  CUA01250
      LM=(N+1)*N/2                                                      CUA01260
      H(1) = C                                                          CUA01270
      JH = 2                                                            CUA01280
      DO 15 I=1,N                                                       CUA01290
         H(JH) = B(I)/2                                                 CUA01300
         JH = JH + 1                                                    CUA01310
   15 CONTINUE                                                          CUA01320
      DO 16 J=1,LM                                                      CUA01330
         H(JH)=A(J)/2                                                   CUA01340
         JH=JH+1                                                        CUA01350
   16 CONTINUE                                                          CUA01360
      DO 17 K=1,KM                                                      CUA01370
         A(K)=H(K)                                                      CUA01380
   17 CONTINUE                                                          CUA01390
      IT=(M+1)*(N+1)                                                    CUA01400
      NI=N+1                                                            CUA01410
      DO 11 K=1,NI                                                      CUA01420
         G(K)=0.D0                                                      CUA01430
   11 CONTINUE                                                          CUA01440
      IN=N                                                              CUA01450
      I1=1                                                              CUA01460
      DO 19 J=1,M                                                       CUA01470
         NI=NI+1                                                        CUA01480
         G(NI)=Q(J)                                                     CUA01490
         DO 20 IJ=I1,IN                                                 CUA01500
```

```
          NI=NI+1                                                    CUA01510
          G(NI)=-P(IJ)                                               CUA01520
20      CONTINUE                                                     CUA01530
        I1=IN+1                                                      CUA01540
        IN=IN+N                                                      CUA01550
19    CONTINUE                                                       CUA01560
      DO 21 K=1,IT                                                   CUA01570
        P(K)=G(K)                                                    CUA01580
21    CONTINUE                                                       CUA01590
      RETURN                                                         CUA01600
      END                                                            CUA01610
                                                                     CUA01620
                                                                     CUA01630
      SUBROUTINE SETV                                                CUA01640
      IMPLICIT REAL*8 (A-H,O-Z)                                      CUA01650
      REAL*8 MAX                                                     CUA01660
      INTEGER V                                                      CUA01670
      COMMON /PIVOTE/ IPIV,KPIV                                      CUA01680
      COMMON /VALUES/ EPS                                            CUA01690
      COMMON /NUMERO/ N,M                                            CUA01700
      COMMON /OFUNCN/ A(100)                                         CUA01710
      COMMON /MAXIMO/ MAX                                            CUA01720
      COMMON /PARAME/ V                                              CUA01730
      KI=KPIV*(N+1)-((KPIV-1)*KPIV)/2+1                              CUA01740
      MAX=0.D0                                                       CUA01750
      IF(A(KI).GT.EPS)MAX=A(KPIV+1)/A(KI)                            CUA01760
      V=-1                                                           CUA01770
      IF(MAX.GT.0.D0)V=1                                             CUA01780
      RETURN                                                         CUA01790
      END                                                            CUA01800
                                                                     CUA01810
                                                                     CUA01820
      SUBROUTINE ADIONE                                              CUA01830
      IMPLICIT REAL*8 (A-H,O-Z)                                      CUA01840
      INTEGER T,BL,Z                                                 CUA01850
      COMMON /PIVOTE/ IPIV,KPIV                                      CUA01860
      COMMON /LENGTH/ BL                                             CUA01870
      COMMON /NUMERO/ N,M                                            CUA01880
      COMMON /PARAME/ V,T                                            CUA01890
      COMMON /SUBJCT/ P(200)                                         CUA01900
      COMMON /OFUNCN/ A(100)                                         CUA01910
      Z=0                                                            CUA01920
      KAP=KPIV+1                                                     CUA01930
      DO 17 K=1,KAP                                                  CUA01940
        JO=K-1                                                       CUA01950
        KU=T*BL+JO+1                                                 CUA01960
        AJO=KPIV+Z+1                                                 CUA01970
        P(KU)=A(AJO)                                                 CUA01980
17    Z=Z+N-JO                                                       CUA01990
      NIL=N-KPIV                                                     CUA02000
      IF(NIL.LT.1)GO TO 1450                                         CUA02010
      DO 18 K=1,NIL                                                  CUA02020
```

```
      KO=T*FL+(KPIV+K)+1                                                CUA02030
      KOJ=KPIV*(N+1)-(KPIV*(KPIV-1))/2+K+1                              CUA02040
   18 P(KO)=A(KOJ)                                                      CUA02050
 1450 IPIV=T                                                            CUA02060
      RETURN                                                            CUA02070
      END                                                               CUA02080
                                                                        CUA02090
                                                                        CUA02100
      SUBROUTINE CHANGE                                                 CUA02110
      IMPLICIT REAL*8 (A-H,O-Z)                                         CUA02120
      INTEGER T,P,BASIC1,BASIC0                                         CUA02130
      COMMON /KNUMBE/ KCO,KVA                                           CUA02140
      COMMON /PARAME/ V,T,P                                             CUA02150
      COMMON /LIBRES/ LIBR(50),LIBRE                                    CUA02160
      COMMON /PIVOTI/ IPIV,KPIV                                         CUA02170
      COMMON /NUMERO/ N,M                                               CUA02180
      COMMON /BASICA/ BASIC0(50),BASIC1(50)                             CUA02190
      COMMON /OTROS / LOS(50),LAS(50)                                   CUA02200
      COMMON /VALUES/ EPS,MIO                                           CUA02210
      LIBRE=LIBRE+1                                                     CUA02220
      LIBR(LIBRE)=KPIV                                                  CUA02230
      BASIC1(M+1)=BASIC0(KPIV)                                          CUA02240
      BASIC0(KPIV)=N+MIO+KPIV                                           CUA02250
      LAS(M+1)=M+1                                                      CUA02260
      KCO=M+1                                                           CUA02270
      IF(KVA.LT.1)GO TO 1540                                            CUA02280
      DO 19 K=1,KVA                                                     CUA02290
         IF(LOS(K).EQ.KPIV) T=K                                         CUA02300
   19 CONTINUE                                                          CUA02310
      KVA=KVA-1                                                         CUA02320
 1540 IF(KVA.LT.T.OR.T.LE.0)GO TO 1560                                  CUA02330
      DO 20 K=T,KVA                                                     CUA02340
         LOS(K)=LOS(K+1)                                                CUA02350
   20 CONTINUE                                                          CUA02360
 1560 M=M+1                                                             CUA02370
      T=M                                                               CUA02380
      P=1                                                               CUA02390
      RETURN                                                            CUA02400
      END                                                               CUA02410
                                                                        CUA02420
                                                                        CUA02430
      SUBROUTINE DELETE                                                 CUA02440
      IMPLICIT REAL*8 (A-H,O-Z)                                         CUA02450
      INTEGER FL,ALIBRE,BASIC1                                          CUA02460
      COMMON /NUMERO/ N,M                                               CUA02470
      COMMON /KNUMBE/ KCO,KVA                                           CUA02480
      COMMON /BASICA/ KBAO(50),BASIC1(50)                               CUA02490
      COMMON /PIVOTE/ IPIV,KPIV                                         CUA02500
      COMMON /LIBRES/ LIBR(50),LIBRE,ALIBR(50)                          CUA02510
      COMMON /OTROS / LOS(50)                                           CUA02520
      COMMON /LENGTH/ FL                                                CUA02530
      COMMON /SUBJCT/ P(200)                                            CUA02540
```

