

A SOLUTION OF QUADRATIC PROGRAMMING USING MATLAB

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Science Degree in Mathematics

BY

MAHALAKSHMI R (38060007)



**DEPARTMENT OF MATHEMATICS
SCHOOL OF SCIENCE AND HUMANITIES**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119

MARCH– 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai – 600 119

www.sathyabama.ac.in

DEPARTMENT OF MATHEMATICS

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **MAHALAKSHMI R(38060007)** who carried out the project entitled "**A SOLUTION OF QUADRATIC PROGRAMMING USING MATLAB**" under our supervision from **DECEMBER 2020 to MARCH 2021**.

Internal Guide External Guide (if Applicable)

(Dr. A. MOHAMED ISMAIL)

(Name in Capital letters (Name in Capital letters
with Signature) with signature)

(Dr. M. NIRMALA)

Head of the Department

(Name in Capital letters with Seal & Signature)

Submitted for Viva Voce Examination held on 09/04/2021

Internal Examiner External Examiner

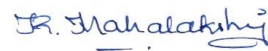
(Name in Capital letters (Name in Capital letters
with Signature) with signature)

DECLARATION

I **MAHALAKSHMI R** hereby declare that the Project Report entitled “**A SOLUTION OF QUADRATIC PROGRAMMING USING MATLAB**” done by me under the guidance of **Dr. A. Mohamed Ismail**(Internal) at **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** is submitted in partial fulfillment of the requirements for the award of Bachelor of Science Degree in Mathematics.

DATE: 24/03/2021

PLACE: CHENNAI



**SIGNATURE OF THE
CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to the Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to Dean, School of Science and Humanities and **Dr.M.Nirmala**, Head of the Department, Dept. of Mathematics for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. A. Mohamed Ismail** for her valuable guidance, suggestions, and constant encouragement that paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the Department of Mathematics who were helpful in many ways for the completion of the project.

ABSTRACT

Optimization is the process of maximizing or minimizing the objective function which satisfies the given constraints. There are two types of optimization problem linear and nonlinear. Linear optimization problem has wide range of applications, but all realistic problem cannot be modelled as linear program, so here nonlinear programming gains its importance. In the present work I have tried to find the solution of nonlinear programming Quadratic problem under different conditions such as when constraints are not present and when constraints are present in the form of equality and inequality sign. Wolfe's modified simplex method helps in solving the Quadratic programming problem by converting the quadratic problem in successive stages to linear programming which can be solved easily by applying two-phase simplex method. A variety of problems arising in the area of engineering, management etc. are modeled as optimization problems thus making optimization an important branch of modern applied mathematics. Because of its usefulness in production planning, financial and corporate planning, health care and hospital planning, quadratic programming (QP) problems have attracted considerable research and interest in recent years. In this paper, we first extend the simplex method for solving QP problems by replacing one basic variable at an iteration of simplex method. We then develop an algorithm and a computer technique for solving quadratic programming problem involving the product of two indefinite factorized linear functions. For developing the technique, we use programming language *MATLAB*. We also illustrate numerical examples to demonstrate our technique.

KEYWORDS

Quadratic programming, Wolfe's modified simplex method, Two-Phase method, Algorithm, Coding(Matlab).

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	i
	LIST OF ABBREVIATIONS	iii
	LIST OF TABLES	iii
1	INTRODUCTION	
	1.1 Introduction	1
	1.2 Pre requisites of Quadratic Programming	2
	1.3 Properties of Quadratic forms	4
2	LITERATURE SURVEY	5
3	AIM AND SCOPE OF THE PRESENT INVESTIGATION	
	3.1 Quadratic Programming	6
4	METHODOLOGY	
	4.1 Lagrangian Method	7
	4.2 Wolfe's simplex method	8
5	RESULTS AND DISCUSSIONS	
	5.1 Quadratic Programming By Wolfes Method in Matlab	15
6	SUMMARY AND CONCLUSIONS	23
	REFERENCES	24

LIST OF ABBREVIATIONS

- | | |
|---------|--------------------------------|
| 1. QP | Quadratic Programming |
| 2. LPP | Linear Programming Problem |
| 3. NLPP | Non-Linear Programming Problem |

LIST OF TABLES

TABLE NO	TITLE	PAGE NO
4.1	Initial Table	12
4.2	Iteration 1	12
4.3	Iteration 2	13
4.4	Iteration 3	13
4.5	Iteration 4	14

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Optimization constitutes a very important branch of modern applied mathematics. A variety of problems arising in the field of engineering design, operations research, management science, computer science, financial engineering and economics can be modelled as optimization is useful in real life.

It was the development of the simplex method for linear programming by G.B. Dantzig in the mid 40's which in the sense started the subject of mathematical optimization. Another major development was due to

H.W. Kuhn and A.W. Tucker in 1951 who gave necessary/sufficient optimality conditions for non-linear programming problem, now known as Karush-Kuhn Tucker(KKT) conditions. In 1939 W. Karush had already developed conditions similar to those given by Kuhn Tucker.

The presence of linearity structure on the given optimization problem gave beautiful mathematical results and also helped greatly in its algorithmic development. However most of the real world applications lead to optimization problems which are inherently nonlinear and are void of linearity.

1.2 PRE-REQUISITES TO QUADRATIC PROGRAMMING

Some definitions:

- **Vector:** A vector in n space is an ordered set of n real numbers. For e.g. $a = (a_1, a_2, \dots, a_n)$ is a vector of elements or components.
- **Null vector:** The null vector is a vector whose elements are all zero. $0 = (0, 0, \dots, 0)$. The null vector corresponds to origin.
- **Sum vector :** The sum vector is a vector whose elements are all one. $1 = (1, 1, \dots, 1)$.
- **Unit vector (e_i) :** The unit vector (e_i) is a vector whose i^{th} element is one. $e_i = (1, 0, \dots, 0)$.
 E^2 , there are two unit vectors. E^n , there are n unit vectors.
- **Orthogonal vectors:** Two vectors a and b are said to be orthogonal if $a \cdot b = 0$
- **Linear independence:** A set of vectors $a_1, a_2, \dots, a_k$ is linearly independent if the equation $\lambda_1 a_1 + \lambda_2 a_2 + \dots + \lambda_k a_k = 0$ is satisfied only if $\lambda_1 = \lambda_2 = \dots = \lambda_k = 0$.
- **Linear dependence:** A set of vectors which are not linearly independent are called linearly dependent.
- **Spanning set:** The set of vectors $a_1, a_2, \dots, a_k$ in E^n is a spanning set in E^n if every vector in E^n can be expressed as a linear combination of vectors $a_1, a_2, \dots, a_k$ where $(k < n)$.
- **Basis set:** A set of vectors $a_1, a_2, \dots, a_k$ in E^n is a basis set if
 - i) it is linearly independent set
 - ii) it is a spanning set of E^n , if it is a basis then $k=n$.
- **Standard basis :** The set of unit vectors $e_1, e_2, e_3, \dots, e_n$ is called the standard basis for E^n .
- **Matrix :** A matrix is a rectangular array of ordered numbers, arranged into rows and columns. $A = [a_{ij}]_{m \times n}$. The elements a_{ij} for $i=j$ i.e. a_{11}, a_{22}, a_{33} and so on are called principal diagonal elements, others are called off diagonal elements.
- **Square matrix:** Any matrix in which number of rows is equal to number of columns is known as square matrix ($m=n$).

- **Diagonal matrix:** A square matrix in which all off diagonal elements are zero i.e, $a_{ij}=0$ for $(i \neq j)$ is called a diagonal matrix.
- **Identity or Unit matrix :** A diagonal matrix whose all principal elements are 1 is called an Identity or unit matrix denoted simply by I .
- **Transpose matrix:** The transpose of a matrix $A=[a_{ij}]$ denoted by A^T is a matrix obtained by interchanging the rows and columns of A .
- **Symmetric matrix:** A square matrix A is said to be symmetric if the matrix A remains the same by interchanging the rows and columns of A (i.e, $a_{ij}=a_{ji}$ or $A^T=A$)
- **Row matrix:** A matrix having only a single row is called a row matrix. It is an $1 \times n$ matrix.
- **Column matrix:** A matrix having only a single column is called a column matrix. It is an $m \times 1$ matrix.
- **Null matrix:** A matrix whose all elements are zero is called a null matrix.
- **Rank of a matrix:** A positive integer r is said to be the rank of a matrix A denoted by $\rho(A)$ if
 - i) Matrix A possess at least one r -rowed minor which is not zero.
 - ii) Matrix A does not possess any nonzero $(r+1)$ -rowed minor.
- **Equivalent matrices:** Two matrices A and B are said to be equivalent, if and only if $\rho(A)=\rho(B)$ denoted by $A \sim B$.
- **Quadratic forms:** Let $x=(x_1, x_2, \dots, x_n)$ and $n \times n$ matrix $A=[a_{ij}]$ then a function of n variables denoted by $f(x_1, x_2, \dots, x_n)$ or $Q(x)$ is called a quadratic forms in n space if $Q(x)=x^T A x = \sum \sum a_{ij} x_i x_j$