```
      BASIC1(IPIV)=BASIC1(M)                                            CUA02550
      KCO=KCO-1                                                         CUA02560
      KVA=KVA+1                                                         CUA02570
      LOS(KVA)=KPIV                                                     CUA02580
      LIBRE=LIBRE-1                                                     CUA02590
      IF(LIBRE.LT.1)GO TO 1640                                          CUA02600
      DO 21 K=1,LIBRE                                                   CUA02610
         IF(LIBR(K).EQ.KPIV)GO TO 1630                                  CUA02620
   21 CONTINUE                                                          CUA02630
      GO TO 1640                                                        CUA02640
 1630 DO 22 I=K,LIBRE                                                   CUA02650
         LIBR(I)=LIBR(I+1)                                              CUA02660
   22 CONTINUE                                                          CUA02670
 1640 NI=N+1                                                            CUA02680
      KIE=IPIV*PL+1                                                     CUA02690
      KIA=M*PL+1                                                        CUA02700
      DO 23 K=1,NI                                                      CUA02710
         KII=K-1                                                        CUA02720
         KIU=KIE+KII                                                    CUA02730
         KIO=KIA+KII                                                    CUA02740
         P(KIU)=P(KIO)                                                  CUA02750
   23 CONTINUE                                                          CUA02760
      M=M-1                                                             CUA02770
      ALIBRE(KPIV)=ALIBRE(KPIV)-1                                       CUA02780
      RETURN                                                            CUA02790
      END                                                               CUA02800
                                                                        CUA02810
                                                                        CUA02820
      SUBROUTINE PROFUN                                                 CUA02830
      IMPLICIT REAL*8 (A-H,O-Z)                                         CUA02840
      DIMENSION X(100)                                                  CUA02850
      INTEGER BO                                                        CUA02860
      COMMON /OFUNCN/ A(100)                                            CUA02870
      COMMON /NUMERO/ N                                                 CUA02880
      COMMON /BASICA/ BO(50)                                            CUA02890
      COMMON /VALUES/ EPS,MIC,ITER                                      CUA02900
      IF(ITER.EQ.0)CALL MESSAG(15)                                      CUA02910
      L=2*N+1                                                           CUA02920
      NI=N+1                                                            CUA02930
      NN=N+1                                                            CUA02940
      IF(NI.GT.5)NI=5                                                   CUA02950
      NO=NI-1                                                           CUA02960
      WRITE(6,214)(K,K=1,NO)                                            CUA02970
      WRITE(6,215)(BO(K),K=1,NO)                                        CUA02980
      DO 216 I=1,NN                                                     CUA02990
         DO 217 K=1,NI                                                  CUA03000
            IF(K.GE.I)GO TO 219                                         CUA03010
            K1=(L-K+1)*(K-1)/2+I                                        CUA03020
            GO TO 217                                                   CUA03030
  219       K1=(L-I+1)*(I-1)/2+K                                        CUA03040
  217       X(K)=A(K1)                                                  CUA03050
         IF(I.NE.1)GO TO 280                                            CUA03060
```

```
240      WRITE(6,220) (X(K),K=1,NI)                                    CUA03070
         GO TO 216                                                     CUA03080
280      WRITE(6,222) B0(I-1),(X(K),K=1,NI)                            CUA03090
216   CONTINUE                                                         CUA03100
285   IF(NO.EQ.N)RETURN                                                CUA03110
      NI=NO+1                                                          CUA03120
      NO=N                                                             CUA03130
      WRITE(6,214) (K,K=NI,NO)                                         CUA03140
      WRITE(6,215) (B0(K),K=NI,NO)                                     CUA03150
      DO 226 I=1,NN                                                    CUA03160
         DO 227 K=NI,NO                                                CUA03170
            K1=K+1                                                     CUA03180
            IF(K1.LT.I) GO TO 228                                      CUA03190
            KI=(L-I+1)*(I-1)/2+K1                                      CUA03200
            GO TO 227                                                  CUA03210
228         KI=(L-K1+1)*(K1-1)/2+I                                     CUA03220
227      X(K)=A(KI)                                                    CUA03230
         IF(I.EQ.1) WRITE(6,230) (X(J),J=NI,NO)                        CUA03240
         IF(I.EQ.1) GO TO 226                                          CUA03250
         WRITE(6,232) B0(I-1),(X(J),J=NI,NO)                           CUA03260
226   CONTINUE                                                         CUA03270
250   RETURN                                                           CUA03280
214   FORMAT(/,5X,'OBJECTIVE FUNCTION',/,11X,4(12X,I2))                CUA03290
215   FORMAT(/,23X,4('X',I2,11X))                                      CUA03300
220   FORMAT(/,3X,5(2X,D12.5))                                         CUA03310
222   FORMAT(/,1X,'X',I2,1X,D12.5,4(2X,D12.5))                         CUA03320
230   FORMAT(/,17X,4(2X,D12.5))                                        CUA03330
232   FORMAT(/,1X,'X',I2,13X,4(2X,D12.5))                              CUA03340
      END                                                              CUA03350
                                                                       CUA03360
                                                                       CUA03370
      SUBROUTINE PRICON                                                CUA03380
      IMPLICIT REAL*8 (A-H,O-Z)                                        CUA03390
      INTEGER B0,B1,PL                                                 CUA03400
      COMMON /NUMERO/ N,M                                              CUA03410
      COMMON /BASIC/ B0(50),B1(50)                                     CUA03420
      COMMON /SUBJCT/ P(200)                                           CUA03430
      COMMON /LENGTH/ PL                                               CUA03440
      N1=N+1                                                           CUA03450
      M1=M+1                                                           CUA03460
      IF(N1.GT.5)N1=5                                                  CUA03470
      NO=N1-1                                                          CUA03480
      CALL MESSAG(16)                                                  CUA03490
      WRITE(6,307) (B0(K),K=1,NO)                                      CUA03500
      DO 306 I=2,M1                                                    CUA03510
         KI=(I-1)*PL+1                                                 CUA03520
         KO=KI+NO                                                      CUA03530
         WRITE(6,309) B1(I-1),(P(K),K=KI,KO)                           CUA03540
306   CONTINUE                                                         CUA03550
      IF(NO.EQ.N)RETURN                                                CUA03560
320   NI=NO+1                                                          CUA03570
      NO=N                                                             CUA03580
```

```
      IF(NO-NI.GT.5)NO=NI+5                                             CUA03590
      WRITE(6,307)(B0(K),K=NI,NO)                                       CUA03600
      DO 310 I=2,M1                                                     CUA03610
         K1=(I-1)*RI+NI+1                                               CUA03620
         K2=K1+NO-NI                                                    CUA03630
         WRITE(6,311)B1(I-1),(P(K),K=K1,K2)                             CUA03640
  310 CONTINUE                                                          CUA03650
      IF(NO.EQ.N)RETURN                                                 CUA03660
      GO TO 320                                                         CUA03670
  307 FORMAT(/,23X,4('X',I2,11X))                                       CUA03680
  309 FORMAT(/,1X,'X',I2,'=',D12.5,4(2X,D12.5))                         CUA03690
  311 FORMAT(/,1X,'X',I2,13X,4(2X,D12.5))                               CUA03700
      END                                                               CUA03710
                                                                        CUA03720
                                                                        CUA03730
      SUBROUTINE MALIB                                                  CUA03740
      IMPLICIT REAL*8 (A-H,O-Z)                                         CUA03750
      REAL*8 MAX                                                        CUA03760
      COMMON /OFUNCN/ A(100)                                            CUA03770
      COMMON /LIBRES/ LIBF(50),LIBPF                                    CUA03780
      COMMON /NUMERO/ N,M                                               CUA03790
      COMMON /PIVOTE/ KDUM,KPIV                                         CUA03800
      COMMON /MAXINO/ MAX                                               CUA03810
      KPIV=LIBF(1)                                                      CUA03820
      KOH=LIBF(1)+1                                                     CUA03830
      MAX=A(KOH)                                                        CUA03840
      IF(LIBFE.LT.2)RETURN                                              CUA03850
      DO 402 K=2,LIBFE                                                  CUA03860
         IF(MAX.EQ.0)GO TO 402                                          CUA03870
         KOH=LIBF(K)+1                                                  CUA03880
         IF(A(KOH).EQ.0)GO TO 402                                       CUA03890
         IF(DABS(MAX).GT.DABS(A(KOH)))GO TO 402                         CUA03900
         KPIV=LIBF(K)                                                   CUA03910
         MAX=A(KOH)                                                     CUA03920
  402 CONTINUE                                                          CUA03930
      RETURN                                                            CUA03940
      END                                                               CUA03950
                                                                        CUA03960
                                                                        CUA03970
      SUBROUTINE OFUACT                                                 CUA03980
      IMPLICIT REAL*8 (A-H,O-Z)                                         CUA03990
      INTEGER FL,I,Z,Z1                                                 CUA04000
      LOGICAL PICO                                                      CUA04010
      COMMON /SUBJCT/ P(200)                                            CUA04020
      COMMON /OFUNCN/ A(100)                                            CUA04030
      COMMON /LENGTH/ RI                                                CUA04040
      COMMON /PIVOTE/ IP,KP                                             CUA04050
      COMMON /NUMERO/ N,M                                               CUA04060
      COMMON /LOGICO/ PICO                                              CUA04070
      IN=N+1                                                            CUA04080
      DO 502 IP=1,IN                                                    CUA04090
         J=IP-1                                                         CUA04100
```

```
      IF(J.EQ.KP)GO TO 502                                              CUA04110
      Z=0                                                               CUA04120
      IF(J.GT.KP)Z1=KP-1                                                CUA04130
      IF(J.LT.KP)Z1=J                                                   CUA04140
      IZ1=Z1+1                                                          CUA04150
      DO 504 IS=1,IZ1                                                   CUA04160
         K=IS-1                                                         CUA04170
         KH=J+Z+1                                                       CUA04180
         KH1=IP*RL+J+1                                                  CUA04190
         KH0=KP+Z+1                                                     CUA04200
         A(KH)=A(KH)+P(KH1)*A(KH0)                                      CUA04210
504   Z=Z+N-K                                                           CUA04220
      T=Z+KP                                                            CUA04230
502 CONTINUE                                                            CUA04240
506 IKP=KP+1                                                            CUA04250
    IF(IKP.GT.N)GO TO 530                                               CUA04260
    DO 510 K=IKP,N                                                      CUA04270
       DO 512 J=K,N                                                     CUA04280
          KH=Z+N-KP+J+1                                                 CUA04290
          KH1=IP*RL+J+1                                                 CUA04300
          KH0=T+K-KP+1                                                  CUA04310
512    A(KH)=A(KH)+P(KH1)*A(KH0)                                        CUA04320
510 Z=Z+N-K                                                             CUA04330
    DO 522 J=IKP,N                                                      CUA04340
       KH=T+J-KP+1                                                      CUA04350
       KH1=IP*RL+J+1                                                    CUA04360
522 A(KH)=A(KH)+P(KH1)*A(T+1)                                           CUA04370
530 Z1=0                                                                CUA04380
    DO 532 IR=1,KP                                                      CUA04390
       J=IR-1                                                           CUA04400
       Z=0                                                              CUA04410
       KH=KP+Z1+1                                                       CUA04420
       KH1=IP*RL+J+1                                                    CUA04430
       SUMA=A(KH)+P(KH1)*A(T+1)                                         CUA04440
       DO 534 IS=1,IR                                                   CUA04450
          K=IS-1                                                        CUA04460
          KH=IP*RL+K+1                                                  CUA04470
          KH1=IR+Z                                                      CUA04480
          A(KH1)=A(KH1)+P(KH)*SUMA                                      CUA04490
534    Z=Z+N-K                                                          CUA04500
532 Z1=Z1+N-J                                                           CUA04510
    IF(IKP.GT.N)GO TO 550                                               CUA04520
    DO 542 J=IKP,N                                                      CUA04530
       Z=0                                                              CUA04540
       IR=J+1                                                           CUA04550
       DO 542 IS=1,IR                                                   CUA04560
          K=IS-1                                                        CUA04570
          IF(K.NE.KP)GO TO 546                                          CUA04580
          Z=Z+1                                                         CUA04590
          GO TO 542                                                     CUA04600
546       KH=J+Z                                                        CUA04610
          KH0=T+J-KP+1                                                  CUA04620
```

```
            KH1=IP*RL+K+1                                                   CUA04630
            A(KH)=A(KH)+P(KH1)*A(KHO)                                       CUA04640
542   Z=Z+N-K                                                               CUA04650
550   IF(.NOT.PTCO) GO TO 560                                               CUA04660
      Z=0                                                                   CUA04670
      KHO=IP*RL+KP+1                                                        CUA04680
      DO 552 IS=1,KP                                                        CUA04690
         K=IS-1                                                             CUA04700
         KH=IKP+Z                                                           CUA04710
         KH1=IP*RL+K+1                                                      CUA04720
         A(KH)=A(KH)+P(KH1)*A(I+                                            CUA04730
552   Z=Z+N-K                                                               CUA04740
      DO 554 J=KP,N                                                         CUA04750
         KH=J+I-KP+1                                                        CUA04760
554   A(KH)=A(KH)*P(KHO)                                                    CUA04770
      Z=0                                                                   CUA04780
      DO 556 IS=1,IKP                                                       CUA04790
         K=IS-1                                                             CUA04800
         KH=IKP+Z                                                           CUA04810
         A(KH)=A(KH)*P(KHO)                                                 CUA04820
556   Z=Z+N-K                                                               CUA04830
      RETURN                                                                CUA04840
560   Z=0                                                                   CUA04850
      DO 562 IS=1,KP                                                        CUA04860
         K=IS-1                                                             CUA04870
         KH=KI+Z+1                                                          CUA04880
         A(KH)=0.D0                                                         CUA04890
562   Z=Z+N-K                                                               CUA04900
      A(I+1)=1.D0/A(I+1)                                                    CUA04910
      IF(IKP.GT.N)GO TO 564                                                 CUA04920
      DO 568 J=IKP,N                                                        CUA04930
         KH=J+I-KP+1                                                        CUA04940
568   A(KH)=0.D0                                                            CUA04950
564   RETURN                                                                CUA04960
      END                                                                   CUA04970
                                                                            CUA04980
                                                                            CUA04990
      SUBROUTINE COLPIV                                                     CUA05000
      IMPLICIT REAL*8 (A-H,O-Z)                                             CUA05010
      COMMON /OFUNCN/ A(100)                                                CUA05020
      COMMON /OTROS / LOS(50)                                               CUA05030
      COMMON /KNUMER/ KCO,KVA                                               CUA05040
      COMMON /NUMERO/ N,M                                                   CUA05050
      COMMON /MINIMO/ YO                                                    CUA05060
      COMMON /PIVOTT/ IPIV,KP                                               CUA05070
      KP=LOS(1)                                                             CUA05080
      KHO=LOS(1)+1                                                          CUA05090
      YO=A(KHO)                                                             CUA05100
      IF(KVA.LT.2)RETURN                                                    CUA05110
      DO 602 K=2,KVA                                                        CUA05120
         KHO=LOS(K)+1                                                       CUA05130
         IF(A(KHO).GE.YO)GO TO 602                                          CUA05140
```

```
      YO=P(KH0)                                                         CUA05150
      KP=LOS(K)                                                         CUA05160
602 CONTINUE                                                            CUA05170
    RETURN                                                              CUA05180
    END                                                                 CUA05190
                                                                        CUA05200
                                                                        CUA05210
    SUBROUTINE RATIO                                                    CUA05220
    IMPLICIT REAL*8 (A-H,O-Z)                                           CUA05230
    INTEGER RL,V,Z                                                      CUA05240
    COMMON /LENGTH/ PI                                                  CUA05250
    COMMON /PIVOTE/ IP,KP                                               CUA05260
    COMMON /SUBJCT/ P(200)                                              CUA05270
    COMMON /OTROS / KLOS(50),LAS(50)                                    CUA05280
    COMMON /KNUMER/ KCO                                                 CUA05290
    COMMON /VALUES/ EPS,MIO,ITER,DSPLAY,KIM                             CUA05300
    COMMON /MINIMO/ YO                                                  CUA05310
    COMMON /PARAME/ V                                                   CUA05320
    COMMON /NUMERO/ N,M                                                 CUA05330
    IP=0                                                                CUA05340
    IF(KCO.LT.1)RETURN                                                  CUA05350
    DO 702 I=1,KCO                                                      CUA05360
       KH=LAS(I)*RL+1                                                   CUA05370
       KH1=KH+KP                                                        CUA05380
       IF(V*P(KH1).GT.EPS)GO TO 704                                     CUA05390
702 CONTINUE                                                            CUA05400
    RETURN                                                              CUA05410
704 YO=V*P(KH)/P(KH1)                                                   CUA05420
    IP=LAS(I)                                                           CUA05430
    Z=I+1                                                               CUA05440
    IF(Z.GT.KCO)RETURN                                                  CUA05450
    DO 706 I=Z,KCO                                                      CUA05460
       KH=LAS(I)*RL+1                                                   CUA05470
       KH1=KH+KP                                                        CUA05480
       IF(V*P(KH1).LE.EPS)GO TO 706                                     CUA05490
       YE=V*P(KH)/P(KH1)                                                CUA05500
       IF(YE.GE.YO)GO TO 708                                            CUA05510
       IP=LAS(I)                                                        CUA05520
       YO=YE                                                            CUA05530
       GO TO 706                                                        CUA05540
708    IF(YE.NE.YO)GO TO 706                                            CUA05550
       IO=LAS(I)                                                        CUA05560
       DO 710 K=1,N                                                     CUA05570
          KH0=IP*RL+K+1                                                 CUA05580
          KH2=IP*RL+KP+1                                                CUA05590
          KH=IO*RL+K+1                                                  CUA05600
          YEP=V*P(KH0)/P(KH2)                                           CUA05610
          YE0=V*P(KH)/P(KH1)                                            CUA05620
          IF(YEP.LT.YE0)GO TO 706                                       CUA05630
          IF(YE0.LT.YEP)GO TO 712                                       CUA05640
710    CONTINUE                                                         CUA05650
712    IP=IO                                                            CUA05660
```