1.3 Properties of Quadratic forms:

- i) **Positive definite** : A quadratic form $Q(x)$ is positive definite iff $Q(x)$ is positive (>0) for all $x \neq 0$.
- ii) **Positive semi-definite**: A quadratic form $Q(x)$ is positive semi definite iff, $Q(x)$ is non-negative (≥ 0) for all x and there exists an $x \neq 0$ for which $Q(x)=0$.
- iii) **Negative definite**: A quadratic form $Q(x)$ is negative definite iff, $-Q(x)$ is positive definite.
- iv) **Negative semi-definite**: A quadratic form $Q(x)$ is negative semi-definite iff, $-Q(x)$ is positive semi- definite.
- v) **Indefinite**: A quadratic form $Q(x)$ is indefinite if $Q(x)$ is positive for some x and negative for some other.

➤ ***Difference equation***: An equation relating the values of a function y and one or more of its differences $\Delta y, \Delta^2 y, \dots$ for each value of a set of numbers is called a difference equation.

➤ ***Order of difference equation***: The difference between the highest and the lowest suffix of the equation is called the order of difference equation.

$$Y_{k+1} + 3y_k = 0$$

Here, $k+1-k=1$ is the order of the difference equation.

➤ ***Feasible solution***: Solution values of the decision variables $x_j (j=1,2,3,\dots,n)$ which satisfy the constraints and non-negativity conditions is known as feasible solution.

➤ ***Basic feasible solution***: Collection to all feasible solutions to a problem constitutes a convex set whose extreme points correspond to the basic feasible solution.

CHAPTER 2

LITERATURE SURVEY

A literature survey or a literature review in a project report is that section which shows the various analyses and research made in the field of your interest and the results already published, taking into account the various parameters of the project and the extent of the project. It is the most important part of your report as it gives you a direction in the area of your research. Various methodologies have been used by the researchers all over the world and some of the methods were discussed below.

Mokhter S Bazaraa, Hanif D Sherali, C.N. Shetty “NONLINEAR PROGRAMMING THEORY AND ALGORITHMS” in their paper carried out various methods for solving nonlinear programming and find the optimal solution. Wolfe’s Modified Simplex Method was taken for finding optimal solution.

Chandra S, Jayadeva, Mehra A, Alpha Science International Ltd., 2009 in their working paper titled “NUMERICAL OPTIMIZATION WITH APPLICATIONS” one may refer the theory of Wolfe’s method.

F.S. Hiller, G.J. Lieberman, The McGraw-Hill Companies, 2001 paper titled as “INTRODUCTION TO OPERATIONS RESEARCH” they used many methods to solve LPP and NLPP.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

QUADRATIC PROGRAMMING

Let x^T and $c \in \mathbb{R}^n$. Let Q be a symmetric $n \times n$ real matrix. Then,

$$\text{Maximize } z = f(x) - cx + \frac{1}{2} x^T Qx$$

subject to the constraints $Ax \leq b$ and $x \geq 0$

where $b^T \in \mathbb{R}^m$ and A is a $m \times n$ real matrix, is called a general quadratic programming problem.

Note: $x^T Qx$ represents a quadratic form. The reader may recall that a quadratic form $x^T Qx$ is said to be positive-definite (negative-definite) if $x^T Qx > 0$ (< 0) for $x \neq 0$ and positive-semi-definite (negative-semi-definite) if $x^T Qx \geq 0$ (≤ 0) for all x such that there is one $x \neq 0$ satisfying $x^T Qx = 0$.

It can easily be shown that

1. If $x^T Qx$ is positive-semi-definite (negative-semi-definite) then it is convex (concave) in x over all of $\mathbb{R}^n$,
2. If $x^T Qx$ is positive-definite (negative-definite) then it is strictly convex (strictly concave) in x over all of $\mathbb{R}^n$.

CHAPTER 4

METHODOLOGY

4.1 LAGRANGIAN METHOD

In non-linear programming problem if objective function is differentiable and has equality constraints optimization can be achieved by the use of Lagrange multipliers.

FORMULATION:

Consider the problem of maximizing or minimizing $z = f(x_1, x_2)$ subject to the constraints $g(x_1, x_2) = c$ and $x_1, x_2 \geq 0$ where c is a constant. We assume that $f(x_1, x_2)$ and $g(x_1, x_2)$ are differentiable with respect to x_1 and x_2 . Let us introduce a differentiable function $h(x_1, x_2)$ differentiable with respect to x_1 and x_2 and defined by $h(x_1, x_2) \equiv g(x_1, x_2) - c$.

The problem is restated as maximize $z = f(x_1, x_2)$ subject to the constraints $h(x_1, x_2) = 0$ and $x_1, x_2 \geq 0$.

To find the necessary conditions for a maximum (or minimum) value of z , a new function is formed by introducing a Lagrange multiplier λ , as $L(x_1, x_2, \lambda) = f(x_1, x_2) - \lambda h(x_1, x_2)$.

The number λ is an unknown constant and the function $L(x_1, x_2, \lambda)$ is called the Lagrangian function with Lagrange multiplier λ .

Necessary condition:

$$\frac{\partial L}{\partial x_1}(x_1, x_2, \lambda) = 0$$

$$\frac{\partial L}{\partial x_2}(x_1, x_2, \lambda) = 0$$

$$\frac{\partial L}{\partial \lambda}(x_1, x_2, \lambda) = 0$$

Their partial derivatives are given by

$$\frac{\partial L}{\partial x_1} = \frac{\partial f}{\partial x_1} - \lambda \frac{\partial h}{\partial x_1}$$

$$\frac{\partial L}{\partial x_2} = \frac{\partial f}{\partial x_2} - \lambda \frac{\partial h}{\partial x_2}$$

$$\frac{\partial L}{\partial \lambda} = -h \text{ where } L, f, h \text{ stands for the functions.}$$

The necessary conditions for maximum or minimum of $f(x_1, x_2)$ are given by $f_1 = \lambda h_1$;

$$f_2 = \lambda h_2 \text{ and } -h(x_1, x_2) = 0.$$

Sufficient condition:

Let the Lagrangian function for n variables and one constraint be $L(x, \lambda) = f(x) - \lambda h(x)$.

The necessary conditions for a stationary point to be a maximum or minimum are

$$\frac{\partial L}{\partial x_j} = \frac{\partial f}{\partial x_j} - \lambda \frac{\partial h}{\partial x_j} = 0 \quad (j=1, 2, \dots, n) \text{ and}$$

$$\frac{\partial L}{\partial \lambda} = -h(x) = 0$$

4.2 WOLFE'S MODIFIED SIMPLEX METHOD

The iterative steps of the method are:

1. Convert the inequality constraints into equations by introducing slack variables y_i^2 ($i = 1, 2, 3, \dots, m$) ≥ 0 ; and the non-negative restrictions by introducing slack variables s_j^2 ($j = 1, 2, 3, \dots, n$) ≥ 0 .
2. Construct the Lagrangian function $L(x, s, y, \lambda, \mu) = f(x) - \lambda[Ax + y^2 - b] - \mu[-x + s^2]$
3. Differentiating $L(x, s, y, \lambda, \mu)$ partially with respect to the components of x, s, y and then equating them to zero. Thus, we have

$$(1) \quad c - \frac{1}{2}(2x^T D) - \lambda A + \mu = 0$$

$$\Rightarrow c_j - \sum_{k=1}^n x_k d_{jk} - \sum_{i=1}^m \lambda_i a_{ij} - \mu_j = 0; \quad j = 1, 2, 3, \dots, n$$

- (2) $-2 \mu_j s_j = 0$ or $\mu_j x_j = 0$; $j = 1, 2, 3, \dots, n$
- (3) $-2 \lambda_i y_i = 0$ or $\lambda_i y_i^2 = 0$ or $\lambda_i [\sum_{j=1}^n a_{ij} x_j - b_i] = 0$; $i = 1, 2, 3, \dots, m$
- (4) $Ax + y^2 - b = 0$ or $Ax \leq b$
- (5) $-x + s^2 = 0$ or $x \geq 0$
- (6) $\lambda_i, \mu_j, x_j, y_j, s_j \geq 0$; $i = 1, 2, 3, \dots, m$ and $j = 1, 2, 3, \dots, n$
4. Introduce the artificial variables a_j ; ($j = 1, 2, 3, \dots, n$) in the condition (1) and construct an objective function, $z = a_1 + a_2 + \dots + a_n$.
5. Obtain an initial basic feasible solution to the L.P.P.:
- Minimize $z = a_1 + a_2 + \dots + a_n$ subject to the constraints:
- $\sum_{k=1}^n d_{jk} x_k + \sum_{i=1}^m \lambda_i a_{ij} - \mu_j + a_j = c_j$; $j = 1, 2, 3, \dots, n$;
- $\sum_{j=1}^n a_{ij} x_j + y_i^2 = b_i$; $i = 1, 2, 3, \dots, m$ and
- $\lambda_i, x_j, \mu_j, y_i, a_j \geq 0$;
- $\lambda_i \cdot y_i = 0, \mu_j \cdot x_j = 0$ (complementary slackness)
6. Use two-phase simplex method to obtain optimum solution to the L.P.P. of step 5.
7. The optimum solution obtained in step 6 is as optimum solution to the given quadratic programming problem also.