```
706 CONTINUE                                                          CUA05670
    RETURN                                                            CUA05680
    END                                                               CUA05690
                                                                      CUA05700
                                                                      CUA05710
    SUBROUTINE CONACT                                                 CUA05720
    IMPLICIT REAL*8 (A-H,O-Z)                                         CUA05730
    INTEGER RL,P                                                      CUA05740
    COMMON /SUBJCT/ P(200)                                            CUA05750
    COMMON /PARAME/ V,T,I0                                            CUA05760
    COMMON /NUMERO/ K1,I1                                             CUA05770
    COMMON /PIVOTE/ IP,KP                                             CUA05780
    COMMON /LENGTH/ RL                                                CUA05790
    COMMON /FINAL / ISDONE                                            CUA05800
    KH=IP*RL+KP+1                                                     CUA05810
    IF(P(KH).EQ.0)GO TO 820                                           CUA05820
    PIVOTA=1.D0/P(KH)                                                 CUA05830
    II0=I0+1                                                          CUA05840
    II1=I1+1                                                          CUA05850
    KK0=1                                                             CUA05860
    KK1=K1+1                                                          CUA05870
    IF(II0.GT.II1)GO TO 802                                           CUA05880
    DO 804 II=II0,II1                                                 CUA05890
       I=II-1                                                         CUA05900
       IF(I.EQ.IP)GO TO 804                                           CUA05910
       KH0=I*RL+KP+1                                                  CUA05920
       P(KH0)=P(KH0)*PIVOTA                                           CUA05930
       DO 806 KK=KK0,KK1                                              CUA05940
          K=KK-1                                                      CUA05950
          IF(K.EQ.KP)GO TO 806                                        CUA05960
          KH1=I*RL+K+1                                                CUA05970
          KH2=IP*RL+K+1                                               CUA05980
          P(KH1)=P(KH1)-P(KH2)*P(KH0)                                 CUA05990
806    CONTINUE                                                       CUA06000
804 CONTINUE                                                          CUA06010
    DO 810 KK=KK0,KK1                                                 CUA06020
       K=KK-1                                                         CUA06030
       KH2=IP*RL+K+1                                                  CUA06040
810 IF(K.NE.KP)P(KH2)=-P(KH2)*PIVOTA                                  CUA06050
802 P(KH)=PIVOTA                                                      CUA06060
    RETURN                                                            CUA06070
820 ISDONE = 0                                                        CUA06080
    RETURN                                                            CUA06090
    END                                                               CUA06100
                                                                      CUA06110
                                                                      CUA06120
    SUBROUTINE SHOWIT                                                 CUA06130
    IMPLICIT REAL*8 (A-H,O-Z)                                         CUA06140
    DIMENSION SOL(50)                                                 CUA06150
    INTEGER DSPLAY,RL,BASIC1,BASIC0                                   CUA06160
    COMMON /VALUES/ EPS,MIO,ITER,DSPLAY                               CUA06170
    COMMON /NUMBIC/ N,M                                               CUA06180
```

```
      COMMON /OFUNCN/ A(100)                                            CUA06190
      COMMON /SUBJCT/ P(200)                                            CUA06200
      COMMON /BASICA/ BASIC0(50),BASIC1(50)                             CUA06210
      COMMON /LENGTH/ RL                                                CUA06220
      IF(DSPLAY.EQ.0)GO TO 1890                                         CUA06230
      IF(DSPLAY.EQ.1.AND.ITER.EQ.0)CALL MESSAG(17)                      CUA06240
      IF(DSPLAY.EQ.1.AND.ITER.EQ.0)GO TO 1820                           CUA06250
      IF(DSPLAY.GE.2.AND.ITER.EQ.0)GO TO 1812                           CUA06260
 1813 WRITE(6,213)ITER                                                  CUA06270
      IF(DSPLAY.EQ.1)GO TO 1820                                         CUA06280
 1812 CALL PROFUN                                                       CUA06290
      IF(DSPLAY.EQ.2.AND.M.EQ.0.AND.ITER.EQ.0)GO TO 1820                CUA06300
      CALL PRICON                                                       CUA06310
 1820 ITER = ITER + 1                                                   CUA06320
      WRITE(6,106)A(1)                                                  CUA06330
      K=M+1                                                             CUA06340
      DO 28 I=1,K                                                       CUA06350
         KIO=(I-1)*RL+1                                                 CUA06360
         SOL(I)=P(KIO)                                                  CUA06370
   28 CONTINUE                                                          CUA06380
      WRITE(6,107)(BASIC1(K),SOL(K+1),K=1,M)                            CUA06390
      CALL MESSAG(18)                                                   CUA06400
 1890 RETURN                                                            CUA06410
  106 FORMAT(//,5X,' THE CURRENT VALUE OF THE OBJECTIVE FUNCTION IS ',  CUA06420
     *F15.8)                                                            CUA06430
  213 FORMAT(/,15X,'ITERATION NUMBER ',I2)                              CUA06440
  107 FORMAT(/,5X,' THE VALUES OF THE VARIABLES ARE ',/,30X,'X',I2,'     CUA06450
     *= ',F15.8,/,30X,'X'))                                             CUA06460
      END                                                               CUA06470
                                                                        CUA06480
                                                                        CUA06490
      SUBROUTINE OPTIMO                                                 CUA06500
      IMPLICIT REAL*8 (A-H,O-Z)                                         CUA06510
      DIMENSION SOL(50)                                                 CUA06520
      COMMON /VALUES/ TPS,MIO,ITER,DSPLAY                               CUA06530
      COMMON /FINAL / ISDONE                                            CUA06540
      COMMON /NUMBRO/ N,M                                               CUA06550
      COMMON /BASICA/ BASIC0(50),BASIC1(50)                             CUA06560
      COMMON /SUBJCT/ P(200)                                            CUA06570
      COMMON /OFUNCN/ A(100)                                            CUA06580
      COMMON /LENGTH/ RL                                                CUA06590
      INTEGER DSPLAY,S,RL,BASIC1,BASIC0                                 CUA06600
      IF(ISDONE.EQ.0)CALL MESSAG(19)                                    CUA06610
      IF(ISDONE.EQ.0)GO TO 1750                                         CUA06620
      IF(ISDONE.EQ.1)CALL MESSAG(20)                                    CUA06630
      IF(DSPLAY.EQ.0.OR.DSPLAY.EQ.1)GO TO 1710                          CUA06640
      CALL PROFUN                                                       CUA06650
      CALL PRICON                                                       CUA06660
 1710 WRITE(6,106)A(1)                                                  CUA06670
      K=M+1                                                             CUA06680
      DO 28 I=1,K                                                       CUA06690
         KIO=(I-1)*RL+1                                                 CUA06700
```

```
      SOL(I)=P(KIO)                                                    CUA06710
 28 CONTINUE                                                           CUA06720
    WRITE(6,107)(BASIC1(K),SOL(K+1),K=1,M)                             CUA06730
    CALL MESSAG(18)                                                    CUA06740
    CALL MESSAG(21)                                                    CUA06750
1750 RETURN                                                            CUA06760
 106 FORMAT(//,5X,' THE OPTIMAL IS ',F15.8)                            CUA06770
 107 FORMAT(/,5X,' THE VALUES OF THE VARIABLES ARE ',/,31X,'X',10(I2, 'CUA06780
   *=',F15.8,/,31X,'X'))                                               CUA06790
 108 FORMAT(/,5X,' ALL THE OTHERS ARE EQUAL TO ZERO ',/,78('-'),/,58X,'CUA06800
   1OSCAR MANRIQUE, 1982',/,78('-'))                                   CUA06810
    END                                                                CUA06820
                                                                       CUA06830
                                                                       CUA06840
    SUBROUTINE DIALOG                                                  CUA06850
    IMPLICIT REAL*8 (A-H,O-Z)                                          CUA06860
    DIMENSION G(200),D(100)                                            CUA06870
    INTEGER DSPLAY                                                     CUA06880
    COMMON /NUMERO/ N,M                                                CUA06890
    COMMON /SUBJCT/ P(200),Q(50)                                       CUA06900
    COMMON /OFUNCN/ A(100),B(50),C                                     CUA06910
    COMMON /VALUES/ F,MIO,ITER,DSPLAY                                  CUA06920
    DATA BLANK/' '/                                                    CUA06930
    CALL MESSAG(1)                                                     CUA06940
    CALL MESSAG(2)                                                     CUA06950
    CALL MESSAG(3)                                                     CUA06960
    KEY=0                                                              CUA06970
 100 CALL MESSAG(4)                                                    CUA06980
    CALL READIT(N)                                                     CUA06990
    IF(N.GT.0)GO TO 41                                                 CUA07000
    WRITE(6,60)N                                                       CUA07010
    GO TO 100                                                          CUA07020
  41 IF(KEY.EQ.1)GO TO 1901                                            CUA07030
 102 CALL MESSAG(5)                                                    CUA07040
    CALL READIT(M)                                                     CUA07050
    IF(M.GE.0)GO TO 42                                                 CUA07060
    CALL MESSAG(6)                                                     CUA07070
    GO TO 102                                                          CUA07080
  42 IF(M.LE.N)GO TO 43                                                CUA07090
    WRITE(6,67)N                                                       CUA07100
    GO TO 102                                                          CUA07110
  43 IF(KEY.EQ.1)GO TO 106                                             CUA07120
    CALL MESSAG(7)                                                     CUA07130
    KO=0                                                               CUA07140
    JIK=0                                                              CUA07150
 121 DO 13 J=1,N                                                       CUA07160
       IF(JIK.EQ.N)GO TO 104                                           CUA07170
       IF(JIK.NE.0)GO TO 122                                           CUA07180
       JL=J-1                                                          CUA07190
       NI=N-JL                                                         CUA07200
       NII=NI                                                          CUA07210
       IF(NI.EQ.1)GO TO 104                                            CUA07220
```

```
      WRITE(6,70)NI,J                                                   CUA07230
      GO TO 105                                                         CUA07240
122   WRITE(6,70)NIL,JIK                                               CUA07250
      GO TO 105                                                         CUA07260
104   CALL MESSAG(8)                                                    CUA07270
105   CALL READEM(D,NIL)                                                CUA07280
      KIN=1                                                             CUA07290
      IF(JIK.NE.0)KO=IO-NIL-1                                           CUA07300
      IF(JIK.NE.0)NI=NIL                                                CUA07310
      DO 14 K=KIN,NI                                                    CUA07320
         KO=KO+1                                                        CUA07330
         A(KO)=D(K)                                                     CUA07340
 14   CONTINUE                                                          CUA07350
      IF(KEY.EQ.1)IA=IA-NIL                                             CUA07360
      IF(KEY.EQ.1)GO TO 1017                                            CUA07370
 13 CONTINUE                                                            CUA07380
118  WRITE(6,68)N                                                      CUA07390
     CALL READEM(B,N)                                                  CUA07400
     IF(KEY.EQ.1)GO TO 115                                             CUA07410
116  CALL MESSAG(9)                                                    CUA07420
     CALL READEM(C,1)                                                  CUA07430
     IF(KEY.EQ.1)GO TO 113                                             CUA07440
     IF(M.EQ.0)GO TO 140                                               CUA07450
130  JA = 1                                                            CUA07460
     JIO = 0                                                           CUA07470
112  DO 11 I=1,M                                                       CUA07480
        IF(JIO.NE.0.AND.KEY.EQ.1)GO TO 110                             CUA07490
        WRITE(6,66)N,I                                                 CUA07500
        GO TO 111                                                      CUA07510
110     WRITE(6,66)N,JIO                                               CUA07520
111     CALL READEM(G,N)                                               CUA07530
        IF(KEY.EQ.0)JIO=I                                              CUA07540
        IF(KEY.EQ.1)JA=(JIO-1)*N+1                                     CUA07550
        JI=JIO*N                                                       CUA07560
        J=1                                                            CUA07570
        DO 12 K=JA,JI                                                  CUA07580
           B(K) = G(J)                                                 CUA07590
           J = J + 1                                                   CUA07600
 12     CONTINUE                                                       CUA07610
        IF(KEY.EQ.1)GO TO 108                                          CUA07620
        JA = JI + 1                                                    CUA07630
 11   CONTINUE                                                         CUA07640
     IF(M.GT.1)GO TO 103                                               CUA07650
     CALL MESSAG(10)                                                   CUA07660
     GO TO 131                                                         CUA07670
103  WRITE(6,64)M                                                      CUA07680
131  CALL READEM(Q,M)                                                  CUA07690
     DO 10 I=1,M                                                       CUA07700
        IF(Q(I).GT.0)GO TO 10                                          CUA07710
        WRITE(6,65)I,Q(I)                                              CUA07720
        GO TO 103                                                      CUA07730
10   CONTINUE                                                          CUA07740
```

```
      IF(KEY.EQ.1)GO TO 180                                             CUA07750
  140 CALL MESSAG(22)                                                   CUA07760
      CALL READEM(I,1)                                                  CUA07770
      IF(KEY.EQ.1)GO TO 181                                             CUA07780
C------ NEXT SECTION CHECKS THE CORRECTNESS OF THE DATA ---------------CUA07790
      CALL MESSAG(11)                                                   CUA07800
      KEY = 1                                                           CUA07810
 1901 WRITE(6,901)N                                                     CUA07820
      REWIND 5                                                          CUA07830
      READ(5,33,END=106)CHAR                                            CUA07840
      IF(CHAR.NE.BLANK)GO TO 100                                        CUA07850
  106 IF(M.EQ.0)CALL MESSAG(12)                                         CUA07860
      IF(M.EQ.0)GO TO 701                                               CUA07870
      WRITE(6,74)M                                                      CUA07880
  701 REWIND 5                                                          CUA07890
      READ(5,33,END=107)CHAR                                            CUA07900
      IF(CHAR.NE.BLANK)GO TO 102                                        CUA07910
  107 IO=1                                                              CUA07920
      IA=0                                                              CUA07930
      DO 16 I=1,N                                                       CUA07940
         IL=I-1                                                         CUA07950
         NIL=N-IL                                                       CUA07960
         IO=IO+NIL                                                      CUA07970
 1017    IF(I.EQ.N)GO TO 128                                            CUA07980
         IA=IA+NIL                                                      CUA07990
         JO=IO-NIL                                                      CUA08000
         WRITE(6,79)NIL,I,(A(KI),KI=JO,IA)                              CUA08010
         GO TO 119                                                      CUA08020
  128    LST=(N+1)*N/2                                                  CUA08030
         WRITE(6,80)A(LST)                                              CUA08040
  119    REWIND 5                                                       CUA08050
         READ(5,33,END=16)CHAR                                          CUA08060
         IF(CHAR.EQ.BLANK)GO TO 16                                      CUA08070
         JIK=I                                                          CUA08080
         GO TO 121                                                      CUA08090
   16 CONTINUE                                                          CUA08100
  115 WRITE(6,78)N,(B(KI),KI=1,N)                                       CUA08110
      REWIND 5                                                          CUA08120
      READ(5,33,END=113)CHAR                                            CUA08130
      IF(CHAR.NE.BLANK)GO TO 118                                        CUA08140
  113 WRITE(6,77)C                                                      CUA08150
      REWIND 5                                                          CUA08160
      READ(5,33,END=1180)CHAR                                           CUA08170
      IF(CHAR.NE.BLANK)GO TO 116                                        CUA08180
 1180 IF(M.EQ.0)GO TO 181                                               CUA08190
      DO 15 IK=1,M                                                      CUA08200
         KA=IK*N                                                        CUA08210
         KO=(IK-1)*N+1                                                  CUA08220
  108    WRITE(6,75)N,IK,(P(J),J=KO,KA)                                 CUA08230
         REWIND 5                                                       CUA08240
         READ(5,33,END=15)CHAR                                          CUA08250
         IF(CHAR.EQ.BLANK)GO TO 15                                      CUA08260
```