SOLVED EXAMPLE:

Use wolfe's method in solving the following quadratic programming problem:

$$\text{Max } z = 6x_1 + 3x_2 - 2x_1^2 - 3x_2^2 - 4x_1x_2$$

Subject to the constraints :

$$x_1 + x_2 \leq 1$$

$$2x_1 + 3x_2 \leq 4$$

$$\text{And } x_1, x_2 \geq 0$$

SOLUTION:

$$L = F - \lambda_1 g_1 - \lambda_2 g_2 - \mu_1 h_1 - \mu_2 h_2 \quad \text{-----(1)}$$

Where

$$F = 6x_1 + 3x_2 - 2x_1^2 - 3x_2^2 - 4x_1x_2$$

$$G_1 = x_1 + x_2 + s_1^2 = 1$$

$$G_2 = 2x_1 + 3x_2 + s_2^2 = 4$$

$$H_1 = -x_1 + r_1^2 = 0$$

$$H_2 = -x_2 + r_2^2 = 0$$

Substituting F, g₁, g₂, μ₁, μ₂ in equation (1)

$$L = (6x_1 + 3x_2 - 2x_1^2 - 3x_2^2 - 4x_1x_2) - \lambda_1 (x_1 + x_2 + s_1^2 - 1) - \lambda_2 (2x_1 + 3x_2 + s_2^2 - 4) - \mu_1 (-x_1 + r_1^2) - \mu_2 (-x_2 + r_2^2)$$

$$\frac{\partial L}{\partial x_1} = 0 \Rightarrow 6 - 4x_1 - 4x_2 - \lambda_1(1) - \lambda_2(2) - \mu_1(-1) = 0$$

$$\Rightarrow 4x_1 + 4x_2 + \lambda_1 + 2\lambda_2 - \mu_1 = 6 \quad \text{-----(i)}$$

$$\frac{\partial L}{\partial x_2} = 0 \Rightarrow 3 - 6x_2 - 4x_1 - \lambda_1(1) - \lambda_2(3) - \mu_2(-1) = 0$$

$$\Rightarrow 4x_1 + 6x_2 + \lambda_1 + 3\lambda_2 - \mu_2 = 3 \quad \text{-----(ii)}$$

$$\frac{\partial L}{\partial \lambda_1} = 0 \Rightarrow -(x_1 + x_2 + s_1^2 - 1) = 0$$

$$\Rightarrow x_1 + x_2 + s_1^2 = 1 \quad \text{----- (iii)}$$

$$\frac{\partial L}{\partial \lambda_2} = 0 \Rightarrow -(2x_1 + 3x_2 + s_2^2 - 4) = 0$$