```
      JIC=IK                                                            CUA08270
      GO TO 112                                                         CUA08280
 15   CONTINUE                                                          CUA08290
100   WRITE(6,72)M,(Q(I),I=1,M)                                         CUA08300
      REWIND 5                                                          CUA08310
      READ(5,33,END=181)CHAR                                            CUA08320
      IF(CHAR.NE.BLANK)GO TO 103                                        CUA08330
181   WRITE(6,81)                                                       CUA08340
      REWIND 5                                                          CUA08350
      READ(5,33,END=888)CHAR                                            CUA08360
      IF(CHAR.NE.BLANK)GO TO 140                                        CUA08370
 888  CALL MESSAG(13)                                                   CUA08380
      CALL READIT(DSPLAY)                                               CUA08390
      RETURN                                                            CUA08400
  60  FORMAT(/,' YOUR VALUE OF N = ',I2,' IS NOT POSITIVE  ! ',/,'      CUA08410
     1RESTART .. .')                                                    CUA08420
  67  FORMAT(/,' THE STANDARIZED PROBLEM REQUIRES  M <= N ... .',/,     CUA08430
     *' IN YOUR PROBLEM, A MAXIMUM OF ',I2,' CONSTRAINTS (M) WILL BE ACCCUA08440
     *EPTED')                                                           CUA08450
  68  FORMAT(/,' TYPE THE ',I2,'-VECTOR B, (THE LINEAR PART OF YOUR OBJECUA08460
     1CTIVE FUNCTION) ')                                                CUA08470
  66  FORMAT(/,' TYPE THE ',I2,' VALUES OF THE ROW NO. ',I2,' IN YOUR COCUA08480
     1NSTRAINTS MATRIX P')                                              CUA08490
  70  FORMAT(/,' TYPE THE ',I2,' VALUES OF THE ROW NUMBER ',I2)         CUA08500
  64  FORMAT(/,' TYPE THE VALUE OF YOUR ',I2,'-VECTOR Q. (THE RIGHT-HANDCUA08510
     1 SIDE VALUE IN YOUR ',/,' CONSTRAINTS)')                          CUA08520
  65  FORMAT(/,' THE COMPONENT NO. ',I2,' OF YOUR Q VECTOR, NAMELY ',D12CUA08530
     *.5,' IS NOT POSITIVE:',/,' YOUR PROBLEM IS NOT IN STANDARD FORM',/CUA08540
     *,'INPUT AGAIN .... ')                                             CUA08550
  79  FORMAT(/,' ARE THESE THE CORRECT ',I2,' ELEMENTS OF THE ROW NUMBERCUA08560
     * ',I2,' OF MATRIX  A ',/,' IN YOUR OBJECTIVE FUNCTION ?',/,1X,99D1CUA08570
     *2.5)                                                              CUA08580
  74  FORMAT(/,' YOU HAVE ',I2,' CONSTRAINTS ? ')                       CUA08590
  33  FORMAT(A5)                                                        CUA08600
 901  FORMAT(/,' YOU HAVE ',I2,' VARIABLES ? ')                         CUA08610
  78  FORMAT(/,' ARE THESE THE CORRECT VALUES OF YOUR ',I2,'-VECTOR B ',CUA08620
     */,' (THE LINEAR PART IN YOUR OBJECTIVE FUNCTION)?',/,50(2X,D12.5))CUA08630
  80  FORMAT(/,' IS ',D12.5,' THE LAST ELEMENT OF THE MATRIX IN YOUR OBJCUA08640
     *ECTIVE FUNCTION ?')                                               CUA08650
  72  FORMAT(/,' ARE THESE THE CORRECT VALUES OF YOUR ',I2,'-VECTOR Q  'CUA08660
     *,/,' (THE RIGHT-HAND SIDE TERMS IN YOUR CONSTRAINTS) ?',/,        CUA08670
     *10(2X,D12.5))                                                     CUA08680
  75  FORMAT(/,' ARE THESE THE CORRECT ',I2,' VALUES OF ROW NUMBER ',I2,CUA08690
     1/,' IN YOUR MATRIX P (CONSTRAINTS) ? ',/,50(2X,D12.5))           CUA08700
  77  FORMAT(/,' THE CONSTANT VALUE C IN YOUR OBJECTIVE FUNCTION IS '   CUA08710
     1,D12.5,' ?')                                                      CUA08720
  81  FORMAT(/,' THE WIDTH OF YOUR ZERO-BAND IS ',D12.5,  ' ?')         CUA08730
      END                                                               CUA08740
                                                                        CUA08750
                                                                        CUA08760
      SUBROUTINE MESSAG(INDEX)                                          CUA08770
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22), CUA08780
```

```
 *INDEX                                                                  CUA08790
 1 WRITE(6,90)                                                           CUA08800
   RETURN                                                                CUA08810
 2 WRITE(6,92)                                                           CUA08820
   RETURN                                                                CUA08830
 3 WRITE(6,93)                                                           CUA08840
   RETURN                                                                CUA08850
 4 WRITE(6,101)                                                          CUA08860
   RETURN                                                                CUA08870
 5 WRITE(6,61)                                                           CUA08880
   RETURN                                                                CUA08890
 6 WRITE(6,62)                                                           CUA08900
   RETURN                                                                CUA08910
 7 WRITE(6,69)                                                           CUA08920
   RETURN                                                                CUA08930
 8 WRITE(6,71)                                                           CUA08940
   RETURN                                                                CUA08950
 9 WRITE(6,67)                                                           CUA08960
   RETURN                                                                CUA08970
10 WRITE(6,59)                                                           CUA08980
   RETURN                                                                CUA08990
11 WRITE(6,73)                                                           CUA09000
   RETURN                                                                CUA09010
12 WRITE(6,700)                                                          CUA09020
   RETURN                                                                CUA09030
13 WRITE(6,88)                                                           CUA09040
   RETURN                                                                CUA09050
14 WRITE(6,63)                                                           CUA09060
   RETURN                                                                CUA09070
15 WRITE(6,212)                                                          CUA09080
   RETURN                                                                CUA09090
16 WRITE(6,302)                                                          CUA09100
   RETURN                                                                CUA09110
17 WRITE(6,181)                                                          CUA09120
   RETURN                                                                CUA09130
18 WRITE(6,108)                                                          CUA09140
   RETURN                                                                CUA09150
19 WRITE(6,105)                                                          CUA09160
   RETURN                                                                CUA09170
20 WRITE(6,140)                                                          CUA09180
   RETURN                                                                CUA09190
21 WRITE(6,141)                                                          CUA09200
   RETURN                                                                CUA09210
22 WRITE(6,150)                                                          CUA09220
   RETURN                                                                CUA09230
90 FORMAT(//,59X,'UNIVERSITY OF OTTAWA',/,62X,'SYSTEMS SCIENCE',/,68XCUA09240
  *,'1982',//,1X,78('='),/,35X,'  CUADRO  ',/,1X,78('='),//,' THIS PACUA09250
  *CKAGE USES THE METHOD OF BEALE TO SOLVE THE QUADRATIC PROGRAMMING'CUA09260
  *,/,' PROBLEM, HAVING N VARIABLES AND M CONSTRAINTS (M<=N), GIVEN ICUA09270
  *N ITS STANDARD ',/,' FORM :',///,                                      CUA09280
  *'                  MIN G = (1/2)X''AX + B''Y + C ',/,                 CUA09290
  *' SUBJECT TO :      PX <= Q             ,   WITH  Q(J) > 0  ',/,      CUA09300
```