$$\Rightarrow 2x_1 + 3x_2 + s_2^2 = 4 \quad \text{-----(iv)}$$

$$\frac{\partial L}{\partial \mu_1} = 0 \Rightarrow -(-x_1 + r_1^2) = 0$$

$$\Rightarrow -x_1 + r_1^2 = 0 \quad \text{-----(v)}$$

$$\frac{\partial L}{\partial \mu_2} = 0 \Rightarrow -(-x_2 + r_2^2) = 0$$

$$\Rightarrow -x_2 + r_2^2 = 0 \quad \text{-----}(vi)$$

$$\frac{\partial L}{\partial s_1} = 0 \Rightarrow -2\lambda_1 s_1 = 0$$

$$\Rightarrow \lambda_1 s_1 = 0 \quad \text{-----}(vii)$$

$$\frac{\partial L}{\partial s_2} = 0 \Rightarrow -2\lambda_2 s_2 = 0$$

$$\Rightarrow \lambda_2 s_2 = 0 \quad \text{-----}(viii)$$

$$\frac{\partial L}{\partial r_1} = 0 \Rightarrow -2\mu_1 r_1 = 0$$

$$\Rightarrow \mu_1 r_1 = 0$$

$$\Rightarrow \mu_1 x_1 = 0 \quad \text{-----}(ix)$$

$$\frac{\partial L}{\partial r_2} = 0 \Rightarrow 2\mu_2 r_2 = 0$$

$$\Rightarrow \mu_2 r_2 = 0$$

$$\Rightarrow \mu_2 x_2 = 0 \quad \text{-----}(x)$$

$$\text{Min } z = A_1 + A_2$$

$$(i) \Rightarrow 4x_1 + 4x_2 + \lambda_1 + 2\lambda_2 - \mu_1 + A_1 = 6$$

$$(ii) \Rightarrow 4x_1 + 6x_2 + \lambda_1 + 3\lambda_2 - \mu_2 + A_2 = 3$$

$$(iii) \Rightarrow x_1 + x_2 + S_1 = 1$$

$$(iv) \Rightarrow 2x_1 + 3x_2 + S_2 = 4$$

$$-x_1 + r_1^2 = -x_2 + r_2^2 = 0$$

$$\lambda_1 s_1 = \lambda_2 s_2 = 0$$

$$\mu_1 x_1 = \mu_2 x_2 = 0$$

$$x_1, x_2, S_1, S_2, \lambda_1, \lambda_2, \mu_1, \mu_2, A_1, A_2 \geq 0$$

INITIAL TABLE:

		C_j	0	0	0	0	0	0	0	0	1	1	
C_B	Y_B	X_B	x_1	x_2	S_1	S_2	λ_1	λ_2	μ_1	μ_2	A_1	A_2	Ratio
1	A_1	6	4	4	0	0	1	2	-1	0	1	0	1.5
1	A_2	3	4	6	0	0	1	3	0	-1	0	1	0.5
0	S_1	1	1	1	1	0	0	0	0	0	0	0	1
0	S_2	4	2	3	0	1	0	0	0	0	0	0	1.33
		Z_j	8	10	0	0	2	5	-1	-1	1	1	
		$C_j - Z_j$	-8	-10	0	0	-2	-5	1	1	0	0	

Table 4.1

Leaving variable : A_2

Entering variable : x_2

Key element : 6

ITERATION 1:

		C_j	0	0	0	0	0	0	0	0	1	1	
C_B	Y_B	X_B	x_1	x_2	S_1	S_2	λ_1	λ_2	μ_1	μ_2	A_1	A_2	Ratio
1	A_1	4	1.33	0	0	0	0.33	0	-1	1	1	-	3
0	x_2	0.5	0.66	1	0	0	0.16	0.5	0	-0.16	0	-	0.75
0	S_1	0.5	0.33	0	1	0	-0.16	-0.5	0	0.16	0	-	1.5
0	S_2	2.5	0	0	0	1	-0.5	-1.5	0	0.5	0	-	-
		Z_j	1.33	0	0	0	0.33	0	-1	1	1	-	
		$C_j - Z_j$	-1.33	0	0	0	-0.33	0	1	-1	0	-	

Table 4.2

Leaving variable : x_2

Entering variable : x_1

Key element : 0.66

ITERATION 2:

		C_j	0	0	0	0	0	0	0	0	1	1	
C_B	Y_B	X_B	x_1	x_2	S_1	S_2	λ_1	λ_2	μ_1	μ_2	A_1	A_2	Ratio
1	A_1	3	0	-2	0	0	0	-1	-1	1.33	1	-	2.25
0	x_1	0.75	1	1.5	0	0	0.25	0.75	0	-0.25	0	-	-3
0	S_1	0.25	0	-0.5	1	0	-0.25	-0.75	0	0.25	0	-	1
0	S_2	2.5	0	0	0	1	-0.5	-1.5	0	0.5	0	-	5
		Z_j	0	-2	0	0	0	-1	-1	1.33	1	-	
		$C_j - Z_j$	0	2	0	0	0	1	1	-1.33	0	-	

Table 4.3

Leaving variable : S_1

Entering variable : μ_2

Key element : 0.25

ITERATION 3:

		C_j	0	0	0	0	0	0	0	0	1	1	
C_B	Y_B	X_B	x_1	x_2	S_1	S_2	λ_1	λ_2	μ_1	μ_2	A_1	A_2	Ratio
1	A_1	1.66	0	0.66	-5.33	0	1.33	3	-1	0	1	-	1.25
0	x_1	1	1	1	1	0	0	0	0	0	0	-	-
0	μ_2	1	0	-2	4	0	-1	-3	0	1	0	-	-
0	S_2	2	0	1	-2	1	0	0	0	0	0	-	-

		Z_j	0	0.66	-5.33	0	1.33	3	-1	0	1	-	
		$C_j - Z_j$	0	-0.66	5.33	0	-1.33	-3	1	0	0	-	

Table 4.4

Leaving variable : A_1

Entering variable : λ_1

Key element : 1.33

ITERATION 4:

		c_j	0	0	0	0	0	0	0	0	1	1
C_B	Y_B	X_B	x_1	x_2	S_1	S_2	λ_1	λ_2	μ_1	μ_2	A_1	A_2
0	λ_1	1.25	0	0.5	-4	0	1	2.25	-0.75	0	-	-
0	x_1	1	1	1	1	0	0	0	0	0	-	-
0	μ_2	2.25	0	-1.5	0	0	0	-0.75	-0.75	1	-	-
0	S_2	2	0	1	-2	1	0	0	0	0	-	-
		Z_j	0	0	0	0	0	0	0	0		
		$C_j - Z_j$	0	0	0	0	0	0	0	0	1	1

Table 4.5

The optimum solution is obtained :

$x_1 = 1$ and $x_2 = 0$

CHAPTER 5

RESULTS AND DISCUSSIONS

5.1 QUADRATIC PROGRAMMING BY WOLFE'S METHOD IN MATLAB:

The function `wolf` solves an QPP using Wolf or restricted entry simplex method.

Input:- 1)D : The Hessian of the objective function. It is a required parameter.

2)l : The (column) vector of coefficient of linear terms in the objective function. It is a required parameter.

3)b : The (column) vector containing right hand side constant of the constraints. It is a required parameter.

4)Mat : The coefficient matrix of the left hand side of the constraints. it is a required parameter.

5)inq : A (row) vector indicating the type of constraints as 1 for $\geq$, 0 for $=$ and -1 for $\leq$ constraints. If inq is not supplied then it is by default taken that all constraints are of $\leq$ type. It is an optional parameter.

6)minimize : This parameter indicates whether the objective function is to be minimized. `minimized = 1` indicates a minimization problem and `minimization = 0` stands for a maximization problem. By default it is taken as 0. It is an optional parameter.

Example :

$$\text{Max } z = 2x_1 + 3x_2 - 3x_1^2 + 2x_1x_2 - 2x_2^2$$

$$\text{Subject to } x_1 + x_2 \geq 1$$

$$3x_1 + 4x_2 \leq 3$$

$$x_1, x_2 \geq 0$$

SOLUTION :

$$D = [-3 \ 1; 1 \ -2]; l = [2; 3]; b = [1; 3]; \text{Mat} = [1 \ 1; 3 \ 4]; \text{inq} = [1 \ -1].$$

After supplying these inputs call `wolf(D,l,b,Mat,inq)`.

```

function f = myfun(x)
f = 2x(1) + 3x(2) - 3x(1)^2 + 2x(1)x(2) - 2x(2)^2
D = [-3 1;1 -2]
l = [2;3]
b = [1;3]
Mat = [1 1;3 4]
inq = [1 -1]
function [x,fval]=wolf(D,l,b,Mat,inq,minimize)
n = length(l);
m = length(b);
if ~isequal(size(Mat,1),m) || ~isequal(length(inq),m) || ~isequal(size(D,1),...
    size(D,2)) || ~isequal(size(D,1),n) || ~isequal(size(Mat,2),n)
    fprintf('\nError: Dimension mismatch!\n');
return
end
if nargin < 4 || nargin > 6
    fprintf('\nError: Number of input arguments are inappropriate!\n');
return
end
if nargin < 5
    minimize = 0;
    inq = -ones(m,1);
elseif nargin < 6
    minimize = 0;
end
if minimize == 1
    l = -l;
    D = -D;
end
if min(eig(-D)) < 0 % Checking convexity of Hessian

```

```

    fprintf('\nError: Wolf method may not converge to global optimum!\n');
return
elseif (min(eig(-D)) == 0) && ~isempty(find(l,1))
    fprintf('\nError: Wolf method may not converge to global optimum!\n');
return
end
count = n;
for i = 1 : m
    if (inq(i) > 0)
        Mat(i,:) = -Mat(i,:);
        b(i) = -b(i);
    elseif (inq(i) == 0)
        count = count + 1;
        Mat(i,count) = -1;
        l(count) = 0;
        D(count,count) = 0;
    end
end
a = [-2*D Mat' -eye(count,count) zeros(count,m); Mat zeros(m,m + count) eye(m,m)];
d = [l;b];
for i = 1 : count + m
    if(d(i) < 0)
        d(i) = -d(i);
        a(i,:) = -a(i,:);
    end
end
cb = zeros(1,count + m);
bv = zeros(1,count + m);
nbv = (1 : 2 * (count + m));
c = zeros(1,2 * (count + m));
rem = zeros(1,count + m);