```
   *'                  X(K) >= 0                        K = 1, 2, ... , N ',/,CUA09310
   *'                                                   J = 1, 2, ... , N ',/,CUA09320
   *' WHERE :    C : THE CONSTANT VALUE OF YOUR OBJECTIVE FUNCTION',/, CUA09330
   *'            B : THE VECTOR OF THE LINEAR PART IN THE Q. F.    ',/,CUA09340
   *'            A : AN NXN MATRIX CONTAINING THE QUADRATIC COEFICIENTSCUA09350
   * OF THE Q. F.',/,                                                 CUA09360
   *'            P : AN NXM MATRIX WITH THE COEFICIENTS OF THE CONSTRAICUA09370
   *NTS ',/,                                                         CUA09380
   *'            Q : AN M-VECTOR WITH THE RIGHT-HAND SIDE VALUES OF THCUA09390
   * CONSTRAINTS',//)                                               CUA09400
92  FORMAT(' NOTE THAT MATRIX A IS A SYMMETRIC MATRIX WITH A(I,J) = 'CUA09410
   *,/,                                                             CUA09420
   *'  TWICE THE COEFICIENT OF X(K)**2 WHEN K=I=J  (THE DIAGONAL OF THCUA09430
   *E MATRIX)',/,                                                   CUA09440
   *'  COEF. OF X(I)X(J) WHEN I DIFFERENT FROM J  (THE OFF-DIAGONAL ELCUA09450
   *EMENTS) ')                                                      CUA09460
 93 FORMAT(/,' WHEN TYPING THE DATA REQUESTED, PLEASE RECALL : ',/,  CUA09470
   *' *   AT LEAST ONE BLANK (SPACE) MUST SEPARATE SUCCESSIVE NUMBERS CUA09480
   *    ',/,                                                        CUA09490
   *' *   THE DECIMAL POINT IS NOT NECESSARY, UNLESS IT IS PART OF THECUA09500
   * NUMBER    ',/,                                                 CUA09510
   *' *   IF YOU WANT TO QUIT TYPE "Q" INSTEAD OF THE INFORMATION REQUCUA09520
   *ESTED',/,                                                       CUA09530
   *' *   WHEN ANSWERING TRUE OR FALSE,  ',/,                       CUA09540
   *'           TYPE "NO" AND <CR>  IF FALSE, OR ',/,              CUA09550
   *'           <CR> (HIT RETURN)   IF TRUE ',/,80('-'))           CUA09560
101 FORMAT(/,' TYPE THE VALUE OF N (I.E. THE NUMBER OF VARIABLES YOU CUA09570
   *HAVE  )')                                                       CUA09580
 61 FORMAT(/,' TYPE THE VALUE OF M (I.E. THE NUMBER OF CONSTRAINTS YOUCUA09590
   * HAVE)    ')                                                   CUA09600
 62 FORMAT(/,' YOUR NUMBER OF CONSTRAINTS CANNOT BE NEGATIVE  !',/,' CUA09610
   *TRY AGAIN .. .')                                               CUA09620
 69 FORMAT(/,' THE QUADRATIC PART OF THE OBJECTIVE FUNCTION, BEING A SCUA09630
   *YMMETRIC MATRIX,',/,' ONLY THE UPPER-TRIANGULAR PART MUST BE TYPEDCUA09640
   *')                                                             CUA09650
 71 FORMAT(/,' TYPE THE VERY LAST ELEMENT OF YOUR MATRIX ')        CUA09660
 67 FORMAT(/,' TYPE THE VALUE OF C, I.E. THE CONSTANT PART IN YOUR OBJCUA09670
   *ECTIVE FUNCTION ')                                             CUA09680
 59 FORMAT(/,' TYPE THE VALUE OF Q IN YOUR CONSTRAINT (I.E., THE RIGHTCUA09690
   *-HAND SIDE VALUE) ')                                           CUA09700
 73 FORMAT(/,' NOW, LET US CHECK THE CORRECTNESS OF THE DATA RECEIVED'CUA09710
   *,/,'     TYPE "NO" AND <CR>      IF FALSE, OR '                CUA09720
   *,/,'     <CR>  (HIT RETURN)      IF TRUE ')                    CUA09730
700 FORMAT(/,' YOUR PROBLEM IS UNCONSTRAINED ?')                   CUA09740
 88 FORMAT(/,' SELECT ONE OF THE FOLLOWING OPTIONS : ',/,' 0  >ONLY THCUA09750
   *E FINAL ANSWER WILL BE SHOWN',/,' 1  >THE VALUES OF THE',      CUA09760
   *' OBJECTIVE FUNCTION ',                                        CUA09770
   *     'AND THE BASIC VARIABLES WILL BE SHOWN ',/,5X,'AT EACH ITERACUA09780
   *ION',/,' 2  >THE COMPLETE TABLEAUX FOR THE METHOD OF BEALE WILL BCUA09790
   * DISPLAYED AT EACH ',/,5X,'ITERATION ',//,' TYPE THE NUMBER OF THECUA09800
   * OPTION YOU CHOOSE')                                           CUA09810
 63  FORMAT(' YOU DID NOT ENTRY ANY DATA ...! INPUT AGAIN OR TYPE Q TO CUA09820
```

```
   * QUIT')                                                                CUA09830
212 FORMAT(/,15X,'ORIGINAL TABLEAU')                                       CUA09840
302 FORMAT(//,5X,'CONSTRAINTS')                                            CUA09850
181 FORMAT(/,15X,'STARTING POINT ')                                        CUA09860
108 FORMAT(/,5X,' ALL THE OTHERS ARE EQUAL TO ZERO ',/,80('-'))            CUA09870
105 FORMAT(//,5X,'  *** THE PROBLEM DOES NOT HAVE A FINITE SOLUTION **CUA09880
   ** ',/,78('-'),/,58Y,'OSCAR MANRIQUE, 1982',/,78('-'))                 CUA09890
140 FORMAT(///,36X,' SOLUTION ')                                           CUA09900
141 FORMAT(58X,'OSCAR MANRIQUE,1982',/,79('-'))                            CUA09910
150 FORMAT(/,' THE VALUES IN THE ZERO-BAND ARE TAKEN AS ZERO IN CRITICCUA09920
   *AL POINTS ',/,' OF THE PROBLEM.',/,' TYPE THE VALUE OF THE ZERO-BACUA09930
   *ND FOR YOUR PROBLEM ')                                                 CUA09940
    END                                                                    CUA09950
                                                                           CUA09960
                                                                           CUA09970
    SUBROUTINE READEM(F,L)                                                 CUA09980
    INTEGER A(80),NUMBR(12),DOT,BLANK,QUIT                                 CUA09990
    INTEGER FMT(40)                                                        CUA10000
    DOUBLE PRECISION F(40)                                                 CUA10010
    LOGICAL FLAG                                                           CUA10020
    DATA BLANK/' '/,A/80*' '/,QUIT/'Q'/,MIN/'-'/,DOT/'.'/                  CUA10030
    DATA NUMBR/'1','2','3','4','5','6','7','8','9','0','D','E'/           CUA10040
40  REWIND 5                                                               CUA10050
    READ(5,100,END=77)(A(I),I=1,80)                                        CUA10060
    FLAG=.TRUE.                                                            CUA10070
    DO 45 I=1,80                                                           CUA10080
       IF(A(I).EQ.QUIT)CALL EXIT                                           CUA10090
45  CONTINUE                                                               CUA10100
77  DO 60 J = 1,80                                                        CUA10110
       IF(A(J).NE.BLANK)GO TO 63                                           CUA10120
60  CONTINUE                                                               CUA10130
    CALL MESSAG(14)                                                        CUA10140
    GO TO 40                                                               CUA10150
63  DO 70 I=1,80                                                           CUA10160
       IF(A(I).EQ.MIN.OR.A(I).EQ.DOT.OR.A(I).EQ.BLANK)GO TO 70             CUA10170
       DO 72 K=1,12                                                        CUA10180
          IF(A(I).EQ.NUMBR(K))GO TO 70                                     CUA10190
72     CONTINUE                                                            CUA10200
       WRITE(6,101)A(I)                                                    CUA10210
       FLAG=.FALSE.                                                        CUA10220
       GO TO 75                                                            CUA10230
70  CONTINUE                                                               CUA10240
75  IF(FLAG)GO TO 66                                                       CUA10250
    WRITE(6,102)(A(K),K=1,80)                                              CUA10260
    GO TO 40                                                               CUA10270
 66 JAK=1                                                                  CUA10280
    I2=0                                                                   CUA10290
    I1 = 0                                                                 CUA10300
    K=1                                                                    CUA10310
50  IF(A(K).EQ.BLANK.AND.I1.NE.0)I2=K-1                                    CUA10320
    IF(A(K).NE.BLANK.AND.I1.EQ.0)I1=K                                      CUA10330
    IF(I2.EQ.0)GO TO 10                                                    CUA10340
```

```
      IF(I2-I1+1.LE.9)GO TO 22                                          CUA10350
      WRITE(6,104)JAK,(A(K),K=1,80)                                     CUA10360
      GO TO 40                                                          CUA10370
   22 WRITE(99,103) (A(KI),KI=I1,I2)                                    CUA10380
      CALL SETB99(FMT,10)                                               CUA10390
      IH=I2-I1+1                                                        CUA10400
      WRITE(99,99)IH                                                    CUA10410
      CALL RESB99                                                       CUA10420
      READ(99,FMT)F(JAK)                                                CUA10430
      JAK=JAK+1                                                         CUA10440
      I1=0                                                              CUA10450
      I2=0                                                              CUA10460
   10 K=K+1                                                             CUA10470
      IF(K.GT.80)GO TO 78                                               CUA10480
      IF(JAK.GT.L)GO TO 78                                              CUA10490
      GO TO 50                                                          CUA10500
   78 RETURN                                                            CUA10510
   99 FORMAT('(D',I1,'.0)')                                             CUA10520
  100 FORMAT(80A1)                                                      CUA10530
  101 FORMAT(' ------------> ',A1)                                      CUA10540
  102 FORMAT(' ***** ERROR IN YOUR ENTRY :',/,80A1,/,' INPUT AGAIN OR TYCUA10550
     *PE Q TO QUIT')                                                    CUA10560
  103 FORMAT(9A1)                                                       CUA10570
  104 FORMAT(/,' YOUR ENTRY NUMBER ',I2,' IS TOO LONG. A MAXIMUM OF 9 POCUA10580
     *SITIONS ',/,' WILL BE ACCEPTED. INPUT AGAIN : ',/,80A1)           CUA10590
      END                                                               CUA10600
                                                                        CUA10610
                                                                        CUA10620
      SUBROUTINE READIT(F)                                              CUA10630
      INTEGER A(80),NUMBR(10),BLANK,QUIT                                CUA10640
      INTEGER F(1),FMT(10)                                              CUA10650
      LOGICAL FLAG                                                      CUA10660
      DATA A/80*' '/, BLANK/' '/,QUIT/'Q'/,MIN/'-'/                     CUA10670
      DATA NUMBR/'1','2','3','4','5','6','7','8','9','0'/              CUA10680
   40 REWIND 5                                                          CUA10690
      READ(5,100,END=77) (A(I),I=1,80)                                  CUA10700
      DO 45 I=1,80                                                      CUA10710
         IF(A(I).EQ.QUIT)CALL EXIT                                      CUA10720
   45 CONTINUE                                                          CUA10730
      FLAG=.TRUE.                                                       CUA10740
   77 DO 60 J=1,80                                                      CUA10750
         IF(A(J).NE.BLANK)GO TO 63                                      CUA10760
   60 CONTINUE                                                          CUA10770
      CALL MESSAG(14)                                                   CUA10780
      GO TO 40                                                          CUA10790
   63 DO 70 I=1,80                                                      CUA10800
         IF(A(I).EQ.MIN.OR.A(I).EQ.BLANK)GO TO 70                       CUA10810
         DO 72 K=1,10                                                   CUA10820
            IF(A(I).EQ.NUMBR(K))GO TO 70                                CUA10830
   72    CONTINUE                                                       CUA10840
         WRITE(6,103)A(I)                                               CUA10850
         FLAG=.FALSE.                                                   CUA10860
```

```
      GO TO 75                                                          CUA10870
 70 CONTINUE                                                            CUA10880
 75 IF(FLAG)GO TO 66                                                    CUA10890
    WRITE(6,102)(A(K),K=1,80)                                           CUA10900
    GO TO 40                                                            CUA10910
 66 MIKE=0                                                              CUA10920
    I2=0                                                                CUA10930
    I1=0                                                                CUA10940
    K=1                                                                 CUA10950
 50 IF(A(K).EQ.BLANK.AND.I1.NE.0)I2=K-1                                 CUA10960
    IF(A(K).NE.BLANK.AND.I1.EQ.0)I1=K                                   CUA10970
    IF(I2.EQ.0)GO TO 10                                                 CUA10980
    WRITE(99,101)(A(KI),KI=I1,I2)                                       CUA10990
    CALL SETB99(FMT,10)                                                 CUA11000
    IH=I2-I1+1                                                          CUA11010
    WRITE(99,99)IH                                                      CUA11020
    CALL RESB99                                                         CUA11030
    READ(99,FMT)F                                                       CUA11040
    MIKE=1                                                              CUA11050
 10 K=K+1                                                               CUA11060
    IF(K.GT.80)GO TO 78                                                 CUA11070
    IF(MIKE.NE.0)GO TO 78                                               CUA11080
    GO TO 50                                                            CUA11090
 78 RETURN                                                              CUA11100
 99 FORMAT('(I',I1,')')                                                 CUA11110
101 FORMAT(9A1)                                                         CUA11120
100 FORMAT(80A1)                                                        CUA11130
102 FORMAT(' ***** ERROR IN YOUR ENTRY :',/,80A1,/,' INPUT AGAIN OR     CUA11140
   *TYPE Q TO QUIT')                                                    CUA11150
103 FORMAT(' ---------------> ',A1)                                     CUA11160
    END                                                                 CUA11170
```

BIBLIOGRAPHY

[1] BEALE E.M.L., "On quadratic programming", Naval Research Logistics Quarterly vol. 6 (1959), pp. 227-243.

[2] BEALE E.M.L., "Applications of mathematical programming techniques", A conference held in Cambridge, England, on the 24th-28th of June 1968 under the aegis of the NATO Scientific Affairs Committee, American Elsevier Publishing Company Inc. New York, 1970.

[3] BEALE E.M.L., "Numerical methods", in: "Nonlinear programming", edited by J. Abadie, North-Holland Publishing Company, Amsterdam, 1967.

[4] BEALE E.M.L., "Mathematical programming in practice", Topics in operations research, Pitman Publishing, Belfast, 1971.

[5] BOOT JOHN C.G., "Quadratic programming : algorithms, anomalies, applications", Collection Studies in Mathematical and Managerial Economics, Vol. 2, North-Holland Publishing Company, Amsterdam, 1964.

[6] BRACKEN JEROME, McCORMICK GARTH P., "Selected applications of nonlinear programming", John Wiley and Sons Inc., New York, 1968.

[7] BRADLEY STEPHEN P., HAX ARNOLDO C., MAGNANTI THOMAS L., "Applied mathematical programming", Addison-Wesley Publishing Company, Massachusetts, 1977.

[8] BRAITSCH JR. R.J., "A computer comparison of four quadratic programming algorithms", Management Science Vol. 18 (1972) pp. 632-643.

[9] CEA JEAN, "Optimisation : théorie et algorithmes", Collection méthodes mathématiques de l'informatique, Dunod, Paris, 1971.

[10] COOPER LEON, STEINBERG DAVID, "Introduction to methods of optimization", W.B. Saunders Company, Philadelphia, Pa, 1970.

[11] DANTZIG G.B., "Quadratic programming: a variant of Wolfe- Markowitz algorithm", Operations Research

Center, University of California, Berkeley, R.P. 2 (april 1961).

[12] GANTMACHER F.R., "Théorie des matrices, Tome I: théorie générale", Traduit du russe par Ch. Sathou, Dunod, Paris, 1966.

[13] GARVIN WALTER W., "Introduction to linear programming", McGraw-Hill Book Company, New York, 1960.

[14] GILL P.E., MURRAY W., "Numerical methods for constrained optimization", from the symposium on numerical methods for constrained optimization held at the National Physical Laboratory, Academic Press, London, 1974.

[15] GILL P.E., MURRAY W., "Numerical stable methods for quadratic programming", Mathematical Programming, vol 14 (1978) pp. 349-372.

[16] GILL P.E. et al., "QP-based methods for large-scale nonlinear constrained optimization", Systems Optimization Laboratory, Technical Report, January 1981, Department of Operations Research, Stanford University, Stanford, Ca., 1981

[17] INDUSI JOSEPH P., "A computer algorithm for constrained minimization", In : "Minimization algorithms, mathematical theories and computer results", edited by G.P.Szego, University of Venice, Italy, Academic Press, New york-London, 1972.

[18] KUHN H.W., TUCKER A.W., "Nonlinear programming", in Proceedings of the Second Berkeley Symposium on mathematical statistics and probability, edited by J. Neyman (Berkeley-Los Angeles, 1951), pp. 481-492

[19] KUNZI H.P., KRELLE W., "La programmation nonlinéaire", Collection de mathématiques économiques, Fascicule VI, Gauthier-Villars, Paris, 1969.

[20] KUNZI H.P., TZSCHACH H.G., ZEHNDER C.A., "Numerical methos of mathematical optimization", series Computer science and applied mathematics, Academic Press, New York-London, 1968.

[21] LANCASTER PETER, "Theory of matrices", Academic Press, New York, 1969.

[22] MANGASARIAN OLVI L., "Nonlinear programming", McGraw-Hill series in Systems Science, McGraw-Hill Book Company, U.S.A.

[23] MARKOWITZ H.M., "The optimization of quadratic functions subject to linear constraints", Naval Research Logistics Quarterly, Vol.3, 1959.

[24] MOORE J.H., WHINSTON A.B., "Experimental methods in quadratic programming", Management Science Vol. 13 (1966), pp. 58-76.

[25] RAVINDRAN A., LEE HARVEY K., "Computer experiments on quadratic programming algorithms", EJOR Vol. 8 (1981) No. 2 (october), pp 166-174.

[26] SEGALEN H., JOUVENT M., "La programmation nonlinéaire", Collection la vie de l'entreprise, Dunod, Paris, 1971.

[27] SIMMONDS DONALD M., "Nonlinear programming for operations Research", Prentice Hall International series in management, Prentice-Hall Inc., Englewood Cliffs, N.J. 1975.

[28] VAN DE PANNE C., "Methods for linear and quadratic programming", Studies in mathematical and managerial economics, edited by Henri Theil, North-Holland Publishing Co., Amsterdam, 1975.

[29] VAN DE PANNE C., WHINSTON A.B., "A comparison of two methods for quadratic programming", Operations Research, Vol. 14 (1966) pp. 422-441.

[30] WOLFE PH., "The simplex method for quadratic programming", Econometrica, Vol. 27, 1959.