```

```

for i = 1 : count + m
if(a(i,count + m + i) == -1)
    bv(i) = 2 * (count + m) + i;
    cb(i) = -1;
elseif(a(i,count + m + i) == 1)
    rem(i)=count + m + i;
    bv(i) = count + m + i;
    cb(i) = 0;
end
end
[h,j,k] = find(rem);
a(:,k) = [];
c(k) = [];
nbv(k) = [];
r = cb * a - c;
exitflg = 0;
iter = 0;
z = cb * d;
[w,y] = size(a);
opt = 0;
while(exitflg == 0)

    iter = iter + 1;
    fprintf('\n\n %d th tableau:\n',iter );% If you don't need
    fprintf('\n\t\t\tBV\t');disp(nbv);%intermedite tableaus be
    disp([bv' d a ;0 z r]);%printed, just comment these lines.
    r_new = r;
    found = 0;

while found == 0

```

```

[u,v] = min(r_new);
leave = 0;

if ~(u < 0)
if abs(z) > 10^-6
    fprintf('\nError: Wolf method fails to find optimum!\n');
    exitflg = 1;
    found = 1;
else
    fprintf('\nThe optimum has achieved!\n');
    exitflg = 1; opt = 1;
    found = 1;
end
else

    ratio = bitshift(1,30);
    check = 0;

for i = 1 : w
if bv(i) <= 2 * (count + m) && abs(bv(i) - nbv(v)) == count + m
    check = 1;
end
end

if check == 0
for i = 1 : w
if a(i,v) > 0 && (d(i) / a(i,v)) < ratio
    ratio = d(i) / a(i,v);
    leave = i;
end
end

```

```

fprintf('\nEntering Variable:'); disp(nbv(v));
fprintf('\nLeaving Variable:'); disp(bv(leave));

for i = 1 : w
for j = 1 : y
if i ~= leave && j ~= v
    
$$a(i,j) = a(i,j) - a(i,v) * a(leave,j) / a(leave,v);$$

end
end
end

    
$$z = z - d(leave) * r(v) / a(leave,v);$$


for j = 1 : y
if j ~= v
    
$$r(j) = r(j) - r(v) * a(leave,j) / a(leave,v);$$

    
$$a(leave,j) = a(leave,j) / a(leave,v);$$

end
end

for i = 1 : w
if i ~= leave
    
$$d(i) = d(i) - a(i,v) * d(leave) / a(leave,v);$$

    
$$a(i,v) = -a(i,v) / a(leave,v);$$

end
end

    
$$d(leave) = d(leave) / a(leave,v);$$

    
$$a(leave,v) = 1 / a(leave,v);$$

    
$$r(v) = -r(v) / a(leave,v);$$


```

```

        temp = nbv(v);
        nbv(v) = bv(leave);
        bv(leave) = temp;
        found = 1;
elseif check == 1
    r_new(v) = 1;
end
end
end
end
if opt == 1
    x = zeros(n,1);
for i = 1 : w
    if bv(i) <= n
        x(bv(i)) = d(i);
    end
end
    fval = x'*I+x'*D*x;
if minimize == 1
    fval = -fval;
end
end

```

OUTPUT:

Minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in [feasibledirections](#), to within the value of the [optimality tolerance](#), and constraints are satisfied to within the value of the [constraint tolerance](#).

[<stopping criteria details>](#)

Ans:

0.3012

0.5201

CHAPTER 6

SUMMARY AND CONCLUSIONS

Present work demonstrates methods to solve the optimization problems which are of Quadratic in nature .As discussed earlier concept Lagrangian method have been used to solve the optimization problems.

Wolfe's method converts the Quadratic programming to linear programming in successive steps which can be solved easily by two phase simplex method. Thus, the Quadratic Programming Problems can be handled easily.

REFERENCES

- Swarup, K., P. k. Gupta and Man Mohan, 2003, "Tracts in Operations Research", Eleventh Thoroughly Revised Edition, ISBN:81-8054-059-6
- Frederick S. Hillier and Gerald J. Lieberman, McGraw-Hill International Edition 1995, " Introduction to Operations Research"
- Wayne L. Winston, (1994), "Application and algorithms," Dexbury press, Belmont, California, U.S.A.
- Wolfe, P., (1959), "The simplex method for quadratic programming," *Econometrica*, **27**(3), 382-398.
- Swarup, K. 1966, "Quadratic Programming" CCERO (Belgium), **8**(2), 132-136.

